

# **INTERNET AND WEB TECHNOLOGY NOTES**

**BY**

**VINEET SINGH**

**(GOVERNMENT POLYTECHNIC KANPUR)**

**(2021-22)**

# Internet

Internet is a global network that connects billions of computers across the world with each other and to the World Wide Web. It uses standard internet protocol suite (TCP/IP) to connect billions of computer users worldwide. It is set up by using cables such as optical fibers and other wireless and networking technologies. At present, internet is the fastest mean of sending or exchanging information and data between computers across the world.



It is believed that the internet was developed by "Defense Advanced Projects Agency" (DARPA) department of the United States. And, it was first connected in 1969.

## Why is the Internet Called a Network?

Internet is called a network as it creates a network by connecting computers and servers across the world using routers, switches and telephone lines, and other communication devices and channels. So, it can be considered a global network of physical cables such as copper telephone wires, fiber optic cables, tv cables, etc. Furthermore, even wireless connections like 3G, 4G, or Wi-Fi make use of these cables to access the Internet.

Internet is different from the **World Wide Web**

as the World Wide Web is a network of computers and servers created by connecting them through the internet. So, the internet is the backbone of the web as it provides the technical infrastructure to establish the **WWW** and acts as a medium to transmit information from one computer to another computer. It uses web browsers to display the information on the client, which it fetches from web servers.

The internet is not owned by a single person or organization entirely. It is a concept based on physical infrastructure that connects networks with other networks to create a global network of billions of computers. As of 12 August 2016, there were more than 300 crores of internet users across the world.

## How does internet work?

### **Before understanding this let us understand some basics related to internet:**

The internet works with the help of clients and servers. A device such as a laptop, which is connected to the internet is called a client, not a server as it is not directly connected to the internet. However, it is indirectly connected to the internet through an Internet Service Provider (ISP) and is identified by an IP address, which is a string of numbers. Just like you have an address for your home that uniquely identifies your home, an IP address acts as the shipping address of your device. The IP address is provided by your ISP, and you can see what IP

address your ISP has given to your system.

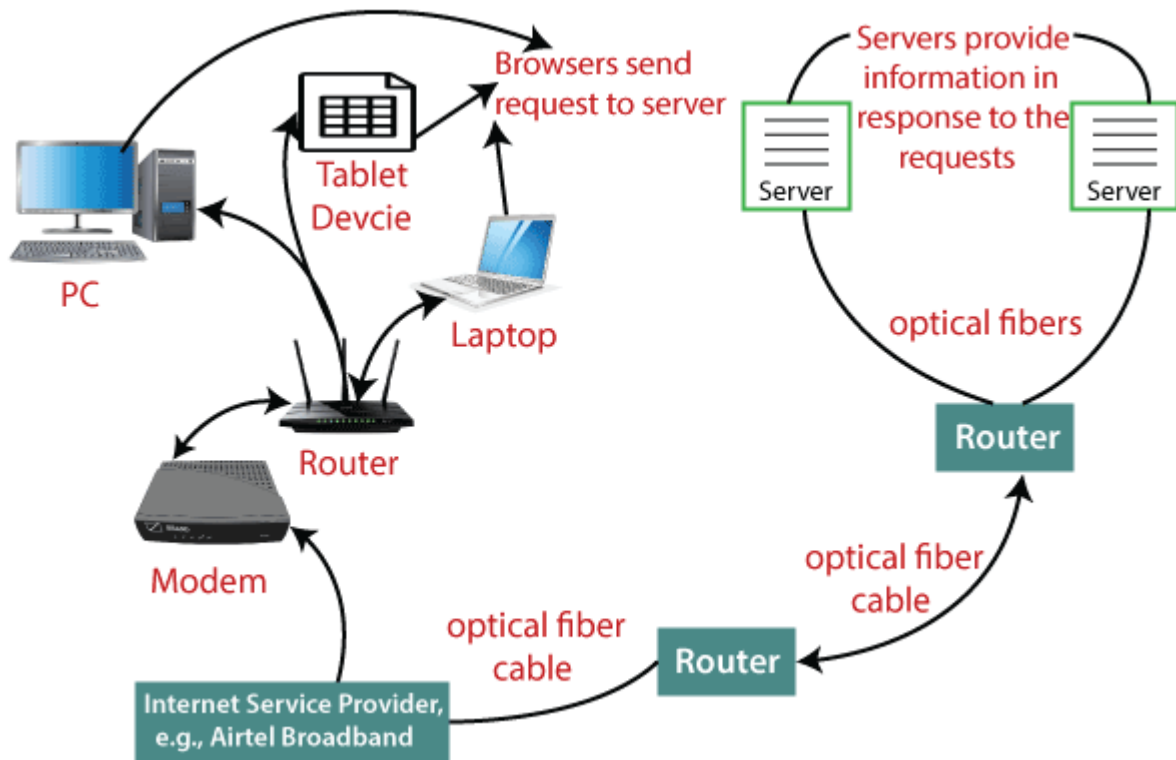
A server is a large computer that stores websites. It also has an IP address. A place where a large number of servers are stored is called a data center. The server accepts requests send by the client through a browser over a network (internet) and responds accordingly.

To access the internet we need a domain name, which represents an IP address number, i.e., each IP address has been assigned a domain name. For example, youtube.com, facebook.com, paypal.com are used to represent the IP addresses. Domain names are created as it is difficult for a person to remember a long string of numbers. However, internet does not understand the domain name, it understands the IP address, so when you enter the domain name in the browser search bar, the internet has to get the IP addresses of this domain name from a huge phone book, which is known as DNS

(Domain Name Server).

For example, if you have a person's name, you can find his phone number in a phone book by searching his name. The internet uses the DNS server in the same way to find the IP address of the domain name. DNS servers are managed by ISPs or similar organizations.

### **Now after understanding the basics, let us see how internet works?**



When you turn on your computer and type a domain name in the browser search bar, your browser sends a request to the DNS server to get the corresponding IP address. After getting the IP address, the browser forwards the request to the respective server.

Once the server gets the request to provide information about a particular website, the data starts flowing. The data is transferred through the optical fiber cables in digital format or in the form of light pulses. As the servers are placed at distant places, the data may have to travel thousands of miles through optical fiber cable to reach your computer.

The optical fiber is connected to a router, which converts the light signals into electrical signals. These electrical signals are transmitted to your laptop using an Ethernet cable. Thus, you receive the desired information through the internet, which is actually a cable that connects you with the server.

Furthermore, if you are using wireless internet using wifi or mobile data, the signals from the optical cable are first sent to a cell tower and from where it reaches to your cell phone in the form of electromagnetic waves.

The internet is managed by ICANN (Internet Corporation for Assigned Names and Numbers) located in the USA. It manages IP addresses assignment, domain name registration, etc.

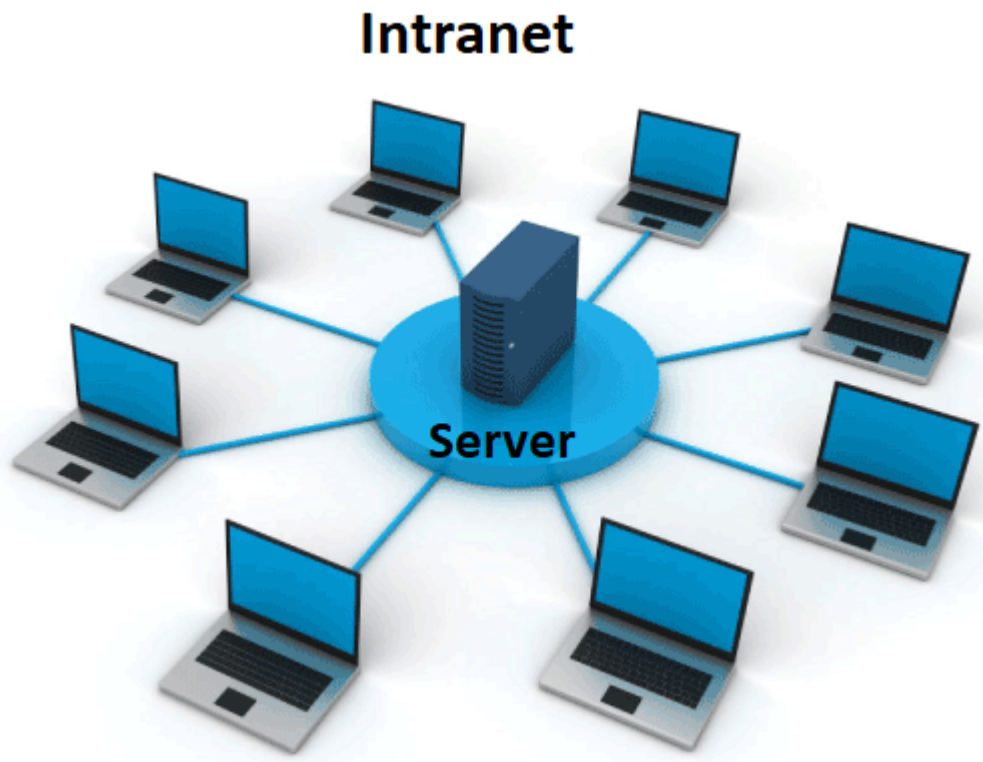
The data transfer is very fast on the internet. The moment you press enter you get the information from a server located thousands of miles away from you. The reason

for this speed is that the data is sent in the binary form (0, 1), and these zeros and ones are divided into small pieces called packets, which can be sent at high speed.

## Advantages of the Internet:

- **Instant Messaging:** You can send messages or communicate to anyone using internet, such as email, voice chat, video conferencing, etc.
- **Get directions:** Using GPS technology, you can get directions to almost every place in a city, country, etc. You can find restaurants, malls, or any other service near your location.
- **Online Shopping:** It allows you to shop online such as you can be clothes, shoes, book movie tickets, railway tickets, flight tickets, and more.
- **Pay Bills:** You can pay your bills online, such as electricity bills, gas bills, college fees, etc.
- **Online Banking:** It allows you to use internet banking in which you can check your balance, receive or transfer money, get a statement, request cheque-book, etc.
- **Online Selling:** You can sell your products or services online. It helps you reach more customers and thus increases your sales and profit.
- **Work from Home:** In case you need to work from home, you can do it using a system with internet access. Today, many companies allow their employees to work from home.
- **Entertainment:** You can listen to online music, watch videos or movies, play online games.
- **Cloud computing:** It enables you to connect your computers and internet-enabled devices to cloud services such as cloud storage, cloud computing, etc.
- **Career building:** You can search for jobs online on different job portals and send you CV through email if required.

# Intranet



The intranet is a private network that belongs to a particular organization. It is designed for the exclusive use of an organization and its associates, such as employees, customers, and other authorized people. It offers a secure platform to convey information and share data with authorized users. Confidential information, database, links, forms, and applications can be made available to the staff through the intranet. So, it is like a private internet or an internal website that is operating within an organization to provide its employees access to its information and records. Each computer in intranet is identified by a unique IP Address.

It is based on internet protocols (TCP/[IP](#)) and is protected from unauthorized access with firewalls and other security systems. The firewall monitors the incoming and outgoing data packets to ensure they don't contain unauthorized requests. So, users on the intranet can access the [internet](#), but the internet users can't access the intranet if they are not authorized for it. Furthermore, to access the intranet, the authorized user is required to be connected to its [LAN](#) (Local Area Network).

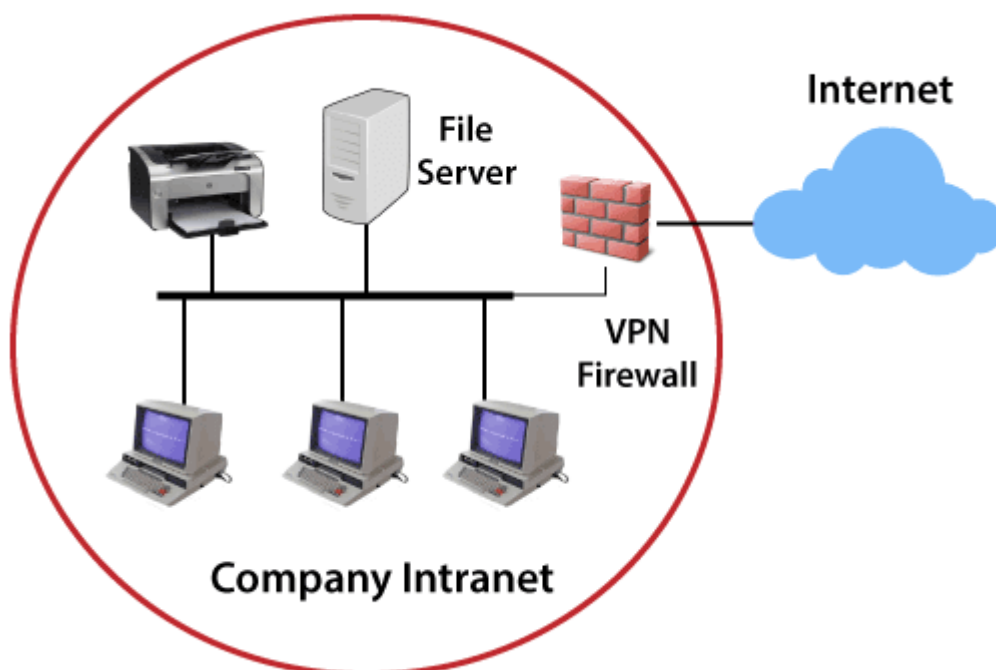
## **Some of the benefits of the intranet are:**

- It is cheap and easy to implement and run, and is more safe than the internet and [extranet](#).
- It streamlines communication that enables the company to share its data, information, and other resources among employees without any delay. The entire staff can receive company's announcements, ask questions, and access internal documents.

- It provides a secure space to store and develop applications to support business operations.
- It improves the efficiency of the company by speeding up workflow and reducing errors. Thus, it helps achieve targets by completing the tasks on time.
- It offers a testing platform for new ideas before they are uploaded on the company's internet webpage. Thus, it helps maintain the credibility of the company
- Information is shared in real-time, or updates are reflected immediately to all the authorized users.
- Modern intranets also offer a mobile app that allows employees to stay connected on the go.
- It aids in project management and tracking workflow and teams' progress.
- It can work with mobile devices, which means it can provide information that exists on intranet directly to mobile devices of employees such as phones, tablets, etc.
- It can also be used to motivate employees, facilitate employee recognition, and to reward them for performing beyond expectations.

## How the Intranet Works:

Intranet basically comprises three components: a web server, an intranet platform, and applications. The web server is hardware that contains all the intranet software and data. It manages all requests for files hosted over the server and finds the requested files and then delivers it to the user's computer.



The intranet platform, which is software, allows communication tools, collaboration apps, and databases to work seamlessly with each other. It is tailored to the specific needs of a business.

The applications are required to enable users to work smoothly. They are the computing tools that allow users to do their work, communicate, and coordinate with each other and retrieve and store information.

Furthermore, the user who wants to access the intranet is required to have a special network password and should be connected to the LAN. A user who is working remotely can gain access to the intranet through a virtual private network ([VPN](#)) that allows them to sign in to the intranet to access the information.

## Disadvantages of Intranet:

- It may be costly to set up an Intranet due to hidden costs and complexity.
- If the firewall does not work properly or not installed, it can be hacked by someone
- High-security passwords are required, which cannot be guessed by outside users
- There is always a fear of losing control over the intranet
- Sometimes document duplication may happen which can cause confusion among employees
- You have to give access to multiple users, so you may find it hard to control this network.

## Examples of Intranet:

**Educational Intranet:** It is generally found in a school, college, etc., For example, a school intranet is intended to allow teaching staff to communicate with each other and get information about upcoming updates such as exam dates, schools functions, holidays, etc.

**Real Estate Intranet:** The intranet of a real estate company allows its sales team to have access to all important brochures, templates, forms that they may need to close a sale. Employees also remain up to date with important events like meetings, training, sessions, etc. It can also be used to share motivational messages with the team.

**Health Care Intranet:** In the healthcare sector, in big hospitals, the Intranet helps health care professionals to work as a team to provide proper care and treatment to their patients. Doctors can share reports, treatment procedures, bills and claims can be settled easily without moving from one department to another department.

**IT Sector Intranet:** In the [IT](#) sector there is always a lot of information that needs to be shared with all the employees at one go. It may be related to a project that needs



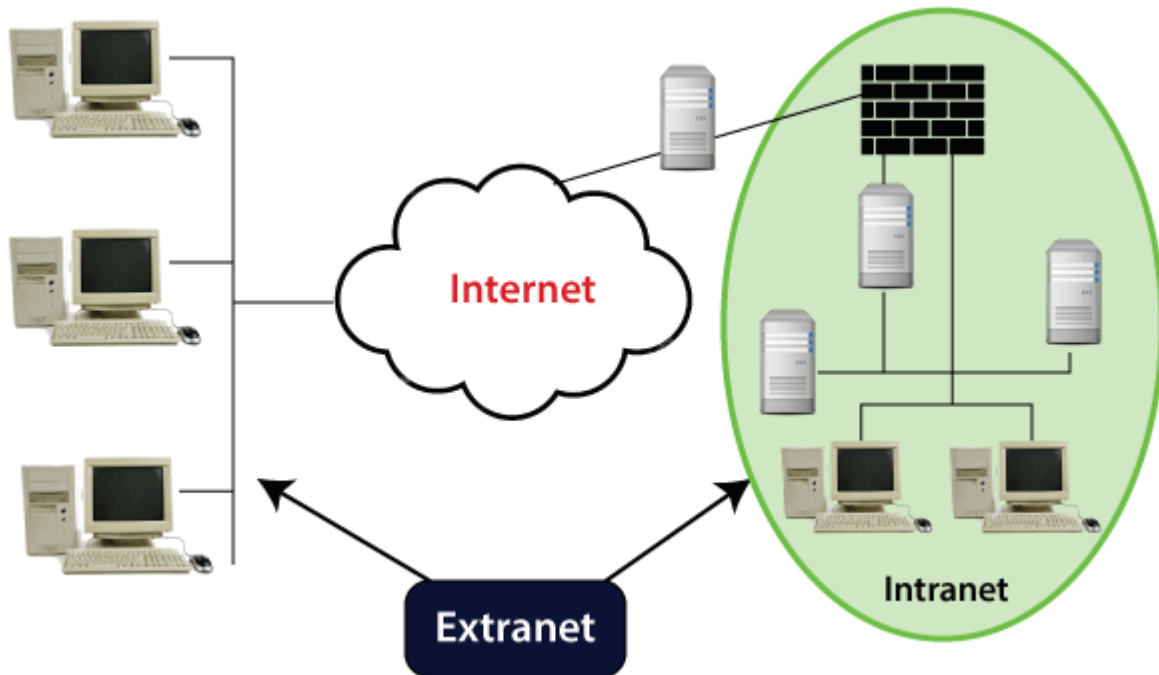
to be completed within the given time frame, such as guidelines, terms and conditions, and rules that are to be followed while working on a project.

## Difference between Intranet and Internet:

Internet	Intranet
It is a medium such as optical fiber cable that connects billions of computers with each other to establish a worldwide network.	It is a small, private network as it belongs to a specific organization.
It has billions of users as it is a public network with a worldwide presence.	It has limited users.
It is not as safe as an intranet.	It is a safer network than the internet.
It can be accessed or used by anyone using an internet-enabled devices, such as laptop, mobile phone, etc.	Only authorized persons can use this network.
It offers a wide range of information, such as news, blogs, websites, etc.	It offers limited information related to organization's work, policies, updates, etc.
It is not owned by a single person or an organization.	It can be owned by a person or an organization.

## Extranet

Extranet is a part of an organization's intranet. It is a communication network that is based on internet protocols (TCP/IP). It provides controlled access to firm's intranet to its trading partners, customers, and other businesses. So, it is a private network that securely shares internal information and operations of a firm with authorized people outside the firm without giving access to the company's entire network. The users are required to have IDs, passwords, and other authentication mechanisms to access this network.



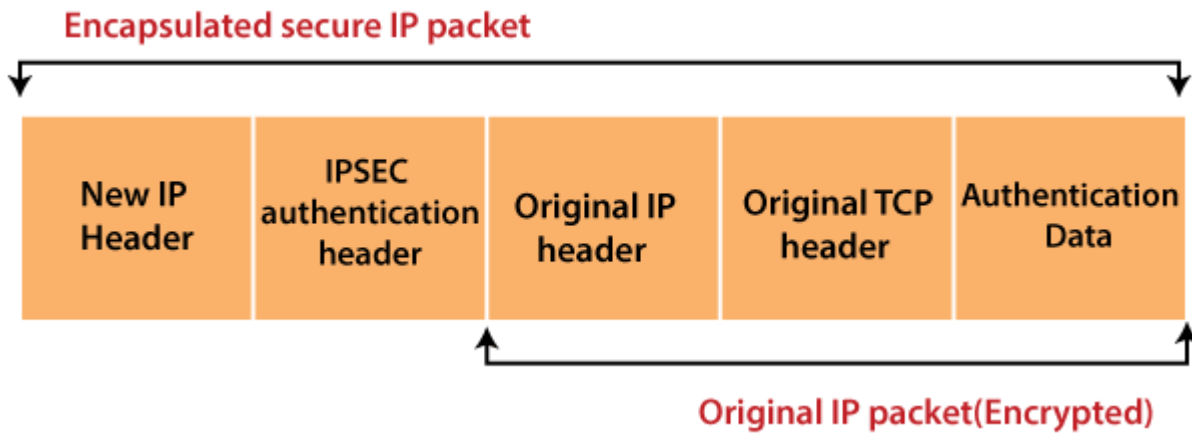
### Some of the benefits of extranet:

- It acts as a single interface between the company and its trading partners.
- It automates the firm's processes like automatically places an order with suppliers when inventory drops.
- It improves customer service by providing customers a platform to resolve their queries and complaints.
- It enables the firm to share information with trading partners without engaging in paper-based publishing processes.
- It streamlines business processes that are repetitive in nature, such as ordering from a vendor on a regular basis.

### How is Extranet Established?

It is set up in the form of a Virtual Private Network as it is prone to security threats due to the use of the **internet** to connect outsiders to an organization's intranet. VPN can assure you a safe network in a public network such as the internet. The transmission control protocol (TCP) and internet protocol (IP) are used for the data transfer.

**VPN** assures secure transactions based on Internet Protocol Security Architecture (IPSEC) protocol as it provides an extra security layer to **TCP/IP** protocol, which is used for data transfer in the extranet. In this layer, the **IP** packet is encapsulated to form a new IP packet, as shown below:



Furthermore, to provide more security to Intranet, the following two measures are also taken by an organization:

- **Firewall:** It prevents unauthorized users from accessing the extranet.
- **Passwords:** It also prevents unauthorized users, including the company's employees from accessing the data stored on its server.

### Limitations of Extranet:

- **Hosting:** If you host extranet pages on your own server, it requires a high bandwidth internet connection, which is may be very expensive.
- **Security:** You need extra firewall security if you host it on your own server. It increases the workload and makes security mechanism very complex.
- **Dependency:** It is dependent on the internet as outsiders cannot access information without using the internet.
- **Less Interaction:** It reduces the face to face interaction between customers, business partners, vendors, etc., which results in poor relationship building.

### Difference between Intranet and Extranet:

Intranet	Extranet
It is a private network, which cannot be accessed externally.	It may not be called a private network, as it can be assessed externally. It provides limited access to authorized outside-users such as vendors, partners, etc.
It connects the employees of the company.	It connects the company's employees with partners.
It is an independent network, not	It is an additional part of company's Intranet.

a part or extension of any other network.	
Communication takes place only within the organization that owns the network.	External users such as suppliers, customers, and partners are allowed to be a part of intranet to get information, updates, about the organization.

## What is World Wide Web?

World Wide Web, which is also known as a Web, is a collection of websites or web pages stored in web servers and connected to local computers through the internet. These websites contain text pages, digital images, audios, videos, etc. Users can access the content of these sites from any part of the world over the internet using their devices such as computers, laptops, cell phones, etc. The WWW, along with internet, enables the retrieval and display of text and media to your device.



The building blocks of the Web are web pages which are formatted in HTML and connected by links called "hypertext" or hyperlinks and accessed by HTTP. These links are electronic connections that link related pieces of information so that users can access the desired information quickly. Hypertext offers the advantage to select a word or phrase from text and thus to access other pages that provide additional information related to that word or phrase.

A web page is given an online address called a Uniform Resource Locator (URL). A particular collection of web pages that belong to a specific URL is called a website, e.g., *www.facebook.com*, *www.google.com*, etc. So, the World Wide Web is like a huge electronic book whose pages are stored on multiple servers across the world.

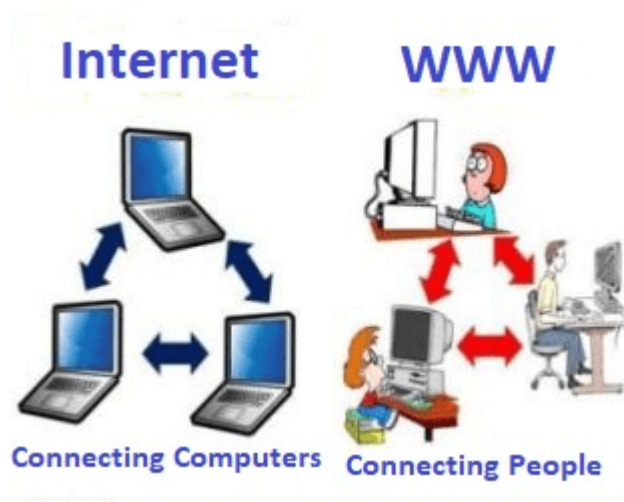
Small websites store all of their WebPages on a single server, but big websites or organizations place their WebPages on different servers in different countries so that

when users of a country search their site they could get the information quickly from the nearest server.

So, the web provides a communication platform for users to retrieve and exchange information over the internet. Unlike a book, where we move from one page to another in a sequence, on World Wide Web we follow a web of hypertext links to visit a web page and from that web page to move to other web pages. You need a browser, which is installed on your computer, to access the Web.

## Difference between World Wide Web and Internet:

Some people use the terms 'internet' and 'World Wide Web' interchangeably. They think they are the same thing, but it is not so. Internet is entirely different from WWW. It is a worldwide network of devices like computers, laptops, tablets, etc. It enables users to send emails to other users and chat with them online. For example, when you send an email or chatting with someone online, you are using the internet.



But, when you have opened a website like google.com for information, you are using the World Wide Web; a network of servers over the internet. You request a webpage from your computer using a browser, and the server renders that page to your browser. Your computer is called a client who runs a program (web browser), and asks the other computer (server) for the information it needs.

## History of the World Wide Web:



The World Wide Web was invented by a British scientist, Tim Berners-Lee in 1989. He was working at CERN at that time. Originally, it was developed by him to fulfill the need of automated information sharing between scientists across the world, so that they could easily share the data and results of their experiments and studies with each other.

CERN, where Tim Berners worked, is a community of more than 1700 scientists from more than 100 countries. These scientists spend some time on CERN site, and rest of the time they work at their universities and national laboratories in their home countries, so there was a need for reliable communication tools so that they can exchange information.

Internet and Hypertext were available at this time, but no one thought how to use the internet to link or share one document to another. Tim focused on three main technologies that could make computers understand each other, HTML, URL, and HTTP. So, the objective behind the invention of WWW was to combine recent computer technologies, data networks, and hypertext into a user-friendly and effective global information system.

## How the Invention Started:

In March 1989, Tim Berners-Lee took the initiative towards the invention of WWW and wrote the first proposal for the World Wide Web. Later, he wrote another proposal in May 1990. After a few months, in November 1990, along with Robert Cailliau, it was formalized as a management proposal. This proposal had outlined the key concepts and defined terminology related to the Web. In this document, there was a description of "hypertext project" called World Wide Web in which a web of hypertext documents could be viewed by browsers. His proposal included the three main technologies (HTML, URL, and HTTP).

In 1990, Tim Berners-Lee was able to run the first Web server and browser at CERN to demonstrate his ideas. He used a NeXT computer to develop the code for his Web server and put a note on the computer "*The machine is a server. Do Not Power It DOWN!!*" So that it was not switched off accidentally by someone.

In 1991, Tim created the world's first website and Web Server. Its address was info.cern.ch, and it was running at CERN on the NeXT computer. Furthermore, the first web page address was <http://info.cern.ch/hypertext/WWW/TheProject.html>. This page had links to the information related to the WWW project, and also about the Web servers, hypertext description, and information for creating a Web server.

## The Web Grows:

NeXT computer platform was accessible by a few users. Later, the development of 'line-mode' browser, which could run on any system, started. In 1991, Berners-Lee introduced his WWW software with 'line-mode' browser, Web server software and a library for developers.

In March 1991, it was available to colleagues who were using CERN computers. After a few months, in August 1991, he introduced the WWW software on internet newsgroups, and it generated interest in the project across the world. Graphic interface for the internet, first introduced to the public on 6 August 1991 by Tim Berners-Lee. On 23 August 1991, it was available to everyone.

## Becoming Global:

The first Web server came online in December 1991 in the United States. At this time, there were only two types of browsers; the original development version which was available only on NeXT machines and the 'line-mode' browser which was easy to install and run on any platform but was less user-friendly and had limited power.

For further improvement, Berners-Lee asked other developers via the internet to contribute to its development. Many developers wrote browsers for the X-Window System. The first web server, outside Europe, was introduced at Standard University in the United States in 1991. In the same year, there were only ten known web servers across the world.

Later at the beginning of 1993, the National Center for Supercomputing Applications (NCSA) introduced the first version of its Mosaic browser. It ran in the X Window System environment. Later, the NCSA released versions for the PC and Macintosh environments. With the introduction of user-friendly browsers on these computers, the WWW started spreading tremendously across the world.

Eventually, the European Commission approved its first web project in the same year with CERN as one of its partners. In April 1993, CERN made the source code of WWW available on a royalty-free basis and thus made it free software. Royalty-free means one has the right to use copyright material or intellectual property without paying any royalty or license fee. Thus, CERN allowed people to use the code and web protocol for free. The technologies that were developed to make the WWW became an open source to allow people to use them for free. Eventually, people started creating websites for online businesses, to provide information and other similar purposes.



At the end of 1993, there were more than 500 web servers, and the WWW has 1% of the total internet traffic. In May 1994, the First International World Wide Web conference was held at CERN and was attended by around 400 users and developers and popularly known as the "Woodstock of the Web." In the same year, the telecommunication companies started providing internet access, and people have access to WWW available at their homes.

In the same year, one more conference was held in the United States, which was attended by over 1000 people. It was organized by the NCSA and the newly-formed International WWW Conference Committee (IW3C2). At the end of this year (1994), the World Wide Web had around 10000 servers and 10 million users. The technology was continuously improved to fulfill growing needs and security, and e-commerce tools were decided to be added soon.

## Open standards:

The main objective was to keep the Web an open standard for all rather than a proprietary system. Accordingly, CERN sent a proposal to the Commission of the European Union under the ESPRIT program "WebCore." This project's objective was to form an international consortium in collaboration with Massachusetts Institute of Technology (MIT), the US. In 1994, Berners-Lee left CERN and joined MIT and established the International World Wide Web Consortium (W3C) and a new European partner was needed for W3C.

The European Commission approached the French National Institute for Research in Computer Science and Controls (INRIA), to substitute the CERN's role. Eventually, in April 1995, INRIA became the first European W3C host and in 1996 Keio University of Japan became another host in Asia.

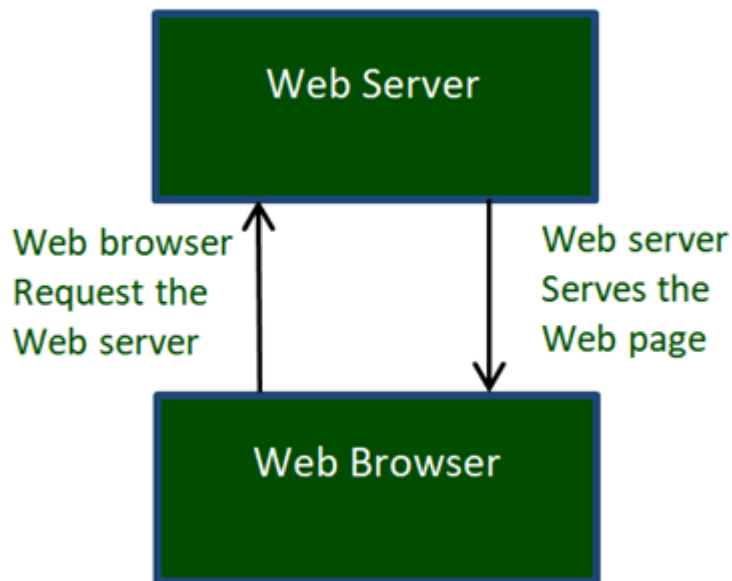
In 2003, ERCIM (European Research Consortium in Informatics and Mathematics) replaced INRIA for the role of European W3C Host. Beihang University was announced as the fourth Host by W3C in 2013. In September 2018, there were over 400 member organizations around the world.

Since its inception, the Web has changed a lot and is still changing today. Search engines have become more advanced at reading, understanding, and processing information. They can easily find the information requested by users and can even provide other relevant information that might interest users.

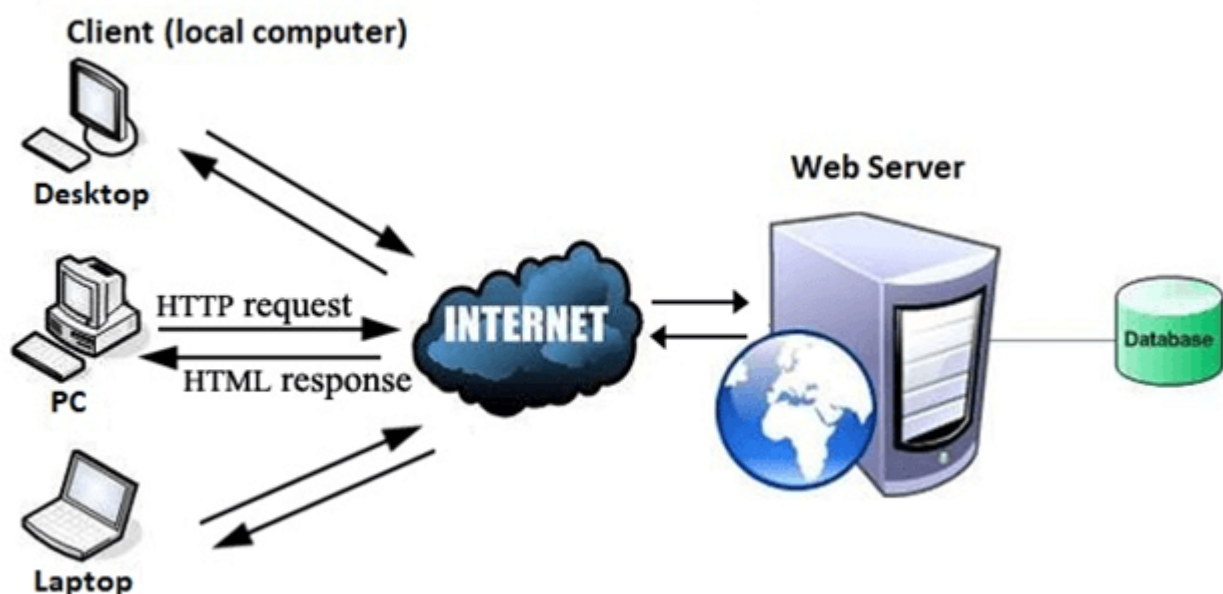
## How the World Wide Web Works?

Now, we have understood that WWW is a collection of websites connected to the internet so that people can search and share information. Now, let us understand how it works!





The Web works as per the internet's basic client-server format as shown in the following image. The servers store and transfer web pages or information to user's computers on the network when requested by the users. A web server is a software program which serves the web pages requested by web users using a browser. The computer of a user who requests documents from a server is known as a client. Browser, which is installed on the user' computer, allows users to view the retrieved documents.



All the websites are stored in web servers. Just as someone lives on rent in a house, a website occupies a space in a server and remains stored in it. The server hosts the

website whenever a user requests its WebPages, and the website owner has to pay the hosting price for the same.

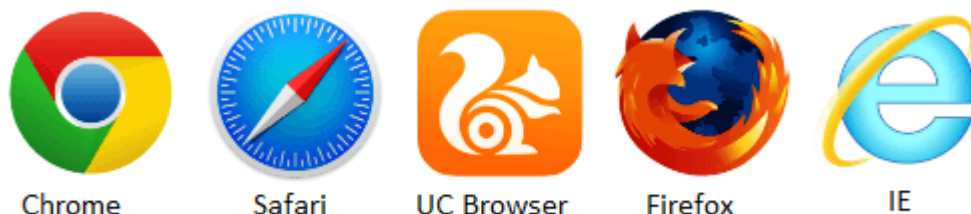
The moment you open the browser and type a URL in the address bar or search something on Google, the WWW starts working. There are three main technologies involved in transferring information (web pages) from servers to clients (computers of users). These technologies include Hypertext Markup Language (HTML), Hypertext Transfer Protocol (HTTP) and Web browsers.

## Hypertext Markup Language (HTML):



HTML is a standard markup language which is used for creating web pages. It describes the structure of web pages through HTML elements or tags. These tags are used to organize the pieces of content such as 'heading,' 'paragraph,' 'table,' 'Image,' and more. You don't see HTML tags when you open a webpage as browsers don't display the tags and use them only to render the content of a web page. In simple words, HTML is used to display text, images, and other resources through a Web browser.

## Web Browser:

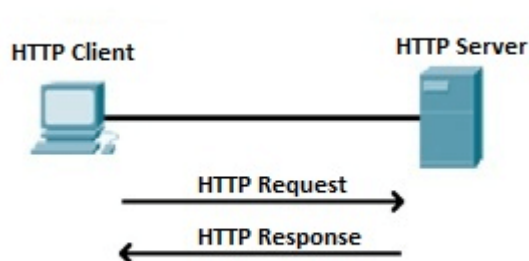


A web browser, which is commonly known as a browser, is a program that displays text, data, pictures, videos, animation, and more. It provides a software interface that allows you to click hyperlinked resources on the World Wide Web. When you double click the Browser icon installed on your computer to launch it, you get connected to the World Wide Web and can search Google or type a URL into the address bar.

In the beginning, browsers were used only for browsing due to their limited potential. Today, they are more advanced; along with browsing you can use them for e-mailing, transferring multimedia files, using social media sites, and participating in online discussion groups and more. Some of the commonly used browsers include Google Chrome, Mozilla Firefox, Internet Explorer, Safari, and more.

## Hypertext Transfer Protocol (HTTP):

Hyper Text Transfer Protocol (HTTP) is an application layer protocol which enables WWW to work smoothly and effectively. It is based on a client-server model. The client is a web browser which communicates with the web server which hosts the website. This protocol defines how messages are formatted and transmitted and what actions the Web Server and browser should take in response to different commands. When you enter a URL in the browser, an HTTP command is sent to the Web server, and it transmits the requested Web Page.



When we open a website using a browser, a connection to the web server is opened, and the browser communicates with the server through HTTP and sends a request. HTTP is carried over TCP/IP to communicate with the server. The server processes the browser's request and sends a response, and then the connection is closed. Thus, the browser retrieves content from the server for the user.

## Search Engine

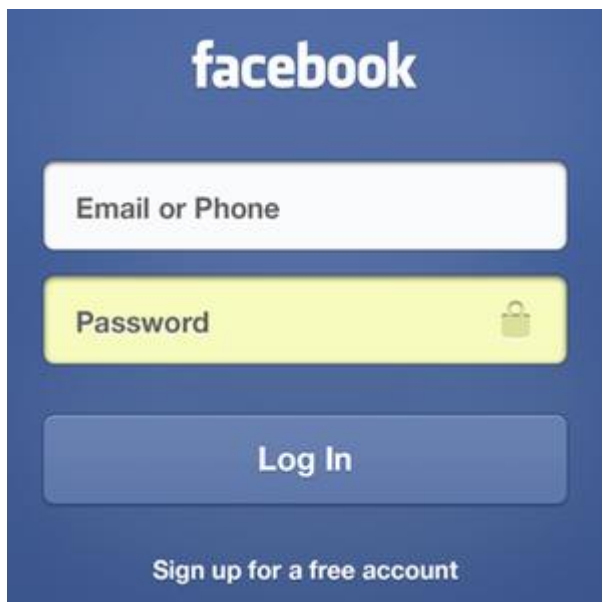
The Search engine is a program which is designed to enable the users to browse information or content on World Wide Web. It helps retrieve the desired information in minimum time. It allows you to input specific keywords or phrases and retrieves a list of items matching those keywords and phrases. Thus, it does not provide information straight away; it just retrieves pages which are related to keywords or other search terms. Some of the popular search engines are Google, Bing, and Yahoo! Search.



---

## Portal

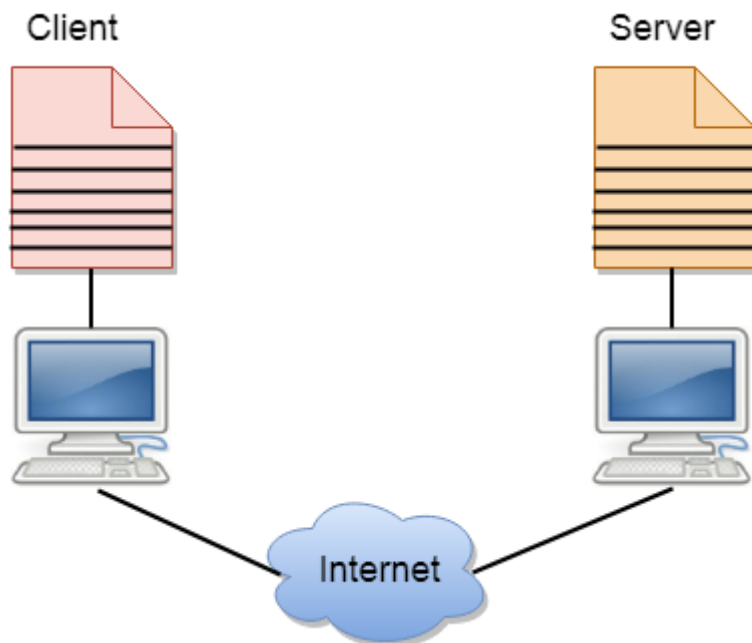
Portal is a private location on the internet which acts as a point of access to the information available on the World Wide Web. A portal is accessed through a unique URL, unique username and password, i.e. apart from URL, personal login is required to see the content on a portal. Some of the popular portals are facebook.com, gmail.com and twitter.com.



## Client and Server model

- A client and server networking model is a model in which computers such as servers provide the network services to the other computers such as clients to perform a user based tasks. This model is known as client-server networking model.

- The application programs using the client-server model should follow the given below strategies:



- An application program is known as a client program, running on the local machine that requests for a service from an application program known as a server program, running on the remote machine.
- A client program runs only when it requests for a service from the server while the server program runs all time as it does not know when its service is required.
- A server provides a service for many clients not just for a single client. Therefore, we can say that client-server follows the many-to-one relationship. Many clients can use the service of one server.
- Services are required frequently, and many users have a specific client-server application program. For example, the client-server application program allows the user to access the files, send e-mail, and so on. If the services are more customized, then we should have one generic application program that allows the user to access the services available on the remote computer.

## Client

A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

## Server

A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.

A server program is an infinite program means that when it starts, it runs infinitely unless the problem arises. The server waits for the incoming requests from the clients. When the request arrives at the server, then it responds to the request.

## Advantages of Client-server networks:

- **Centralized:** Centralized back-up is possible in client-server networks, i.e., all the data is stored in a server.
- **Security:** These networks are more secure as all the shared resources are centrally administered.
- **Performance:** The use of the dedicated server increases the speed of sharing resources. This increases the performance of the overall system.
- **Scalability:** We can increase the number of clients and servers separately, i.e., the new element can be added, or we can add a new node in a network at any time.

## Disadvantages of Client-Server network:

- **Traffic Congestion** is a big problem in Client/Server networks. When a large number of clients send requests to the same server may cause the problem of Traffic congestion.
- It does not have a robustness of a network, i.e., when the server is down, then the client requests cannot be met.
- A client/server network is very decisive. Sometimes, regular computer hardware does not serve a certain number of clients. In such situations, specific hardware is required at the server side to complete the work.
- Sometimes the resources exist in the server but may not exist in the client. For example, If the application is web, then we cannot take the print out directly on printers without taking out the print view window on the web.

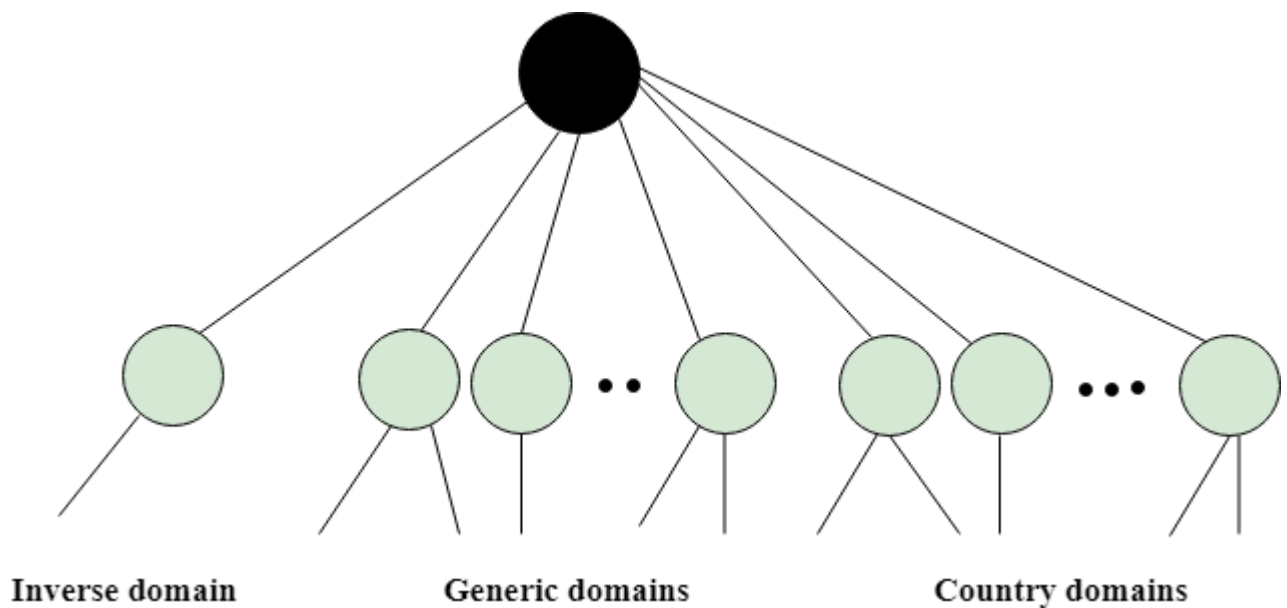
## DNS

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.

- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.

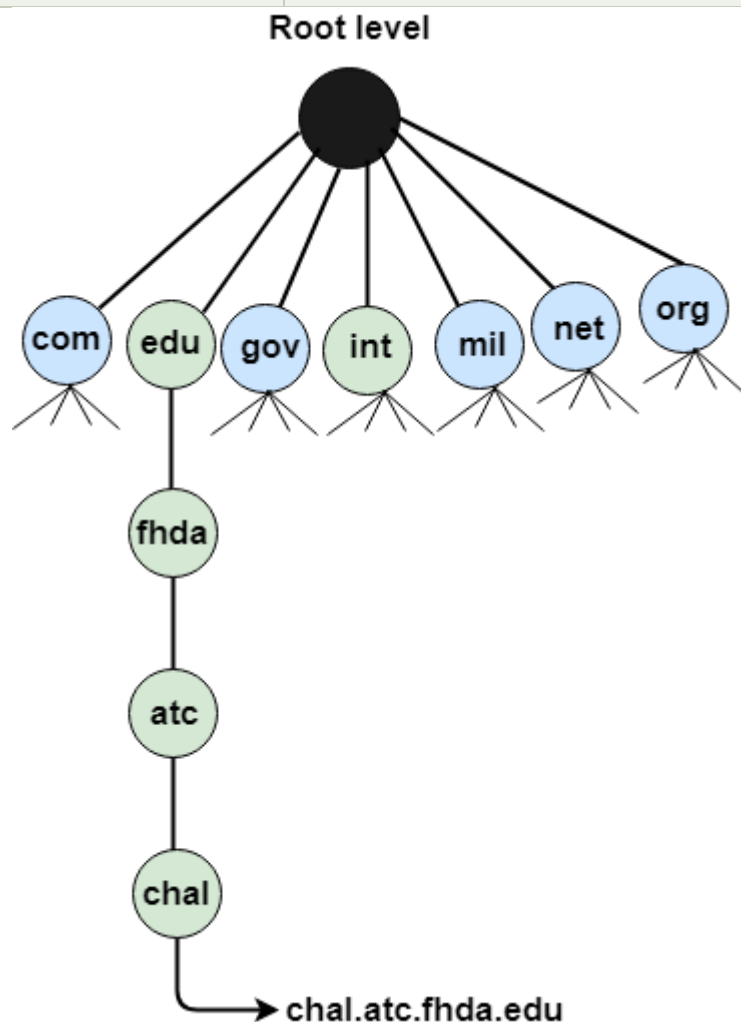


## Generic Domains

- It defines the registered hosts according to their generic behavior.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, and these labels describe the organization type.

Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms
com	Commercial Organizations
coop	Cooperative business Organizations
edu	Educational institutions

gov	Government institutions
info	Information service providers
int	International Organizations
mil	Military groups
museum	Museum & other nonprofit organizations
name	Personal names
net	Network Support centers
org	Nonprofit Organizations
pro	Professional individual Organizations



## Country Domain

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.



## Inverse Domain

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

## Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as **DNS resolver** sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

## What is the URL?

A URL is a type of uniform resource identifier and is address of a resource on the World Wide Web and the protocol used to access it. It is used to indicate the location of a web resource to access the web pages. For example, to visit the javatpoint website, you will go to the URL [www.javatpoint.com](http://www.javatpoint.com), which is the URL for the [javatpoint website](http://www.javatpoint.com).



The URL sends users to a specific resource online such as video, webpage, or other resources. When you search any query on Google, it will display the multiple URLs of

the resource that are all related to your search query. The displayed URLs are the hyperlink to access the webpages.

A [URL](#) (Uniform Resource Locator) contains the information, which is as follows:

- The port number on the server, which is optional.
- It contains a protocol that is used to access the resource.
- The location of the server
- A fragment identifier
- In the directory structure of the server, it contains the location of the resource.

The additional information about the URL is described below with the help of an example:

Let's take an example: <https://www.javatpoint.com/jtp.htm>, it indicates the [jtp.htm](#) is a file located on the server with the address of [javatpoint.com](#).

### **[http://](#) or [https://](#)**

The [http](#) is a protocol that stands for Hypertext Transfer Protocol. It tells the browser to which protocol will be preferred to use for accessing the information that is specified in the domain.

The [https](#) (Hypertext Transfer Protocol Secure) is an enhanced protocol as compared to [http](#) as it concerned with security. It provides the surety that the information, which is transmitted over [HTTP](#) is secure and encrypted. The colon (:) and two forward slashes (//) are used to separate the protocol from the rest of the part of the URL.

### **[www.](#)**

The [www](#) is used to distinguish the content, which stands for World Wide Web. This portion of the URL can be left out many times, as it is not required. For instance, if you type "[http://javatpoint.com](#)," you will still get the [javatpoint](#) website. For an important subpage, this portion can also be substituted, which is known as a subdomain.

### **[javatpoint.com](#)**

The [javatpoint.com](#) is the domain name for the website, and the [.com](#) is called TLD or suffix. It helps to identify the location or type of the website. For example, ".org" stands for an organization, ".co.uk" stands for the United Kingdom, and ".com" is for commercial. There are various types of domain suffixes available; you are required to register the name through a domain registrar to get a domain.

## jtp.htm

The jtp.htm is the name of the web page, and the .htm is the file extension of the web page, which describes the file is an [HTML](#) file. There are many other file extensions available on the internet such as .php, .html, .xml, .jpg, .gif, .asp, .cgi, etc.

## Where is the URL located?

A URL is located in the address bar or search bar at the top of the browser window. The URL is always visible in the desktop computers and laptop unless your browser is being displayed in full screen. In most of the smartphones and tablets, when you scroll down the page, the URL will disappear and only show the domain when visible. To visible the address bar, you need to scroll up the page. And, if only the domain is shown and you want to see full address, tapping on the address bar to show the full address.

## What characters cannot be used in the URL?

It is realized by many people that space is not allowed in a URL. The URL string can contain only symbols `! $-_*'()`, including **alphanumeric characters** as it is documented in RFC 1738. Any other characters must be encoded in the URL if needed.

## Is an IP address the same as a web address or a URL?

An [IP](#) address is not the same as a web address or a URL, as it is a unique number that is assigned to each device on a network. A domain name is assigned a unique [IP](#) address on the World Wide Web, and when entered an URL like javatpoint.com, it is translated by [DNS](#) into an IP address that used by routers to find web servers. Instead of using an IP address, a domain name is used as it is easy to remember by humans. For example, to remember an IP address like 216.58.216.164 is hard, and much easier to remember 'javatpoint.com.'

## Understanding more complex URLs and parameters

A URL performs additional functions and added the parameters (additional information) to the end of the URL when it points to a script. For example, when you search any query on any search engine, it points to a search results page, including the additional information with the search query words.

An URL example is given below that points to the javatpoint search page, including the search query parameter of example search.

1. <https://www.javatpoint.com/cgi-bin/search.cgi?q=example%20search>

In cgi-bin directory, the script file being pointed to is search.cgi in the above URL example. It is assumed to be a Perl script as this file ends with .cgi.

The file name is a question mark (?) after the script. In the URL, the question mark separates the URL from all the variables or parameters to be sent to the script. The parameter being sent is `q=example%20search`, in the above URL example. The "example%20search" is the value that is sent to the "q" variable. Space is encoded as %20 as spaces are allowed in a URL. Furthermore, a+ is also used to represent space in many scripts.

There is a variable in the example that is executed, as the script uses it. Also, scripts can contain multiple variables; each variable is separated by a symbol & (ampersand), as shown in the following example:

1. <https://www.javatpoint.com/cgi-bin/search.cgi?q=example%20search&example=test>

The above example contains two different variables; the q" variable equals "example search" and the "example" variable equals "test."

## Why URL?

- The URL is beneficial, as the written information in the URL provides users the option to switch from one web page to another by clicking only one mouse click.
- Every URL is unique and tells users how to access a specific resource.
- When a user types a URL into the web browser and opens any hyperlink from search results, the browser forwards a request to a webserver to fetch files related to the search query.
- A website domain or URL identifies one particular file, and it is the most important part of your website. Usually, by using words that end with .net, .com, or .org, you can get traffic on your website.

## What is URL Redirect?

A URL redirect is a web server function that takes your URL and points it to another. For example, consider, you had the old URL "myvlogsite.com," and you wanted to visitors' access directly to the new URL "javatpoint.com." Redirect is the best solution for it; when anyone type "myvlogsite.com" in the browser would be redirected to the new URL "javatpoint.com." There are various kinds of redirects for web developers, such as HTTP 3xx series status codes, manual redirects, JavaScript, metatag refreshes, server-side scripts, frame redirects, and more. Furthermore, the URL redirect may also be known as URL forwarding, domain forwarding, HTTP code 3xx redirect, and domain redirection.

There are many reasons for web users may be redirected from one URL to another, such as follows:

- Merging of two websites
- Change of business name

- To direct content to a recently updated domain name
- Landing page-split testing for marketing tests
- To direct traffic toward recently updated content

A URL redirect is also used to cause problems for users and their computers through illegal activities like phishing. Additionally, it can be used to remove the search results of web browsers, but nowadays, most of the search engines are capable of detecting these types of fraud attempts. Redirect a web page, the several HTTP protocol 3xx series codes are the most common way. The members of this series have various attributes, such as follows:

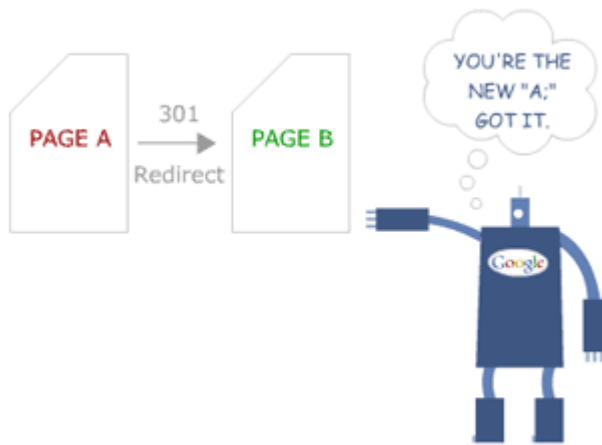
- The number 300 provides various redirect choices. For instance, an option to select alternative languages.
- The number 301 indicates when a site is moved permanently. For example, when the name of a business has changed.
- The number 302 is used for an unspecified redirect.
- 303 display the output of CGI (common gateway interface) scripts.
- 307 is used, when a site is to be redesigned.

The website address obtains a new URL when a website visitor is redirected to a newly named website URL. Businesses often change their website's homepage into a redirect page. They transform page with a concurrent message briefly describing the redirect. A meta tag is embedded into the website's source code behind the scenes. The regular visitors of the website will receive an error message "404 - Not Found" without a redirect.

## **Different types of redirects**

### **1. 301 Redirect**

It is a permanent type of unmasked redirect that instructs web browsers to move from one site destination to another automatically. It is one of the most common and searches engine-friendly method for implementing redirects. It should be used when your website was permanently moved to the new address.



2. The redirect can also be used in some programming languages like PHP; programmers can use a canonical 301 redirect to perform a change for many pages in a domain. Furthermore, the 301 redirect passes over 90% of the link juice; thus, it is also beneficial for SEO purposes.

### 3. **302 Redirect**

It is a temporary type of unmasked redirect and not widely used. It is a name for an [HTTP](#) status code that is used when a certain URL has been changed temporarily to a different address. Search engines will not index the destination URL, index the original URL, and display it in search results. The browser is redirected from one URL to another with the help of 302 redirects. Additionally, it is characterized as a permanent redirect and based on a different HTTP status code. In many cases, it can return a cleaner and simpler URL for users. To use 302 redirects, other technologies and different search engines have their own specific strategies.

### 4. **303 Redirect**

A 303 redirect is also known as HTTP 303 that is a response to an HTTP status code. It is a specific type of redirect as a response to a request for a URI (Unified Resource Identifier). It also has its own syntax; the W3C specifies to use a GET method to access the desired destination if a request for a different URI.

## When should be used a redirect?

### 1. You have duplicate content

Duplicate content is that it appears more than once on the page. There are multiple pages on Google that contain duplicate content. In this situation, it is difficult for Google to understand which page is the correct one. You can use a 301 redirect on the duplicate piece of content to direct to the original page. It will create a better experience for your users and help to improve your search engine rankings.

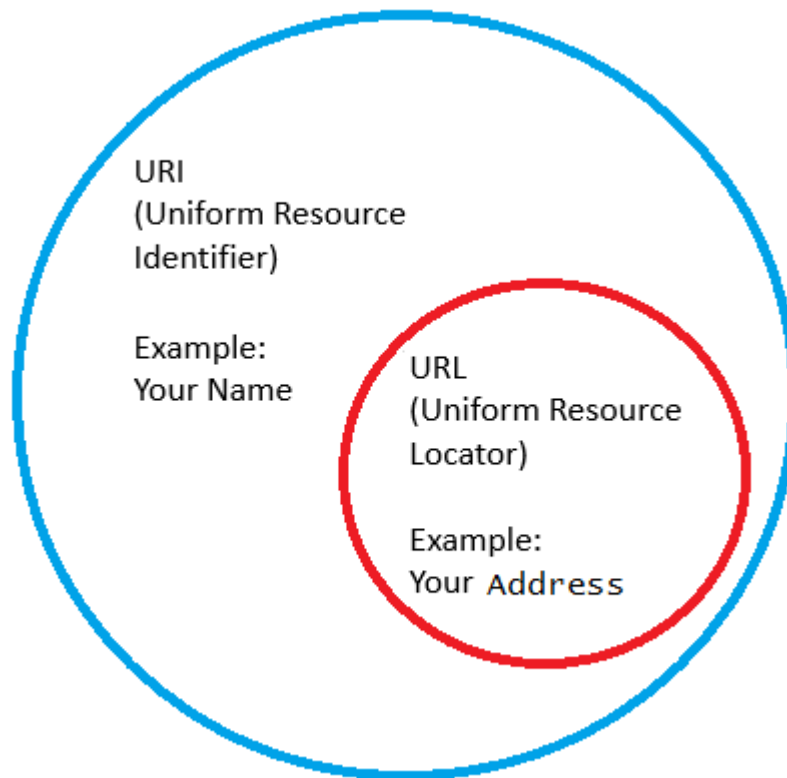
### 2. You have changed your domain

The use of redirect is useful when you are making change your domain name and probably do not want to lose any built links.

### 3. You have multiple domains

To protect the online brand, some people purchase multiple domain names. So, they will need to redirect any of the old domain to the new domain. Many companies do this to gain additional traffic from common misspellings. Also, they can prevent competitors from buying a similar domain and can redirect them to their own site.

### Difference between URL and URI



There are numerous differences between URL and [URI](#), which are as follows:

URL	URI
URL stands for Uniform Resource Locator that used to describe the identity of an item.	URI stands for Uniform Resource Identifier, which offers a technique for defining the identity of an item.

The primary objective of the URL is to get the address or location of the resource.	The primary objective is to find a resource and distinguish it from other resources with the help of a name or location.
URL is a type of URI; therefore, all URLs can be URIs.	URI is the superset of URL; thus, all URIs are not URLs as a URI can be a name rather than a locator.
It is only used for locating web pages.	It is used in various languages such as HTML, XML and other files XSLT, and more.
A URL specifies where a resource is occurring and a way for retrieving a resource.	A URI identifies a resource either by URL or URN or both.
In URL, the scheme must be a protocol such as FTP, HTTP, HTTPS, and more.	The scheme may be anything in URI like a name, specification, protocol, and more.
It contains the protocol information in the URL.	It does not include protocol information.
It includes components like path, domain, hash, string, query, and more.	It includes components such as path, scheme, query, fragment component, and more.
It offers specification on what type of protocol is to be used.	It does not contain protocol specification.
An example of URL: <a href="https://google.com">https://google.com</a>	An example of URI: urn:isbn:0-486-27557-4

## What is a Webpage

*A web page is a single hypertext document available on World Wide Web (WWW). It is composed of HTML elements and displayed on the user's browser such as [Mozilla](#), [Firefox](#), [Chrome](#), etc. It is also referred to as "Page."*

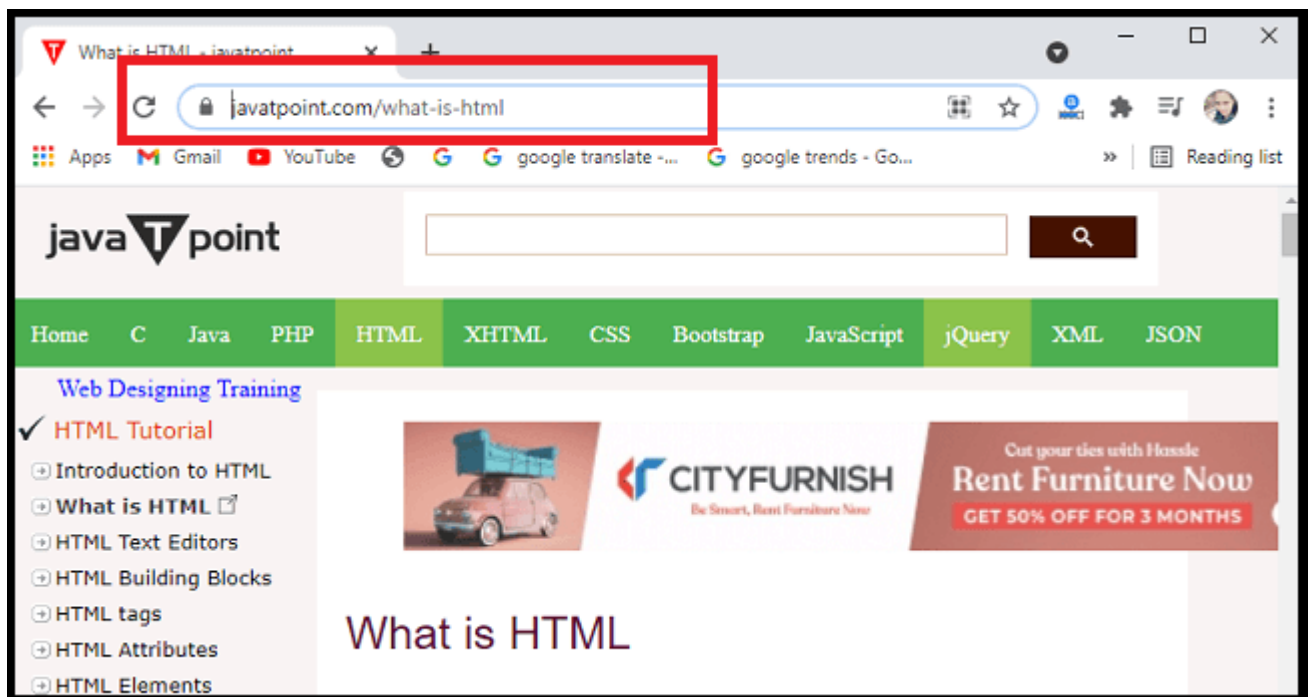


In this topic, we are going to discuss various details of the webpage, including the following topics:

- [What is a Webpage?](#)
- [Characteristics of a Webpage](#)
- [Difference between a Webpage and a Website](#)
- [How does a Web Page Work?](#)
- [Elements of a Webpage](#)
- [Types of a Web page](#)
- [How to Create a Simple Webpage?](#)

## What is a Webpage?

A webpage is a document written in HTML and can be viewed on any web browser. It is contained within the web server, which can be accessed by entering the URL for that web page, and once it is loaded, it appears on the user's web browser. Each webpage is linked with a unique URL; hence two pages cannot have the same URL.



A webpage may contain **text, links for other pages, graphics, videos, etc.** Moreover, it is mainly used to provide information to the user in text, images, etc.

*A webpage is a part of a website; it means a website contains different web pages.* Such as **javaTpoint.com** is a website, and the page currently you are accessing is the webpage. It can be understood as an example of a book. So, a Website is like a complete book, and a webpage is like a page of that book.

The WWW or Internet contains millions of web pages, and daily, a lot is being added. Tim Berners-Lee developed **the first webpage**.

Let's understand some basic terms that are used with Webpage:

- **WebSite:** A website is a collection of several web pages. These pages are linked together with hyperlinks. A website has a unique domain name, and we can access it by entering that domain name in the URL.
- **Search Engine:** A search engine is an internet service that helps users find any information available on the internet. Some examples of search engines are **Google, Yahoo, Bing**. It is usually accessed with the help of Web browser.
- **Web Browser:** A web browser or simply browser is application software used to access the internet. Some examples of Web browsers are **Google Chrome, Microsoft Internet Explorer, Safari, etc.** It does two things:
  - It connects to a web server on the internet and requests a page that the user wants to view; once it finds that page, it displays it on its device.
  - It can interpret the set of HTML tags within a page to display the page in the correct format.
- **Webserver:** A web server can be understood as a computer that hosts or provide a website on the internet. It contains webserver software and component files of a website such as **HTML document, images, CSS stylesheet, and JS files.**

*Note: For practice, you can create web pages on your own without the need for a web server, and your browser will display those pages on your machine only.*

- **HTML:** HTML is an abbreviation of **Hyper-Text Markup Language**. A markup language is a computer language that specifies how a page should be formatted. With the help of HTML, one can specify fonts, colors, images, headings, and everything that he wants to display on a page displayed by the browser.

*Note: A web browser can also display other documents such as a PDF document or images, but only an HTML document is referred to as Web page.*

## Characteristics of a Web Page

Following are some characteristics of a Web page:

- A simple webpage can be created very quickly.
- It takes very little time to create a webpage compared to a Website.
- A web page and a website should be compatible with any device, such as Mobile, Desktop, Laptop, etc.
- The search engine provides a web page through a link, and when a user clicks on that link, it is redirected to the webpage of a website.
- A webpage can have any type of information including videos, and audios.

- It can be made up of only HTML(Hypertext Markup Language), or CSS, or JavaScript for dynamic and attractive behavior.

## Difference between a Webpage and a Website

Since both Websites and Web pages are related to each other, some users may use them interchangeably, but they are much different from each other. The basic difference between them is that **webpage is a single web document, whereas a Website is a collection of different web pages**. Here are some more differences between both of them:

Website		Webpage
A website is a collection of different web pages that are linked together with hyperlinks.		A webpage is a single hypertext document.
It consists of more than one webpage.		It is a single document that is displayed on the user's browser.
To develop a website, developers need more skills and more time compared to a webpage.	To develop a webpage, developers need basic HTML knowledge and less time.	
A website is accessed through its domain name, and it does not include any extension in the URL.		A webpage is accessed through a unique URL with some extension.
It can contain information for different entities, such as Javatpoint.com, which contains information about different technologies.		It can contain information for a single entity, such as currently viewing a web page containing information about this page only.
It is a little challenging to create a perfect website and requires lots of programming.		It is very simple to create a webpage.
Some examples of the website are Javatpoint.com, Amazon.com, etc.		Some examples of Webpages are the currently viewing page, contact page, registration page, the

	home page, etc.
--	-----------------

*Note: The terms Webpage and Web page are the same, and both are technically correct. However, most style guides suggest using a Webpage instead of a Web page.*

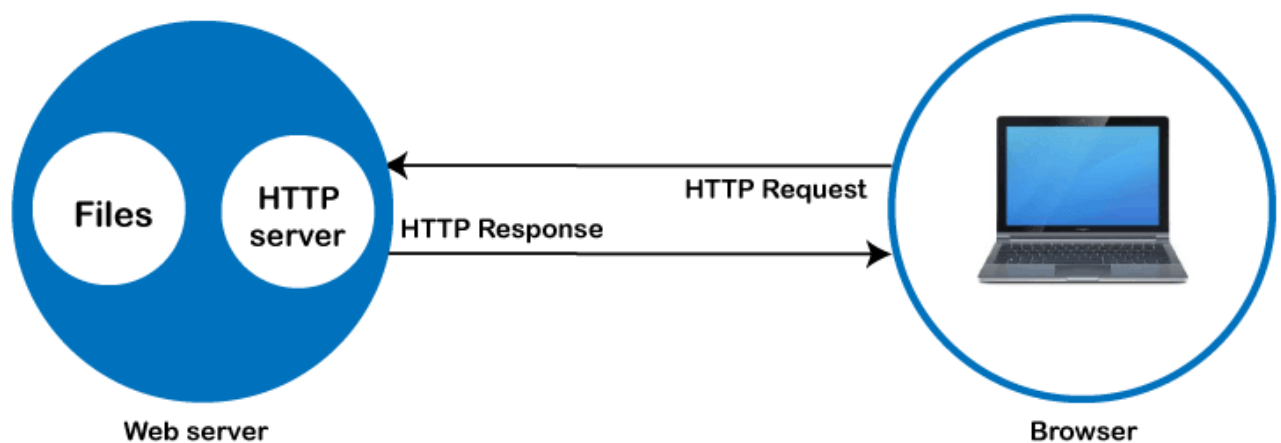
## How does a Web Page Work?

A simple web page is created using HTML, which is a markup language. However, we can also use CSS and JavaScript to add more functionalities and make it more attractive.

It is created using HTML, hence containing different markup tags that specify how the data should be formatted on screen.

The webpage is contained within the webserver. To load this webpage, a client sends the request to the server, and generally, the browser is known as the client, which can request the page on the internet.

The web browser requests the page on the internet. Once it is responded to by the server, the browser interprets the markup tags and displays them on the user's screen in the correct format.



The browser sends the request for a page or a file via **an HTTP request**. The HTTP is the **Hypertext Transfer Protocol**, a network protocol that allows transferring hypermedia documents over the internet between a browser and server.

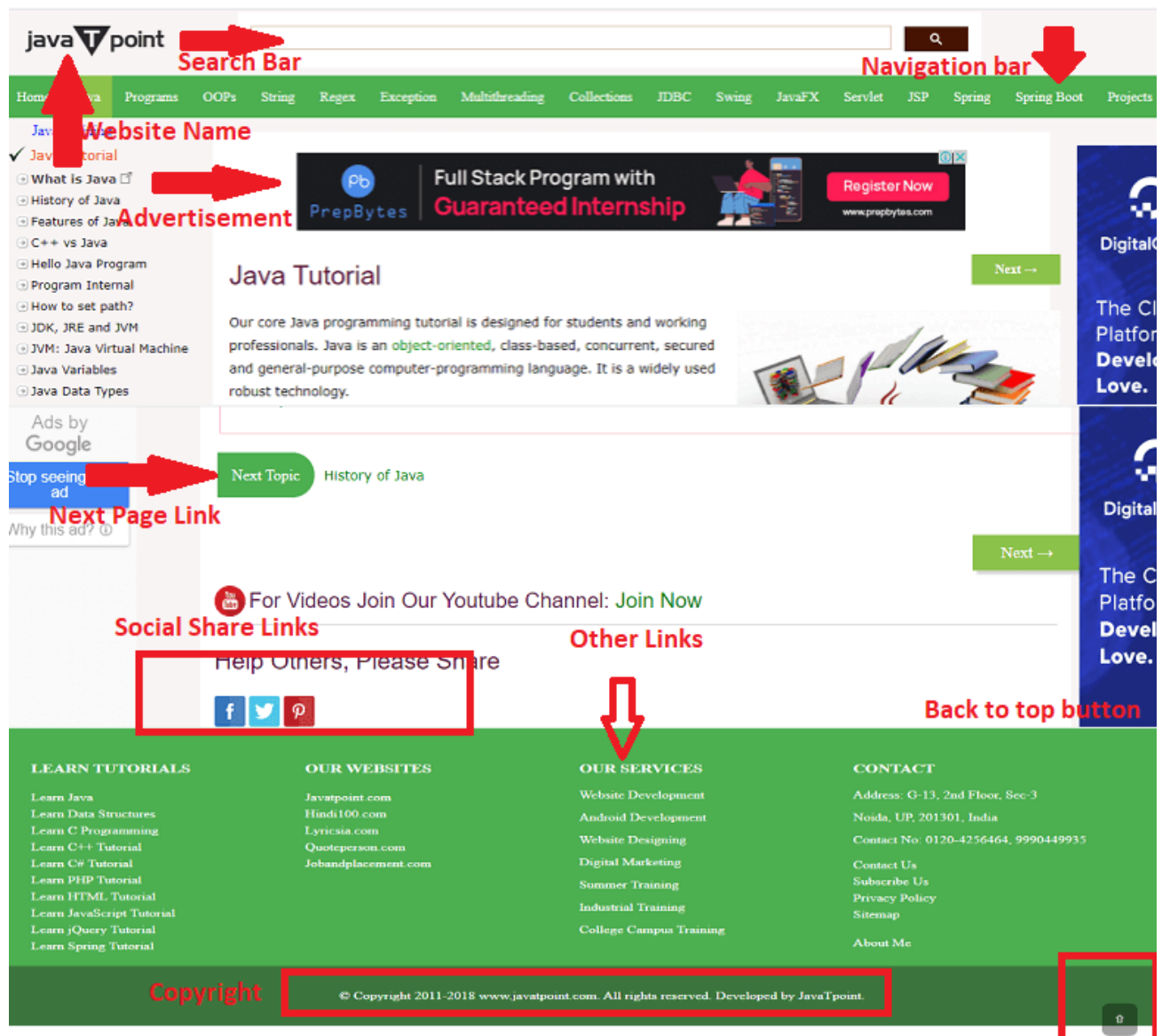
Once the request reaches the server, the HTTP server accepts the request, finds the requested page, and sends it back to the browser through **the HTTP response**. If a server is unable to find the requested page, it returns a **404 response**.

## Elements of a Webpage

The main element of the webpage is a text file composed of [HTML](#). Apart from this, a webpage can also have the following elements:

- **CSS:** The CSS code is used to make the page more interactive and control its look and feel.
- **Scripts:** The JavaScript code is included in a webpage to add interactivity to the page and add more functionalities to it.
- **Media:** It is used to include media components such as audio, video, and images.

Although every web page is different from another web, some components are common to almost all the pages. Some of these components are given below; you can also relate these elements by the given image:



1. **Name of the Website:** Each webpage includes the name of the website or company, or blog to which it is attached. The name of the website and the logo are mostly situated at the top-left corner of the page. The logo may also contain a slogan of the site or a brief introduction to the site so that visitors can quickly identify what this site is about. It is one of the important components of the webpage. The website's name also has a link that can redirect to the home page of that site. ***The name of the website usually includes at the header of the page.***

2. **Search bar:** A search bar is also an important component that should present on each page of a website or blog. The search bar allows the visitor to search related information on that website.
3. **Navigation Bar:** A navigation bar is a component of a webpage that contains links to some important sections of the website. It helps the visitors to easily traverse some major sections of the website. It is placed mainly on the top of the web page or the left side of the page. When the user clicks on any link given in the navigation bar, it redirects to the page.
4. **Heading of the page:** Heading of the page tells the main information about the page, i.e., what this is all about. The heading is available on the top of the page, and it is included with the help of the **<h1> tag** of HTML.
5. **Content of the Page:** The content of the page means the information of the page. As you are visiting this page and reading this information, all the information contained within this page is known as the page's content. It may contain below sub-elements:
  - **Paragraphs:** A webpage can have different paragraphs as per their length. The opening paragraph is crucial on the whole page, as it draws the attention of the visitor. If the first paragraph is not interesting and not related to the topic, a user may leave the page immediately. To create a **paragraph <P> tag is used in HTML.**
  - **Subheadings:** A page may have different subheadings as per the topic, whether related to information about something or a website's web page. In HTML from **<H2>** to **<H6>** tags are used for including subheadings. Each page should be divided into different subheadings to make it easier for the users to read and understand.
  - **Images:** Each webpage contains images to make its content more attractive. To include an image, **<img>** tag is used in HTML.
6. **Feedback or Comment form:** Different websites include a Feedback or Comment form on each webpage. It is used to know the visitor's views about the information of that page and any other feedback for the page or site. It lets to know the creator of the page that if the information is helpful or not.
7. **Social Share links:** Social share links allow the visitors to share that webpage with their friends on different social sites such as Facebook, Linked In, Twitter, etc.
8. **Copyright Info:** Each webpage must have Copyright information along with Privacy Policy. This information is available at the end of the page or on the footer of the page.
9. **Go to Top:** On most of the pages, a Back to Top button or link is available. This option helps the users to return to the upper section of the page.

10. **Advertisement Banner:** Whenever you visit any good website, you see different advertisement banners on each site's page. These banners are used to display ads in different places on a page. These banners are used to provide monetary benefits to the owner of the website.
11. **Previous and Next buttons or links:** On most of the pages, previous and Next buttons are available that help the user easily navigate to the previous and next page related to the topic. A page can also have links to other pages.
12. **Company Info:** On the footer of the page, there is a brief introduction about the company or website of that page. By this, visitors get to know about the company and can build trust in it.

There can also be some additional information and tools such as a button to print the page that can also be helpful for users.

## Types of a Web page

There are mainly two types of a Web page based on functionality:

- **Static Webpage**
- **Dynamic Webpage**

### Static Webpage

Static webpages are those webpages that cannot be modified or altered by the client. These are also known as stationary or flat web pages. They are displayed on the client's browser in the same format and manner as they are saved in the webserver. Users can only load the page and read the information but cannot perform any change on the page.

A static webpage is generally made up of HTML and CSS only.

### Dynamic Webpage

As the name suggests, Dynamic webpages are dynamic, which means it shows different information at different point of time.

The dynamic webpage shows different content each time it is viewed. There are two types of Dynamic web pages, which are:

- **Server-Side Dynamic Webpage:** These web pages are created using Server-side scripting. These pages are changed when they are visited or viewed. Some examples of server-side pages are **login pages, submission forms, shopping carts, etc.** Various scripting languages such as **PHP, ASP.Net, JSP, etc.**, can be used for server-side scripting.

- **Client-side Dynamic Webpage:** These web pages are created using Client-side scripting. These pages get changed in response to actions that occur within that page, such as mouse or keyboard action. Scripting languages such as **JavaScript, Dart, etc.**, can be used for client-Side scripting.

*Note: Scripting languages are programming languages that allow us to write programs in the form of scripts, and these are interpreted, not compiled.*

Apart from these two Webpages, there are some examples of common web pages that can be found on most of the websites, and these are as follow:

- **Home Page:** The home page of any website is one of the most important pages. It is called a home page because this is like a starting point from where users can go anywhere on that website. This page usually contains links to the important zones of the site. It can also be known as the **index page**.
- **Feed Page:** The feed page is usually found in those websites that update content. It is used to provide information to users for the latest information that has been updated.
- **Menu Page:** The menu page is created to accomplish the navigation goal. The page contains a collection of different links that give access to different categories and zones.
- **About-us Page:** This page contains brief information and details of the company, product, or website. It allows the visitors to know the details of the website that they are using.
- **Registration Page:** The registration page allows users to create an account by signing up, and hence they can create a personalized account. It helps the company to know the visitor and provide personalized offers and deals to them.
- **Contacts Page:** This is a simple page made for the visitors to contact the website owner. For any issue or any feedback, users can use this page.
- **Landing Page:** This is one of the special types of a Web page used as a core part of a website or as an individual page. The main aim of creating this page is to convert the visitors into the lead. It represents clear and focused content on a specific goal.

## How to Create a Simple Webpage?

Creating a simple webpage is very easy; anyone with basic knowledge of computers and HTML can create it. But before creating a webpage, you should be aware of the below points:

- A simple webpage can be created using HTML code only. Such pages are simple but not interactive and have very few functionalities.



- To make your webpage interactive and add functionality, you need to learn and use scripting languages, such as **PHP, Python**, etc.
- A web page can also be created using Notepad, but it is recommended to use IDEs for advanced uses such as **Atom, Sublime Text editor, PyCharm**,

### Follow the below steps to create your webpage:

1. Open the Notepad application on your computer.
2. Write Below code on it.

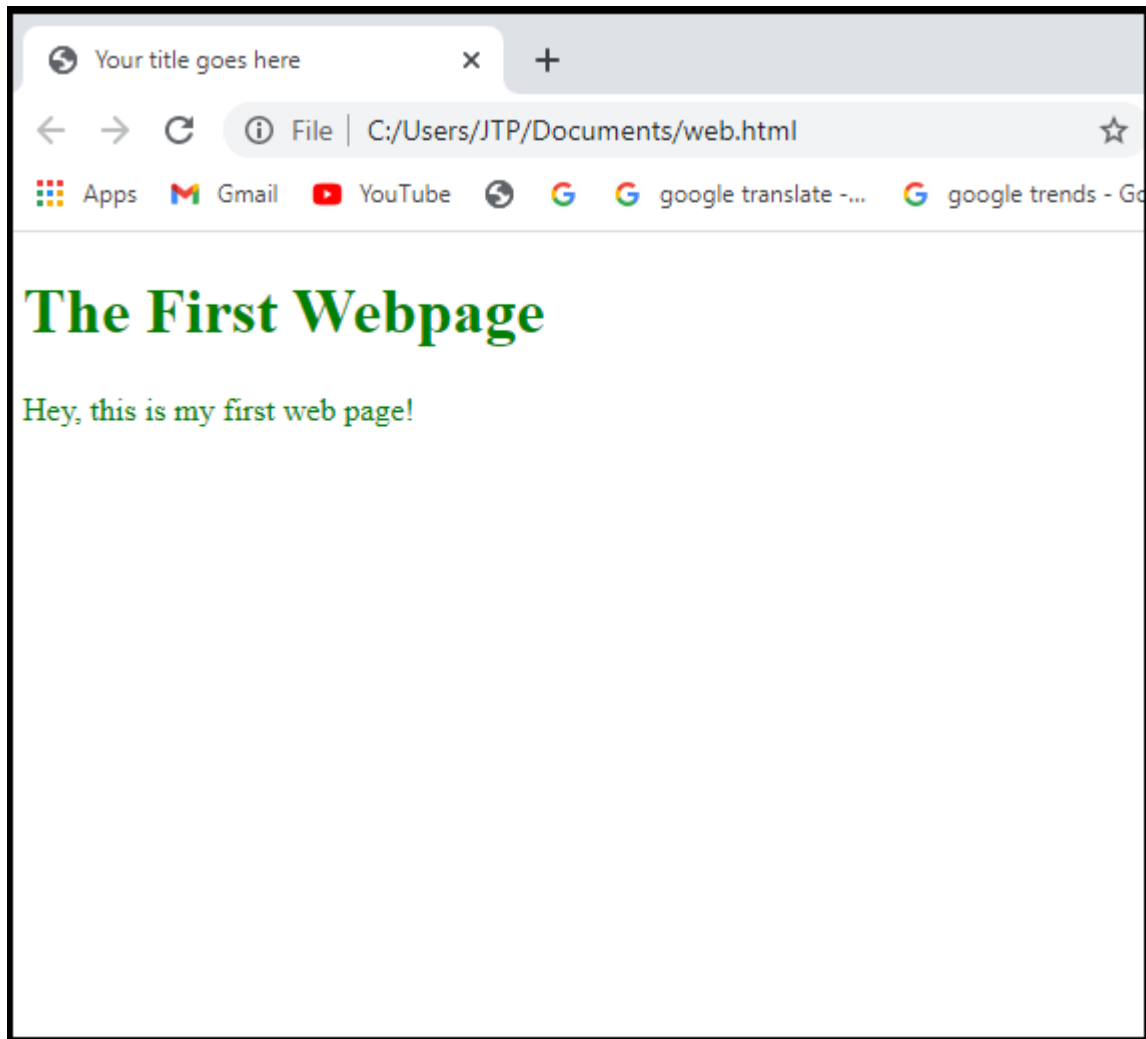
1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Your title goes here</title>`
5. `</head>`
6. `<body bgcolor="white" text="red">`
7. `<h1>first Web page</h1>`
8. `<p>This is my first web page!</p>`
9. `</body>`
10. `</html>`

In the above code, the following tags are used:

- **<!DOCTYPE html>**: It is used for document type declaration, which means which version of HTML you are using. It indicates the browser that which language it is supposed to interpret.
- **<html>**: it indicates the start and of the HTML code.
- **<head>**: It can have different types of information such as title, meta tag, etc.; this information will not appear on the webpage.
- **<body>**: This tag contains other tags on the webpage, and users can see them. In above code, we have included `<h1>` heading tag and `<p>` paragraph tags.

*Note: It is important to end each tag in html, and put all the tags either in upper case or lower case. However, a lower case is recommended.*

3. Save the file with any name and **.html extension**. For example, save it with **html** name.
4. Go to the saved file (web.html), double click or right-click and run it.
5. It will open on your default browser and will display the below output:



You can also add more tags for different elements, such as add images, background images, border, table, table, etc., using HTML. You can learn all these from [here](#).

*Note: This webpage is local to your machine only, and only you can see this on your browser. To view this on the internet, you need to first publish it.*

After creating the page, you can also make changes in your file through the editor. Just make the change, save the file again, and reload the page; those changes will appear on the screen.

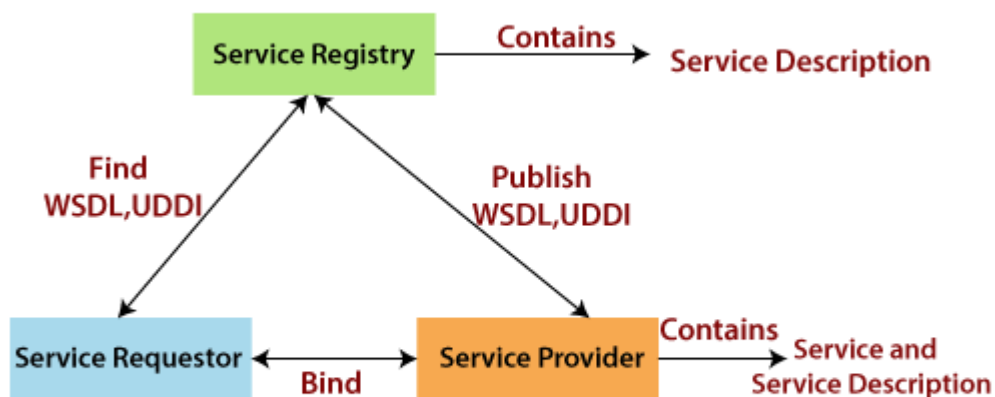
## Architecture of Web Services

The Web Services architecture describes how to instantiate the elements and implement the operations in an interoperable manner.

The architecture of web service interacts among three roles: **service provider**, **service requester**, and **service registry**. The interaction involves the three operations: **publish**, **find**, and **bind**. These operations and roles act upon the **web services artifacts**. The web service artifacts are the web service software module and its description.

The service provider hosts a network-associable module (web service). It defines a service description for the web service and publishes it to a service requestor or service registry. These service requestor uses a find operation to retrieve the service description locally or from the service registry. It uses the service description to bind with the service provider and invoke with the web service implementation.

The following figure illustrates the operations, roles, and their interaction.



Web Service Roles, Operations and Artifacts

## Roles in a Web Service Architecture

There are three roles in web service architecture:

- Service Provider
- Service Requestor
- Service Registry

### Service Provider

From an architectural perspective, it is the platform that hosts the services.

### Service Requestor

Service requestor is the application that is looking for and invoking or initiating an interaction with a service. The browser plays the requester role, driven by a consumer or a program without a user interface.

### Service Registry

Service requestors find service and obtain binding information for services during development.

## Operations in a Web Service Architecture

Three behaviors that take place in the microservices:

- Publication of service descriptions (**Publish**)
- Finding of services descriptions (**Find**)
- Invoking of service based on service descriptions (**Bind**)

**Publish:** In the publish operation, a service description must be published so that a service requester can find the service.

**Find:** In the find operation, the service requestor retrieves the service description directly. It can be involved in two different lifecycle phases for the service requestor:

- At design, time to retrieve the service's interface description for program development.
- And, at the runtime to retrieve the service's binding and location description for invocation.

**Bind:** In the bind operation, the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact, and invoke the service.

## Artifacts of the web service

There are two artifacts of web services:

- Service
- Service Registry

**Service:** A service is an **interface** described by a service description. The service description is the implementation of the service. A service is a software module deployed on network-accessible platforms provided by the service provider. It interacts with a service requestor. Sometimes it also functions as a requestor, using other Web Services in its implementation.

**Service Description:** The service description comprises the details of the **interface** and **implementation** of the service. It includes its **data types, operations, binding information, and network location**. It can also categorize other metadata to enable discovery and utilize by service requestors. It can be published to a service requestor or a service registry.

## Web Service Implementation Lifecycle

A web service implementation lifecycle refers to the phases for developing web services from the requirement to development. An Implementation lifecycle includes the following phases:

- Requirements Phase
- Analysis Phase
- Design Phase
- Coding Phase
- Test Phase
- Deployment Phase



### **Requirements Phase**

The objective of the requirements phase is to understand the business requirement and translate them into the web services requirement. The requirement analyst should do requirement elicitation (it is the practice of researching and discovering the requirements of the system from the user, customer, and other stakeholders). The analyst should interpret, consolidate, and communicate these requirements to the development team. The requirements should be grouped in a centralized repository where they can be viewed, prioritized, and mined for interactive features.

### **Analysis Phase**

The purpose of the analysis phase is to refine and translate the web service into conceptual models by which the technical development team can understand. It also defines the high-level structure and identifies the web service interface contracts.

## Design Phase

In this phase, the detailed design of web services is done. The designers define web service interface contract that has been identified in the analysis phase. The defined web service interface contract identifies the elements and the corresponding data types as well as mode of interaction between web services and client.

## Coding Phase

Coding and debugging phase is quite similar to other software component-based coding and debugging phase. The main difference lies in the creation of additional web service interface wrappers, generation of WSDL, and client stubs.

## Test Phase

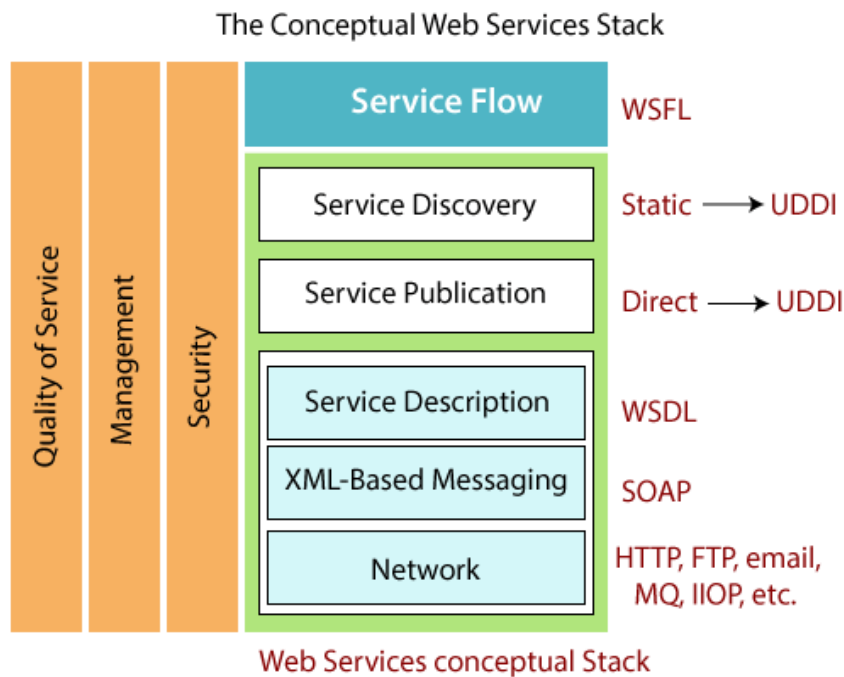
In this phase, the tester performs interoperability testing between the platform and the client's program. Testing to be conducted is to ensure that web services can bear the maximum load and stress. Other tasks like profiling of the web service application and inspection of the SOAP message should also perform in the test phase.

## Deployment Phase

The purpose of the deployment phase is to ensure that the web service is properly deployed in the distributed system. It executes after the testing phase. The primary task of deployer is to ensure that the web service has been properly configured and managed. Other optional tasks like specifying and registering the web service with a UDDI registry also done in this phase.

## Web Service Stack or Web Service Protocol Stack

To perform three operations: publish, find, and bind in an interoperable manner, there must be a **web service stack**. The web service stack embraces the standard at each level.



In the above figure, the top most layers build upon the capabilities provided by the lower layers. The three vertical towers represent the requirements that are applied at every level of the stack. The text on the right represents technologies that apply at that layer of the stack. A web service protocol stack typically stacks four protocols:

- Transport Protocol
- Messaging Protocol
- Description Protocol
- Discovery Protocol

**(Service) Transport Protocol:** The network layer is the foundation of the web service stack. It is responsible for transporting a message between network applications. HTTP is the network protocol for internet available web services. It also supports other network protocol such as **SMTP**, **FTP**, and **BEEP**(Block Extensible Exchange Protocol).

**(XML) Messaging Protocol:** It is responsible for encoding message in a common XML format so that they can understand at either end of a network connection. SOAP is the chosen XML messaging protocol because it supports three operations: publish, find, and bind operation.

**(Service) Description Protocol:** It is used for describing the public interface to a specific web service. WSDL is the standard for XML-based service description. WSDL describes the interface and mechanics of service interaction. The description is necessary to specify the **business context**, **quality of service**, and **service-to-service** relationship.

**(Service) Discovery Protocol:** It is a centralized service into a common registry so that network Web services can publish their location and description. It makes it easy to discover which services are available on the network.

The first three layers of the stack are required to provide or use any web service. The simplest stack consists of HTTP for the network layer, SOAP protocol for the XML-based messaging, and WSDL for the service description layer. These three-layer provides interoperability and enables web service to control the existing internet infrastructure. It creates a low cost of entry to a global environment.

The bottom three layers of the stack identify technologies for compliance and interoperability, the next two layer- **Service Publication** and **Service Discovery** can be implemented with a range of solutions.

## Types of Web Services

There are two types of web services:

- RESTful Web Services
- SOAP Web Services

## RESTful Web Services

REST stands for **REpresentational State Transfer**. It is developed by **Roy Thomas Fielding** who also developed HTTP. The main goal of RESTful web services is to make web services **more effective**. RESTful web services try to define services using the different concepts that are already present in HTTP. REST is an **architectural approach**, not a protocol.

It does not define the standard message exchange format. We can build REST services with both XML and JSON. JSON is more popular format with REST. The **key abstraction** is a resource in REST. A resource can be anything. It can be accessed through a **Uniform Resource Identifier (URI)**. For example:

The resource has representations like XML, HTML, and JSON. The current state is captured by representational resource. When we request a resource, we provide the representation of the resource. The important methods of HTTP are:

- **GET:** It reads a resource.



- **PUT:** It updates an existing resource.
- **POST:** It creates a new resource.
- **DELETE:** It deletes the resource.

For example, if we want to perform the following actions in the social media application, we get the corresponding results.

**POST /users:** It creates a user.

**GET /users/{id}:** It retrieve the detail of one user.

**GET /users:** It retrieve the detail of all users.

**DELETE /users:** It delete all users.

**DELETE /users/{id}:** It delete a user.

**GET /users/{id}/posts/post\_id:** It retrieve the detail of a specific post.

**POST / users/{id}/ posts:** It creates a post for a user.

**GET /users/{id}/post:** Retrieve all posts for a user

HTTP also defines the following standard status code:

- **404:** RESOURCE NOT FOUND
- **200:** SUCCESS
- **201:** CREATED
- **401:** UNAUTHORIZED
- **500:** SERVER ERROR

## RESTful Service Constraints

- There must be a service producer and service consumer.
- The service is stateless.
- The service result must be cacheable.
- The interface is uniform and exposing resources.
- The service should assume a layered architecture.

## Advantages of RESTful web services

- RESTful web services are **platform-independent**.
- It can be written in any programming language and can be executed on any platform.
- It provides different data format like **JSON, text, HTML, and XML**.

- It is fast in comparison to SOAP because there is no strict specification like SOAP.
- These are **reusable**.
- These are **language neutral**.

## SOAP Web Services

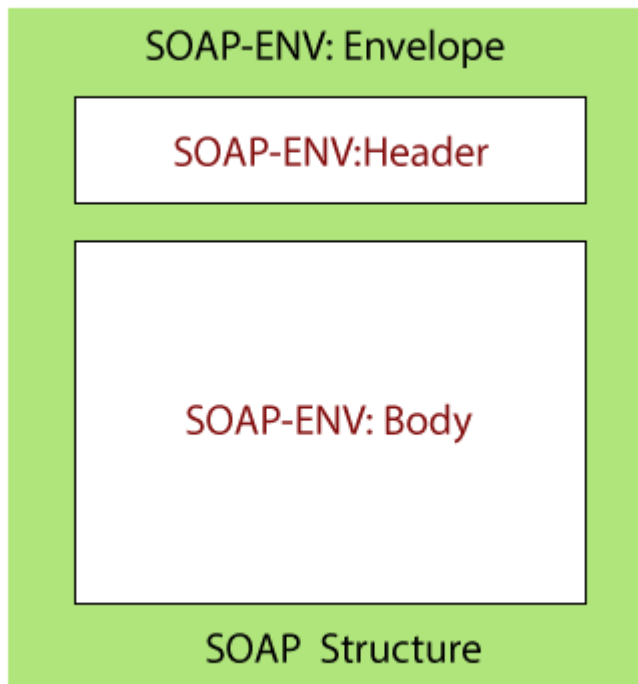
REST defines an architectural approach whereas SOAP poses a restriction on the format of the XML. XML transfer data between the service provider and service consumer. Remember that SOAP and REST are not **comparable**.

**SOAP:** SOAP acronym for **Simple Object Access Protocol**. It defines the standard XML format. It also defines the way of building web services. We use Web Service Definition Language (WSDL) to define the format of **request XML** and the **response XML**.

For example, we have requested to access the **Todo** application from the **Facebook** application. The Facebook application sends an XML request to the Todo application. Todo application processes the request and generates the XML response and sends back to the Facebook application.



If we are using SOAP web services, we have to use the **structure** of SOAP.



In the above figure, the **SOAP-Envelope** contains a **SOAP-Header** and **SOAP-Body**. It contains meta-information needed to identify the request, for example, authentication, authorization, signature, etc. SOAP-Header is optional. The **SOAP-Body** contains the real XML content of request or response. In case of an error, the response server responds back with SOAP-Fault.

Let's understand the SOAP XML request and response structure.

### XML Request

1. `<Envelop xmlns=?http://schemas.xmlsoap.org/soap/envelop/?>`
2.     `<Body>`
3.         `<getCourseDetailRequest xmlns=?http://udemy.com/course?>`
4.             `<id>course1</id>`
5.         `</getCourseDetailRequest>`
6.     `</Body>`
7. `</Envelop>`

### XML Response

1. <SOAP-ENV:Envelope xmlns:SOAP-ENV=?http://schemas.xmlsoap.org/soap/envelope/?>
2.       <SOAP-ENV:Header />       <!--empty header-->
3.       <SOAP-ENV:Body>       <!--body begin-->
4.           <ns2:getCourseDetailsResponse xmlns:ns2=?http://in28mi> <!--  
content of the response-->
5.           <ns2:course>
6.           <ns2:id>Course1</ns2:id>
7.           <ns2:name>Spring<ns2:name>
8.           <ns2:description>10 Steps</ns1:description>
9.           </ns2:course>
10.          </ns2:getCourseDetailResponse>
11.       </SOAP-ENV:Body>       <!--body end-->
12. </SOAP-ENV:Envelope>

## Points to remember

- SOAP defines the format of **request** and **response**.
- SOAP does not pose any restriction on transport. We can either use **HTTP** or **MQ** for communication.
- In SOAP, service definition typically done using **Web Service Definition Language (WSDL)**. WSDL defines **Endpoint**, **All Operations**, **Request Structure**, and **Response Structure**.

The **Endpoint** is the connection point where HTML or ASP pages are exposed. It provides the information needed to address the Web Service endpoint. The operations are the services that are allowed to access. Request structure defines the structure of the request, and the response structure defines the structure of the response.

## Website

Website is a collection of related web pages that may contain text, images, audio and video. The first page of a website is called home page. Each website has specific internet address (URL) that you need to enter in your browser to access a website.

Website is hosted on one or more servers and can be accessed by visiting its homepage using a computer network. A website is managed by its owner that can be an individual, company or an organization.



A website can be of two types:

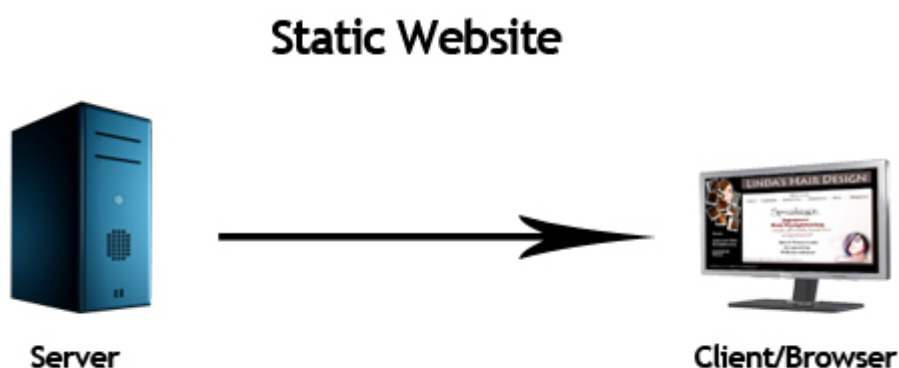
- Static Website
- Dynamic Website

---

## Static website

Static website is the basic type of website that is easy to create. You don't need the knowledge of web programming and database design to create a static website. Its web pages are coded in HTML.

The codes are fixed for each page so the information contained in the page does not change and it looks like a printed page.



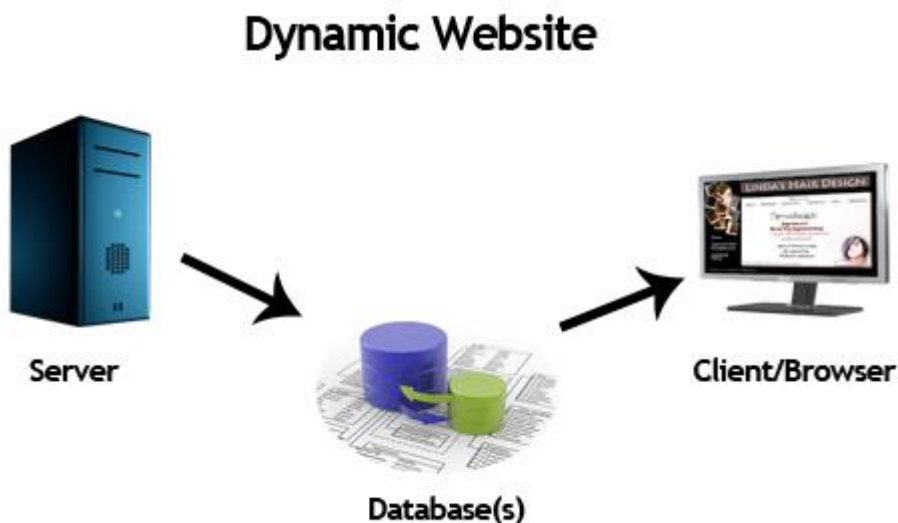
## Dynamic website

Dynamic website is a collection of dynamic web pages whose content changes dynamically. It accesses content from a database or Content Management System (CMS). Therefore, when you alter or update the content of the database, the content of the website is also altered or updated.

Dynamic website uses client-side scripting or server-side scripting, or both to generate dynamic content.

Client side scripting generates content at the client computer on the basis of user input. The web browser downloads the web page from the server and processes the code within the page to render information to the user.

In server side scripting, the software runs on the server and processing is completed in the server then plain pages are sent to the user.



## Static vs Dynamic website

Static Website	Dynamic Website
Prebuilt content is same every time the page is loaded.	Content is generated quickly and changes regularly.
It uses the <b>HTML</b> code for developing a website.	It uses the server side languages such as <b>PHP, SERVLET, JSP, and ASP.NET</b> etc. for developing a website.
It sends exactly the same response for every request.	It may generate different HTML for each of the request.
The content is only changed when	The page contains "server-side" code which allows the

someone publishes and updates the file (sends it to the web server).	server to generate the unique content when the page is loaded.
Flexibility is the main advantage of static website.	Content Management System (CMS) is the main advantage of dynamic website.

## Cyber Security



Cybersecurity is the protection of Internet-connected systems, including hardware, software, and data from cyber attackers. It is primarily about people, processes, and technologies working together to encompass the full range of threat reduction, vulnerability reduction, deterrence, international engagement, and recovery policies and activities, including computer network operations, information assurance, law enforcement, etc.

It is the body of technologies, processes, and practices designed to protect networks, devices, programs, and data from attack, theft, damage, modification, or unauthorized access. Therefore, it may also be referred to as **information technology security**.

Cyber-attack is now an international concern. It has given many concerns that could endanger the global economy. As the volume of cyber-attacks grows, companies and organizations, especially those that deal with information related to national security, health, or financial records, need to take steps to protect their sensitive business and personal information.

This Cyber Security tutorial provides basic and advanced concepts of Cyber Security technology. It will cover the most popular concept of Cyber Security, such as what is Cyber Security, Cyber Security goals, types of cyber-attacks, types of cyber attackers, policies, digital signature, Cyber Security tools, security risk analysis, challenges, etc.

## Prerequisites

It is a basic tutorial where we can quickly understand the topics discussed if we have a basic understanding of how a firm or organization handles computer security. It is also helpful for us to have some prior experience with computer updates, firewalls, antiviruses, and other security measures.

## Audience

Our Cyber Security tutorial is designed to help beginners and professionals.

## Problems

We assure you that you will not find any problem with this tutorial. However, if you find any, you can post it on the contact form.

## Types of Cyber Security

Every organization's assets are the combinations of a variety of different systems. These systems have a strong cybersecurity posture that requires coordinated efforts across all of its systems. Therefore, we can categorize cybersecurity in the following sub-domains:

- **Network Security:** It involves implementing the hardware and software to secure a computer network from unauthorized access, intruders, attacks, disruption, and misuse. This security helps an organization to protect its assets against external and internal threats.
- **Application Security:** It involves protecting the software and devices from unwanted threats. This protection can be done by constantly updating the apps to ensure they are secure from attacks. Successful security begins in the design stage, writing source code, validation, threat modeling, etc., before a program or device is deployed.
- **Information or Data Security:** It involves implementing a strong data storage mechanism to maintain the integrity and privacy of data, both in storage and in transit.
- **Identity management:** It deals with the procedure for determining the level of access that each individual has within an organization.
- **Operational Security:** It involves processing and making decisions on handling and securing data assets.
- **Mobile Security:** It involves securing the organizational and personal data stored on mobile devices such as cell phones, computers, tablets, and other similar devices against various malicious threats. These threats are unauthorized access, device loss or theft, malware, etc.
- **Cloud Security:** It involves in protecting the information stored in the digital environment or cloud architectures for the organization. It uses various cloud service



providers such as AWS, Azure, Google, etc., to ensure security against multiple threats.

- **Disaster Recovery and Business Continuity Planning:** It deals with the processes, monitoring, alerts, and plans to how an organization responds when any malicious activity is causing the loss of operations or data. Its policies dictate resuming the lost operations after any disaster happens to the same operating capacity as before the event.
- **User Education:** It deals with the processes, monitoring, alerts, and plans to how an organization responds when any malicious activity is causing the loss of operations or data. Its policies dictate resuming the lost operations after any disaster happens to the same operating capacity as before the event.

## Why is Cyber Security important?

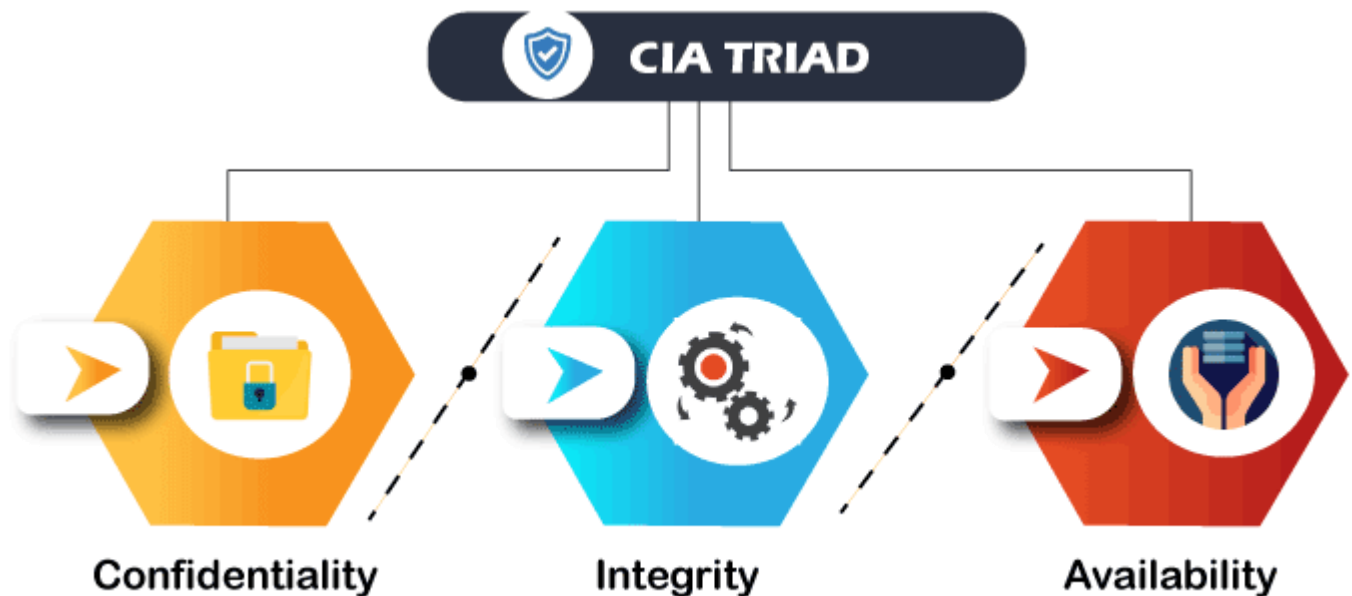
Today we live in a digital era where all aspects of our lives depend on the network, computer and other electronic devices, and software applications. All critical infrastructure such as the banking system, healthcare, financial institutions, governments, and manufacturing industries use **devices connected to the Internet** as a core part of their operations. Some of their information, such as intellectual property, financial data, and personal data, can be sensitive for unauthorized access or exposure that could have **negative consequences**. This information gives intruders and threat actors to infiltrate them for financial gain, extortion, political or social motives, or just vandalism.

Cyber-attack is now an international concern that hacks the system, and other security attacks could endanger the global economy. Therefore, it is essential to have an excellent cybersecurity strategy to protect sensitive information from high-profile security breaches. Furthermore, as the volume of cyber-attacks grows, companies and organizations, especially those that deal with information related to national security, health, or financial records, need to use strong cybersecurity measures and processes to protect their sensitive business and personal information.

## Cyber Security Goals

Cyber Security's main **objective is to ensure data protection**. The security community provides a triangle of three related principles to protect the data from cyber-attacks. This principle is called the **CIA triad**. The CIA model is designed to guide policies for an organization's information security infrastructure. When any security breaches are found, one or more of these principles has been violated.

We can break the **CIA model into three parts**: Confidentiality, Integrity, and Availability. It is actually a security model that helps people to think about various parts of IT security. Let us discuss each part in detail.



### Confidentiality

Confidentiality is equivalent to privacy that avoids unauthorized access of information. It involves ensuring the data is accessible by those who are allowed to use it and blocking access to others. It prevents essential information from reaching the wrong people. **Data encryption** is an excellent example of ensuring confidentiality.

### Integrity

This principle ensures that the data is authentic, accurate, and safeguarded from unauthorized modification by threat actors or accidental user modification. If any modifications occur, certain measures should be taken to protect the sensitive data from corruption or loss and speedily recover from such an event. In addition, it indicates to make the source of information genuine.

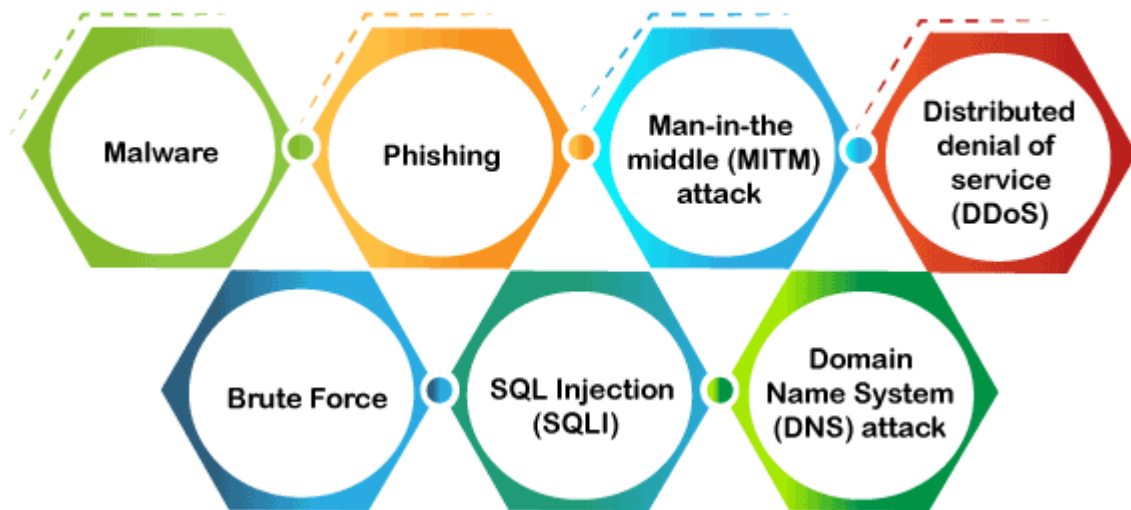
### Availability

This principle makes the information to be available and useful for its authorized people always. It ensures that these accesses are not hindered by system malfunction or cyber-attacks.

## Types of Cyber Security Threats

A threat in cybersecurity is a malicious activity by an individual or organization to corrupt or steal data, gain access to a network, or disrupts digital life in general. The cyber community defines the following threats available today:

## Types of Cyber Threats



### Malware

Malware means malicious software, which is the most common cyber attacking tool. It is used by the cybercriminal or hacker to disrupt or damage a legitimate user's system. The following are the important types of malware created by the hacker:

- **Virus:** It is a malicious piece of code that spreads from one device to another. It can clean files and spreads throughout a computer system, infecting files, steals information, or damage device.
- **Spyware:** It is a software that secretly records information about user activities on their system. **For example,** spyware could capture credit card details that can be used by the cybercriminals for unauthorized shopping, money withdrawing, etc.
- **Trojans:** It is a type of malware or code that appears as legitimate software or file to fool us into downloading and running. Its primary purpose is to corrupt or steal data from our device or do other harmful activities on our network.
- **Ransomware:** It's a piece of software that encrypts a user's files and data on a device, rendering them unusable or erasing. Then, a monetary ransom is demanded by malicious actors for decryption.
- **Worms:** It is a piece of software that spreads copies of itself from device to device without human interaction. It does not require them to attach themselves to any program to steal or damage the data.
- **Adware:** It is an advertising software used to spread malware and displays advertisements on our device. It is an unwanted program that is installed without the user's permission. The main objective of this program is to generate revenue for its developer by showing the ads on their browser.

- **Botnets:** It is a collection of internet-connected malware-infected devices that allow cybercriminals to control them. It enables cybercriminals to get credentials leaks, unauthorized access, and data theft without the user's permission.

## Phishing

Phishing is a type of cybercrime in which **a sender seems to come from a genuine organization** like PayPal, eBay, financial institutions, or friends and co-workers. They contact a target or targets via email, phone, or text message with a link to persuade them to click on that links. This link will redirect them to fraudulent websites to provide sensitive data such as personal information, banking and credit card information, social security numbers, usernames, and passwords. Clicking on the link will **also install malware** on the target devices that allow hackers to control devices remotely.

## Man-in-the-middle (MITM) attack

A man-in-the-middle attack is a type of cyber threat (a form of eavesdropping attack) in which a cybercriminal **intercepts a conversation or data transfer between two individuals**. Once the cybercriminal places themselves in the middle of a two-party communication, they seem like genuine participants and can get sensitive information and return different responses. The main objective of this type of attack is to gain access to our business or customer data. **For example**, a cybercriminal could intercept data passing between the target device and the network on an unprotected Wi-Fi network.

## Distributed denial of service (DDoS)

It is a type of cyber threat or malicious attempt where cybercriminals disrupt targeted servers, services, or network's regular traffic by fulfilling legitimate requests to the target or its surrounding infrastructure with Internet traffic. Here the requests come from several IP addresses that can make the system unusable, overload their servers, slowing down significantly or temporarily taking them offline, or preventing an organization from carrying out its vital functions.

## Brute Force

A brute force attack is a **cryptographic hack that uses a trial-and-error method** to guess all possible combinations until the correct information is discovered. Cybercriminals usually use this attack to obtain personal information about targeted passwords, login info, encryption keys, and Personal Identification Numbers (PINS).

## SQL Injection (SQLI)

SQL injection is a common attack that occurs when cybercriminals use malicious SQL scripts for backend database manipulation to access sensitive information. Once the

attack is successful, the malicious actor can view, change, or delete sensitive company data, user lists, or private customer details stored in the SQL database.

## Domain Name System (DNS) attack

A DNS attack is a type of cyberattack in which cyber criminals take advantage of flaws in the Domain Name System to redirect site users to malicious websites (DNS hijacking) and steal data from affected computers. It is a severe cybersecurity risk because the DNS system is an essential element of the internet infrastructure.

## Latest cyber threats

The following are the latest cyber threats reported by the U.K., U.S., and Australian governments:

### Romance Scams

The U.S. government found this cyber threat in **February 2020**. Cybercriminals used this threat through dating sites, chat rooms, and apps. They attack people who are seeking a new partner and duping them into giving away personal data.

### Dridex Malware

It is a type of financial Trojan malware identifies by the U.S. in **December 2019** that affects the public, government, infrastructure, and business worldwide. It infects computers through phishing emails or existing malware to steal sensitive information such as passwords, banking details, and personal data for fraudulent transactions. The National Cyber Security Centre of the United Kingdom encourages people to make sure their devices are patched, anti-virus is turned on and up to date, and files are backed up to protect sensitive data against this attack.

### Emotet Malware

Emotet is a type of cyber-attack that steals sensitive data and also installs other malware on our device. The Australian Cyber Security Centre warned national organizations about this global cyber threat in 2019.

**The following are the system that can be affected by security breaches and attacks:**

- **Communication:** Cyber attackers can use phone calls, emails, text messages, and messaging apps for cyberattacks.
- **Finance:** This system deals with the risk of financial information like bank and credit card detail. This information is naturally a primary target for cyber attackers.
- **Governments:** The cybercriminal generally targets the government institutions to get confidential public data or private citizen information.

- **Transportation:** In this system, cybercriminals generally target connected cars, traffic control systems, and smart road infrastructure.
- **Healthcare:** A cybercriminal targets the healthcare system to get the information stored at a local clinic to critical care systems at a national hospital.
- **Education:** A cybercriminals target educational institutions to get their confidential research data and information of students and employees.

## Benefits of cybersecurity

The following are the benefits of implementing and maintaining cybersecurity:

- Cyberattacks and data breach protection for businesses.
- Data and network security are both protected.
- Unauthorized user access is avoided.
- After a breach, there is a faster recovery time.
- End-user and endpoint device protection.
- Regulatory adherence.
- Continuity of operations.
- Developers, partners, consumers, stakeholders, and workers have more faith in the company's reputation and trust.

## Cyber Safety Tips

Let us see how to protect ourselves when any cyberattacks happen. The following are the popular cyber safety tips:

**Conduct cybersecurity training and awareness:** Every organization must train their staffs on cybersecurity, company policies, and incident reporting for a strong cybersecurity policy to be successful. If the staff does unintentional or intentional malicious activities, it may fail the best technical safeguards that result in an expensive security breach. Therefore, it is useful to conduct security training and awareness for staff through seminars, classes, and online courses that reduce security violations.

**Update software and operating system:** The most popular safety measure is to update the software and O.S. to get the benefit of the latest security patches.

**Use anti-virus software:** It is also useful to use the anti-virus software that will detect and removes unwanted threats from your device. This software is always updated to get the best level of protection.

**Perform periodic security reviews:** Every organization ensures periodic security inspections of all software and networks to identify security risks early in a secure

environment. Some popular examples of security reviews are application and network penetration testing, source code reviews, architecture design reviews, and red team assessments. In addition, organizations should prioritize and mitigate security vulnerabilities as quickly as possible after they are discovered.

**Use strong passwords:** It is recommended to always use long and various combinations of characters and symbols in the password. It makes the passwords are not easily guessable.

**Do not open email attachments from unknown senders:** The cyber expert always advises not to open or click the email attachment getting from unverified senders or unfamiliar websites because it could be infected with malware.

**Avoid using unsecured Wi-Fi networks in public places:** It should also be advised not to use insecure networks because they can leave you vulnerable to man-in-the-middle attacks.

**Backup data:** Every organization must periodically take backup of their data to ensure all sensitive data is not lost or recovered after a security breach. In addition, backups can help maintain data integrity in cyber-attack such as SQL injections, phishing, and ransomware.

## What is Cyber Law?

Cyber law, also known as Internet Law or Cyber Law, is the part of the overall legal system that is related to legal informatics and supervises the digital circulation of information, e-commerce, software and information security. It is associated with legal informatics and electronic elements, including information systems, computers, software, and hardware. It covers many areas, such as access to and usage of the Internet, encompassing various subtopics as well as freedom of expression, and online privacy.





Cyber laws help to reduce or prevent people from cybercriminal activities on a large scale with the help of protecting information access from unauthorized people, freedom of speech related to the use of the [Internet](#)

, privacy, communications, email, websites, intellectual property, hardware and software, such as data storage devices. As Internet traffic is increasing rapidly day by day, that has led to a higher percentage of legal issues worldwide. Because cyber laws are different according to the country and jurisdiction, restitution ranges from fines to imprisonment, and enforcement is challenging.

Cyberlaw offers legal protections for people who are using the Internet as well as running an online business. It is most important for Internet users to know about the local area and cyber law of their country by which they could know what activities are legal or not on the network. Also, they can prevent ourselves from unauthorized activities.

The Computer Fraud and Abuse Act was the first cyber law, called CFFA, that was enacted in 1986. This law was helpful in preventing unauthorized access to computers. And it also provided a description of the stages of punishment for breaking that law or performing any illegal activity.

Nested Structure in C

Keep Watching

## Why are cyber laws needed?

There are many security issues with using the Internet and also available different malicious people who try to unauthorized access your computer system to perform potential fraud. Therefore, similarly, any law, cyber law is created to protect online organizations and people on the network from unauthorized access and malicious people. If someone does any illegal activity or breaks the cyber rule, it offers people



or organizations to have that persons sentenced to punishment or take action against them.

## What happens if anyone breaks a cyber law?

If anyone breaks a cyber law, the action would be taken against that person on the basis of the type of cyberlaw he broke, where he lives, and where he broke the law. There are many situations like if you break the law on a website, your account will be banned or suspended and blocked your [IP \(Internet Protocol\)](#)

address. Furthermore, if any person performs a very serious illegal activity, such as causing another person or company distress, hacking, attacking another person or website, advance action can be taken against that person.

## Importance of Cyber Law

Cyber laws are formed to punish people who perform any illegal activities online. They are important to punish related to these types of issues such as online harassment, attacking another website or individual, data theft, disrupting the online workflow of any enterprise and other illegal activities.

If anyone breaks a cyber law, the action would be taken against that person on the basis of the type of cyberlaw he broke, where he lives, and where he broke the law. It is most important to punish the criminals or to bring them to behind bars, as most of the cybercrimes cross the limit of crime that cannot be considered as a common crime.

These crimes may be very harmful for losing the reliability and confidentiality of personal information or a nation. Therefore, these issues must be handled according to the laws.

- When users apply transactions on the Internet, cyber law covers every transaction and protect them.
- It touches every reaction and action in cyberspace.
- It captures all activities on the Internet.

## Areas involving in Cyber Laws

These laws deal with multiple activities and areas that occur online and serve several purposes. Some laws are formed to describe the policies for using the Internet and the computer in an organization, and some are formed to offer people security from unauthorized users and malicious activities. There are various broad categories that come under cyber laws; some are as follows:

### **Fraud**

Cyber laws are formed to prevent financial crimes such as identity theft, credit card theft and other that occurring online. A person may face confederate or state criminal charges if he commits any type of identity theft. These laws have explained strict policies to prosecute and defend against allegations of using the internet.

### **Copyrighting Issues**

The Internet is the source that contains different types of data, which can be accessed anytime, anywhere. But it is the authority of anyone to copy the content of any other person. The strict rules are defined in the cyber laws if anyone goes against copyright that protects the creative work of individuals and companies.

### **Scam/ Treachery**

There are different frauds and scams available on the Internet that can be personally harmful to any company or an individual. Cyber laws offer many ways to protect people and prevent any identity theft and financial crimes that happen online.

## **Online Insults and Character Degradation**

There are multiple online social media platforms that are the best resources to share your mind with anyone freely. But there are some rules in cyber laws if you speak and defaming someone online. Cyber laws address and deal with many issues, such as racism, online insults, gender targets to protect a person's reputation.

### **Online Harassment and Stalking**

Harassment is a big issue in cyberspace, which is a violation of both criminal laws and civil. In cyber laws, there are some hard laws defined to prohibit these kinds of despicable crimes.

### **Data Protection**

People using the internet depends on cyber laws and policies to protect their personal information. Companies or organizations are also relying on cyber laws to protect the data of their users as well as maintain the confidentiality of their data.

### **Contracts and Employment Law**

When you are visiting a website, you click a button that gives a message to ask you to agree for terms and conditions; if you agree with it, that ensures you have used cyber law. For every website, there are terms and conditions available that are associated with privacy concerns.

### **Trade Secrets**

There are many organizations that are doing online businesses, which are often relying on cyber laws to protect their trade secrets. For example, online search

engines like Google spend much time to develop the algorithms that generate a search result. They also spend lots of time developing other features such as intelligent assistance, flight search services, to name a few and maps. Cyber laws help these organizations to perform legal action by describing necessary legal laws for protecting their trade secrets.

## How to protect yourself on the Internet

Although the Internet is a resource that contains multiple different types of content, there are many hackers or unauthorized users that may be harmful to you in order to thief your personal information. Below are given all of the steps that may help you to keep your personal information and computers safe while using the Internet. All of the given steps or suggestions can be beneficial for all computer users, even if what type of computer, device, or operating system they are using.



### Verify data is encrypted

When you are sending any confidential information, such as debit card numbers, credit card numbers, usernames, or passwords, send these types of information securely. In Internet browsers, look for a small lock (Internet browser security lock) to verify this; an icon will be shown in the right corner of the bottom of the browser address bar or browser Window. If you see the icon, it should be in a locked condition and not in an unlocked position. Also, make sure the [URL](#)

starts with [https \(Hypertext Transfer Protocol Secure\)](#)

, as displaying in the below screenshot:



### **Internet Explorer secure address bar.**

If the lock icon is in the locked position and data is intercepted, the data is encrypted that helps to keep secure your data and prevent others to understand it. The data can be read by anyone if the lock is in the unlocked position or no lock is visible because all information will be in the form of plain text. For example, an online forum is not secure, use a password, but you will not use the password with protected sites like an online banking website.

### **Use a safe password**

Like online bank site or other websites that contain confidential information, need to use very strong passwords, it is also recommended; you must use the different and strong password for all websites that require login id and password. You could use a password manager if you required help to remember your password.

### **Keep your software and operating system up-to-date**

To protect yourself on the Internet, it is better to update your software installed on your computer and operating system regularly. It is necessary because many updates are released by the developers of the operating system that are related to computer security-related issues. Therefore, you should update your system when the latest updates are released.

### **When available always enable two-factor authentication**

You can use the two-factor authentication feature to make more secure your accounts, like Gmail or others that require a login and contain your private data. It offers advanced protection by adding an additional step in verifying you at the time of login. If you enable two-factor authentication and the service does not verify your computer or other devices after authenticating your password, it sends a text message with a verification code on your cell phone. It includes more powerful security; for example, if someone knows your password of any account and tries to access your account, but he does not have your phone, he cannot access your account even with a valid password.

### **Always be cautious of e-mail links and attachments**

The email attachments and hyperlinks sent through email are the most common resources to spread viruses and malware. It is recommended to always be extremely cautious to open any attachments and hyperlinks, which you have received through email from others, even if they have sent by friend or family.

## Be aware of phishing scams

There are many phishing scams and techniques that can be more harmful in respect to losing your secret information. Therefore, it is necessary to familiarize yourself with these types of techniques. Hackers mainly target websites that need a login, such as PayPal, eBay, Amazon, online banking sites, and other popular sites.

## E-mail is not encrypted

If you send any confidential information through email, it can be read or understood by unauthorized users as email is not encrypted. Therefore, confidential data like debit card information, credit card information, password and more should not be transmitted over e-mail.

## Use an alternative browser

For protecting your systems, Internet browsers also play an important role. For example, earlier versions of Internet Explorer are not more secure. If you are using a less secure browser in terms of your [browser](#)

like [Internet Explorer](#)

, you should switch to another browser like [Mozilla Firefox](#)

or [Google Chrome](#)

. Also, if you are using Microsoft Windows 10 operating system on your computer and want to stay to use a Microsoft Internet browser, you can switch to the Microsoft Edge rather than Internet Explorer that is more secure in terms of protecting your systems.

## Use caution when accepting or agreeing to prompts

When you are indicated to install an add-on or any program, before clicking on the Ok button, you need to read and understand the agreement carefully. If you do not understand the agreement or feel it is not necessary to install, you should not install this kind of program, cancel or close the window, which may be harmful for you.

Also, when you are installing an add-on or any program, you need to care about any check box that asks if this third-party program will be ok to install. These often cause more issues and leave these boxes unchecked because these are never required.

## Be cautious where you are logging in from

### Business

If you are working in any organization, your place of work can monitor your computer by installing key loggers or use other methods. In this case, someone can collect usernames and passwords and read these logs if he has access to this information. It can be more harmful to lose your personal information. Additionally, if

your computer is shared with other co-workers, do not store any passwords in your browser.

### **Wireless network**

When you are using a wireless network, you must be careful that all the information sent from your computer and to your computer can be read and intercepted by any unauthorized person. You can log in to the network securely with the help of using WPA or WEP and prevent losing your secret information. Furthermore, make sure the network is secure if it is a home wireless network.

### **Friend's house**

Sometimes, you may use your friend's computer and log in to your account on that computer, which may not be fully secure. Intentionally or unintentionally, you can enter your username and password on your friend's computer or the computer with whom you are not familiar. Finally, never save the password information on your friend's computer browser when you are logging into any site on a friend's computer.

### **Always think before you share something**

There are many social media sites, such as Instagram, Facebook, that enable you to make online friends and connect with them. The networking sites are also the best place to share your personal information with your friends, family or others. When you share something on social networking sites or the Internet, make sure you are not sending any information that can be harmful to you if everyone sees it. The sent information on the social network or the Internet should be public. Also, make sure you are sharing such something that will not offend anyone or embarrass you, and you must not be uploaded on the Internet.

### **Update Internet browser plugins**

You should update Internet browser plugins or install the latest plugins to protect yourself while online on the computer. Due to browser plugins like Adobe Flash, attackers may find some easiness or security vulnerabilities to hack any system. Therefore, you need to check out regularly that all your installed Internet plug-ins are up-to-date.

### **Be aware of those around you**

If you are working on the computer at any public area, school, library and more, make sure anyone is not looking at your screen, as there will be many people around you. On the other hand, it can be cautious if anyone is looking at your system screen that is called shoulder surfing. If you are required to system screen private, you can use a privacy filter for the display.

## Secure saved passwords

There are many users that are habitual to save login information and password on the system, but it can be insecure. Therefore, make sure you are storing your personal details, such as credit card detail and account passwords, in a secure area. It is recommended for everyone to use a password manager to save your passwords.

A password manager is a software that holds all securely encrypts and login information, and password protects that information. If you save a password in a browser and anyone has access to your Internet browser, the password information may be seen by that person. For instance, in the Firefox Internet browser, anyone can see all stored passwords if you do not set up a master password.

## Do not always trust what you read online

You should be aware about that it is possible for anyone to publish a website on the Internet. There are various creators who may have intention for creating a site only for malicious purposes. For instance, a website can be created to gain unauthorized access and spread fear, lies, or malware.

# UNIT-2

## What is HTML

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

**Hyper Text:** HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

**Markup language:** A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

**Web Page:** A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages.**

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML

document is made of many HTML tags and each HTML tag contains different content.

**Let's see a simple example of HTML.**

1. `<!DOCTYPE>`
2. `<html>`
3. `<head>`
4. `<title>Web page title</title>`
5. `</head>`
6. `<body>`
7. `<h1>Write Your First Heading</h1>`
8. `<p>Write Your First Paragraph.</p>`
9. `</body>`
10. `</html>`

**Test it Now**

---

## Description of HTML Example

**<!DOCTYPE>**: It defines the document type or it instruct the browser about the version of HTML.

**<html >**: This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except <!DOCTYPE>

**<head>**: It should be the first element inside the <html> element, which contains the metadata(information about the document). It must be closed before the body tag opens.

**<title>**: As its name suggested, it is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately. (Optional)

**<body>** : Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.

**<h1>** : Text between <h1> tag describes the first level heading of the webpage.

**<p>** : Text between <p> tag describes the paragraph of the webpage.

---

## Brief History of HTML



In the late 1980's , a physicist, Tim Berners-Lee who was a contractor at CERN, proposed a system for CERN researchers. In 1989, he wrote a memo proposing an internet based hypertext system.

**Tim Berners-Lee** is known as the father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991. The latest version of HTML is HTML5, which we will learn later in this tutorial.

---

## HTML Versions

Since the time HTML was invented there are lots of HTML versions in market, the brief introduction about the HTML version is given below:

**HTML 1.0:** The first version of HTML was 1.0, which was the barebones version of HTML language, and it was released in 1991.

**HTML 2.0:** This was the next version which was released in 1995, and it was standard language version for website design. HTML 2.0 was able to support extra features such as form-based file upload, form elements such as text box, option button, etc.

**HTML 3.2:** HTML 3.2 version was published by W3C in early 1997. This version was capable of creating tables and providing support for extra options for form elements. It can also support a web page with complex mathematical equations. It became an official standard for any browser till January 1997. Today it is practically supported by most of the browsers.

**HTML 4.01:** HTML 4.01 version was released on December 1999, and it is a very stable version of HTML language. This version is the current official standard, and it provides added support for stylesheets (CSS) and scripting ability for various multimedia elements.

**HTML5 :** HTML5 is the newest version of HyperText Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG( Web Hypertext Application Technology Working Group) which are involved in the development of HTML 5 version, and still, it is under development.

---

## Features of HTML

- 1) It is a very **easy and simple language**. It can be easily understood and modified.
- 2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.

- 3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.
- 4) It facilitates programmers to add a **link** on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.
- 5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
- 6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.
- 7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

*NOTE: It is recommended to write all tags in lower-case for consistency, readability, etc.*

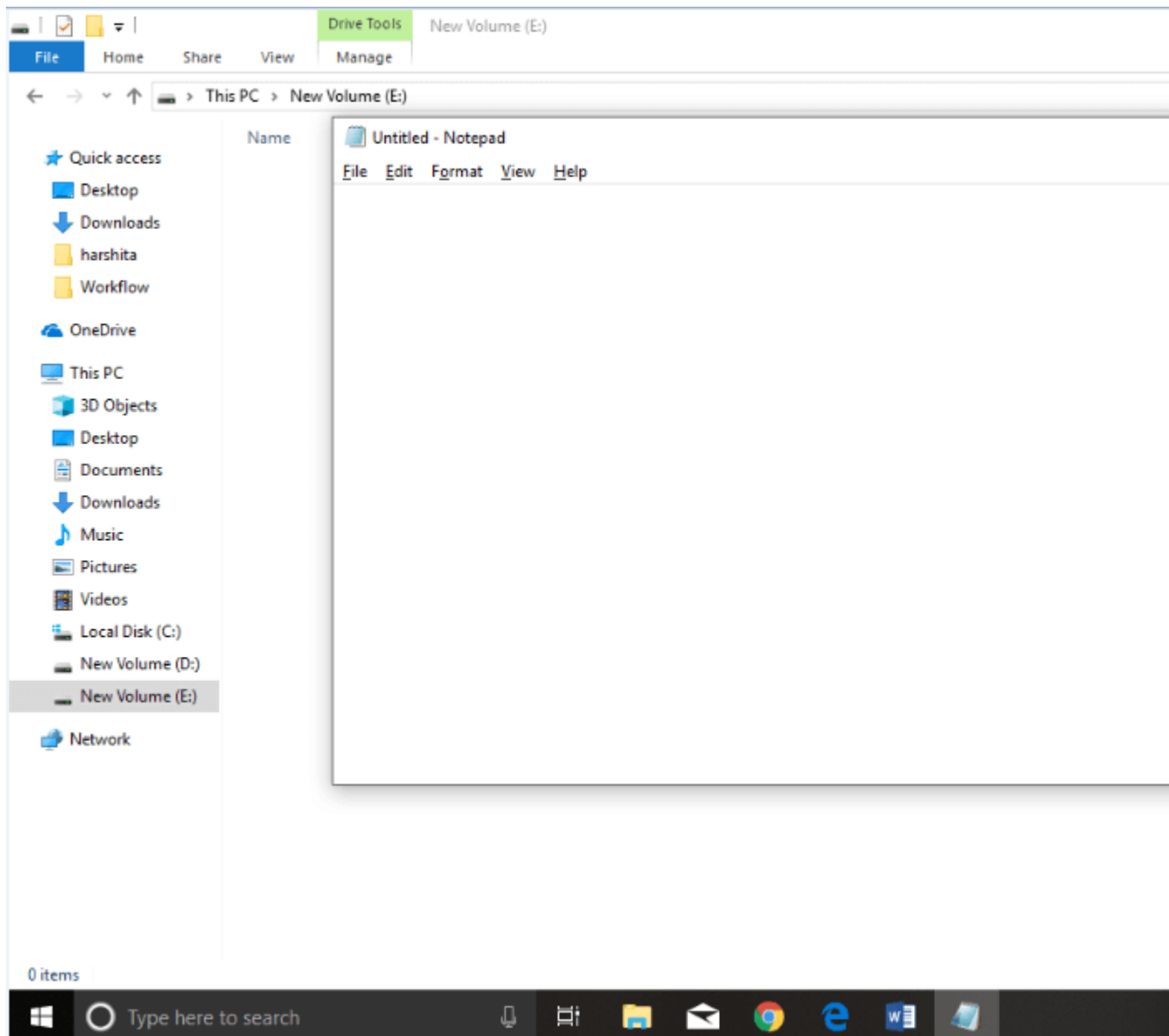
## HTML text Editors

- An HTML file is a text file, so to create an HTML file we can use any text editors.
- Text editors are the programs which allow editing in a written text, hence to create a web page we need to write our code in some text editor.
- There are various types of text editors available which you can directly download, but for a beginner, the best text editor is Notepad (Windows) or TextEdit (Mac).
- After learning the basics, you can easily use other professional text editors which are, **Notepad++**, **Sublime Text**, **Vim**, etc.
- In our tutorial, we will use Notepad and sublime text editor. Following are some easy ways to create your first web page with Notepad, and sublime text.

### A. HTML code with Notepad. (Recommended for Beginners)

Notepad is a simple text editor and suitable for beginners to learn HTML. It is available in all versions of Windows, from where you easily access it.

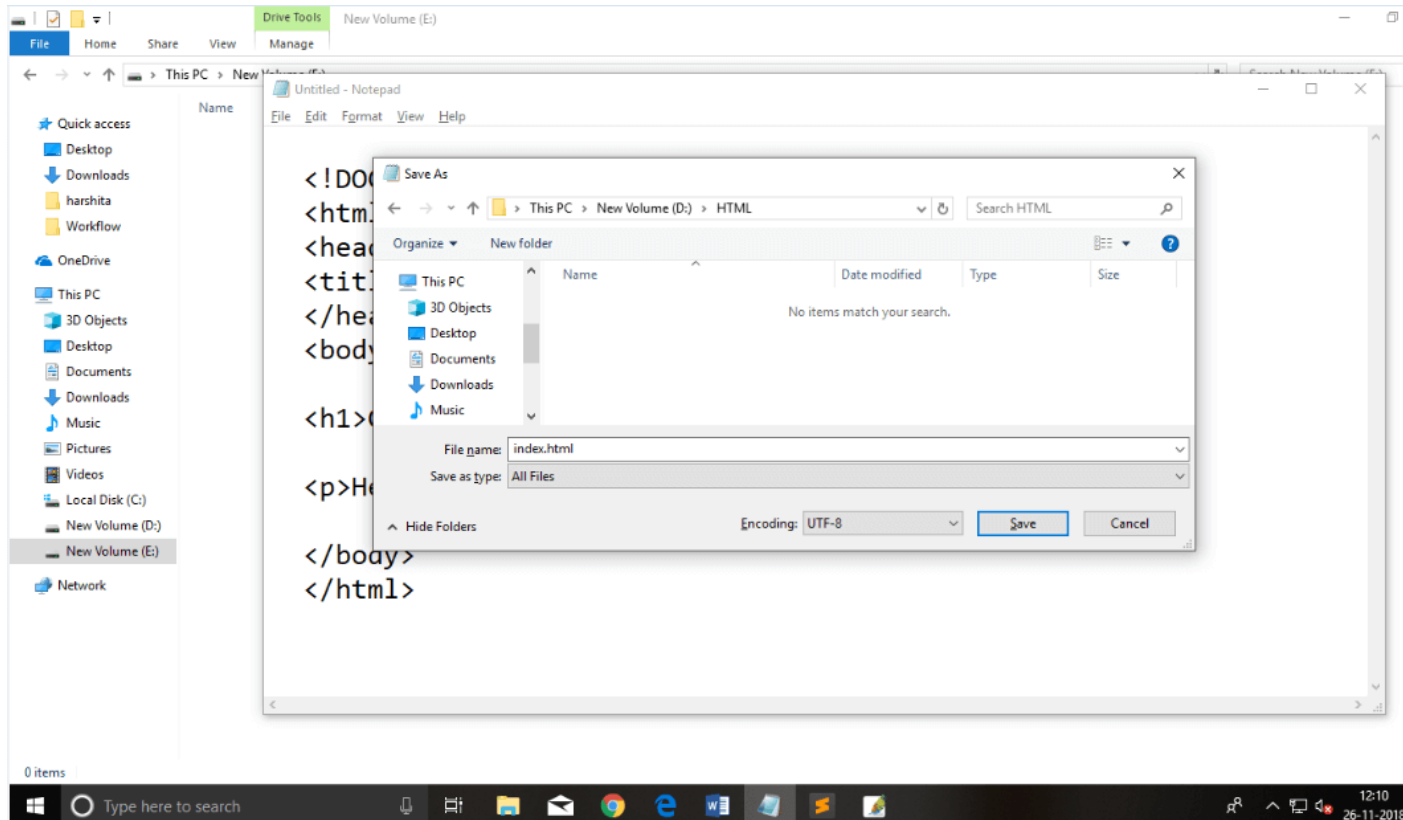
#### Step 1: Open Notepad (Windows)



## Step 2: Write code in HTML

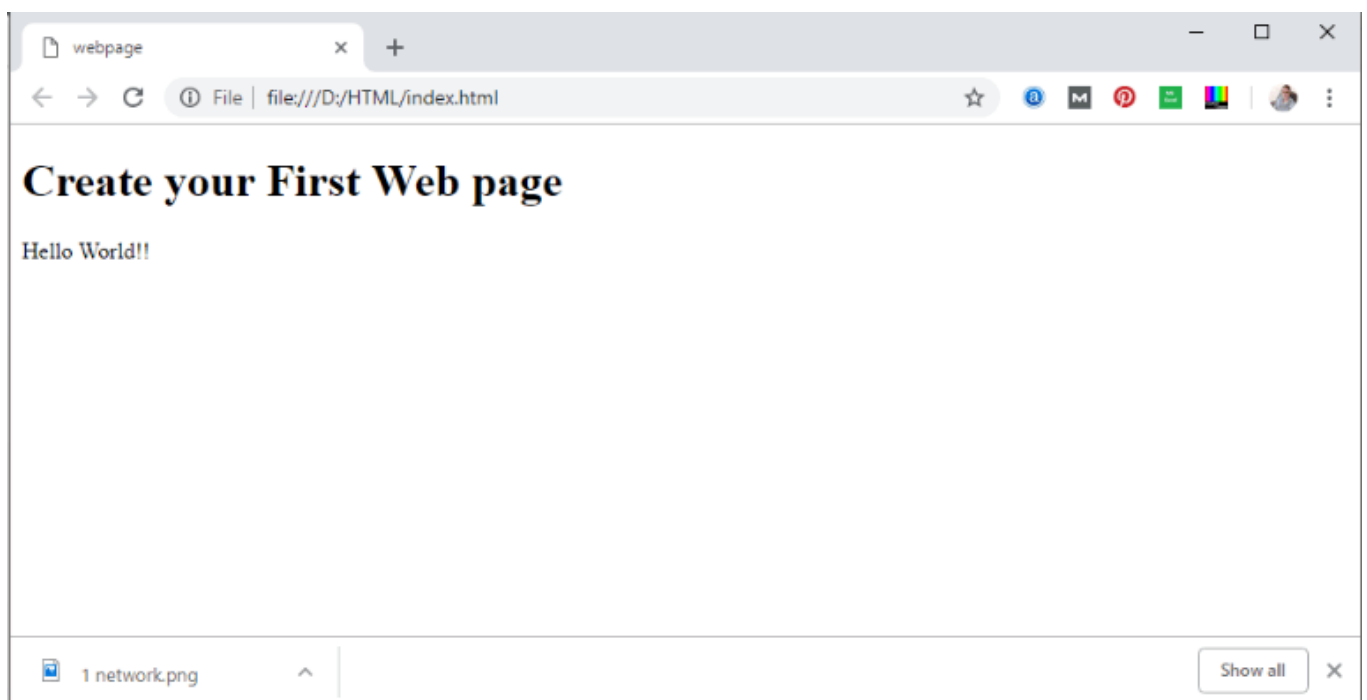
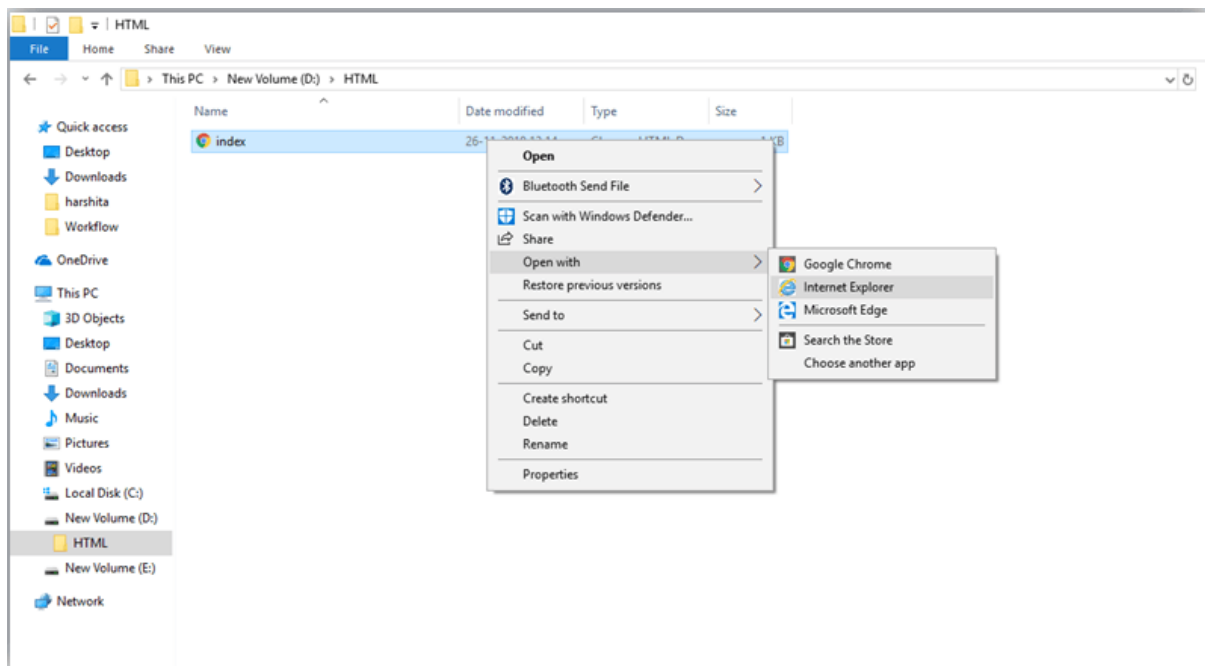


**Step 3: Save the HTML file with .htm or .html extension.**



**Step 4: Open the HTML page in your web browser.**

To run the HTML page, you need to open the file location, where you have saved the file and then either double-click on file or click on open with option



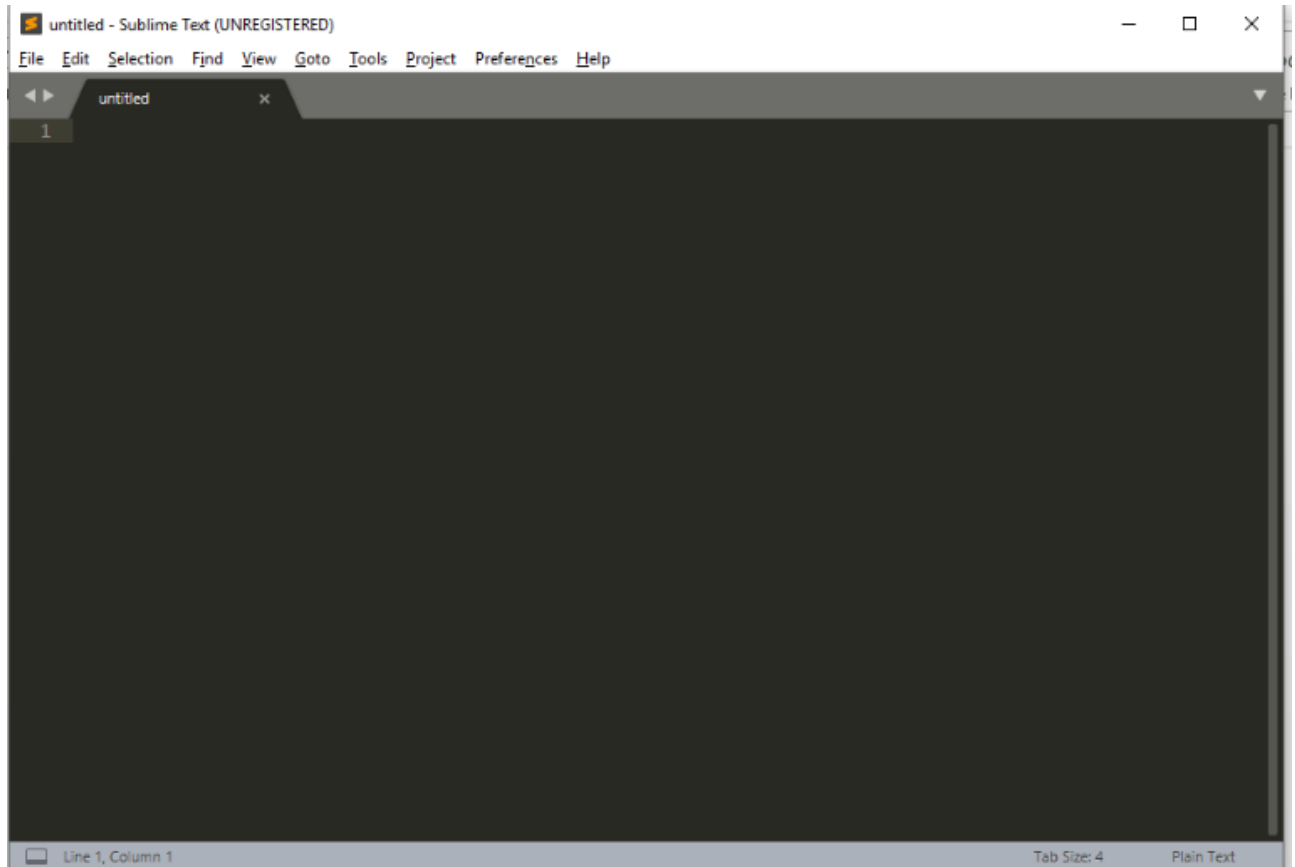
## B. HTML code with Sublime Text-editor.(Recommended after learning basics of HTML)

When you will learn the basics of HTML, then you can use some professional text editors, which will help you to write an efficient and fast code. So to use Sublime Text editors, first it needs to download and install from internet. You can easily download it from this <https://www.sublimetext.com/download> link and can install in your PC.

When installation of Sublime text editor done then you can follow the simple steps to use it:

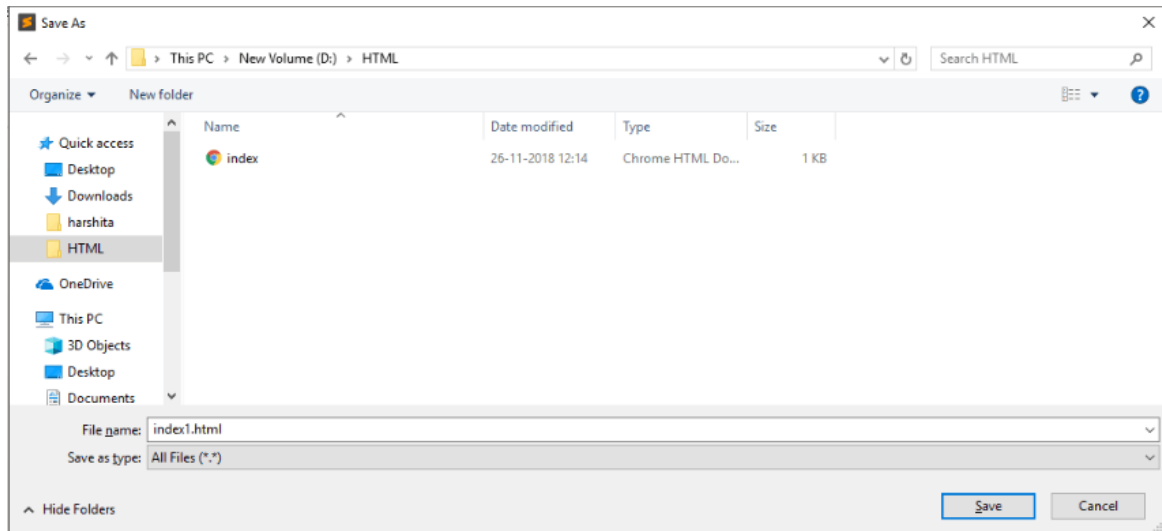
### **Step 1: Open Sublime Text editor(Windows 8):**

To open Sublime Text editor go to **Start screen** --> **type Sublime Text**--> **Open** it. To open a new page press **CTRL+N**.

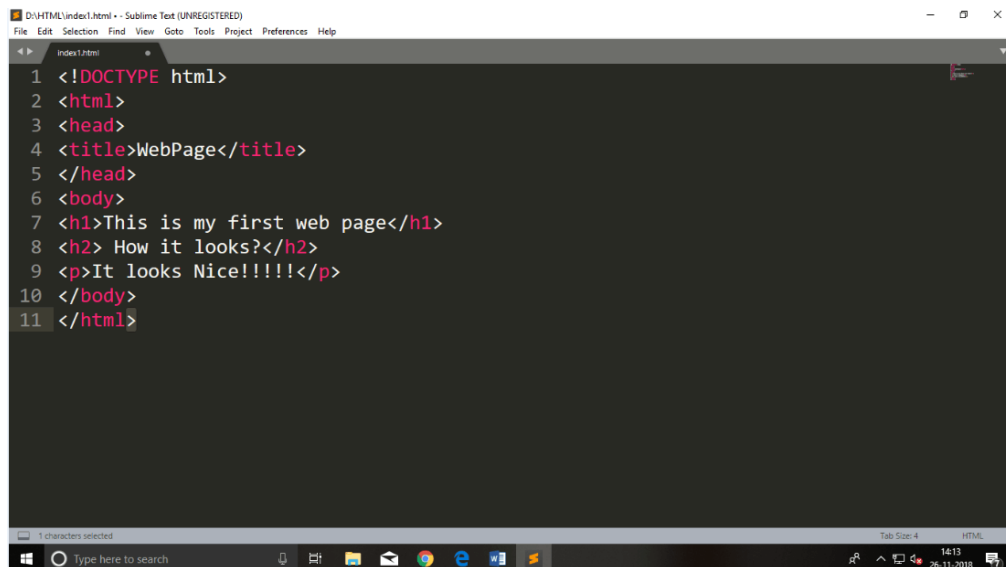


### **Step 2: Save the page before writing any code.**

To save your page in Sublime Text press Ctrl+S or go to File option --> save, to save a file use extension .htm or .html. We recommend to save the file first then write the code because after saving the page sublime text editor will give you suggestions to write code.



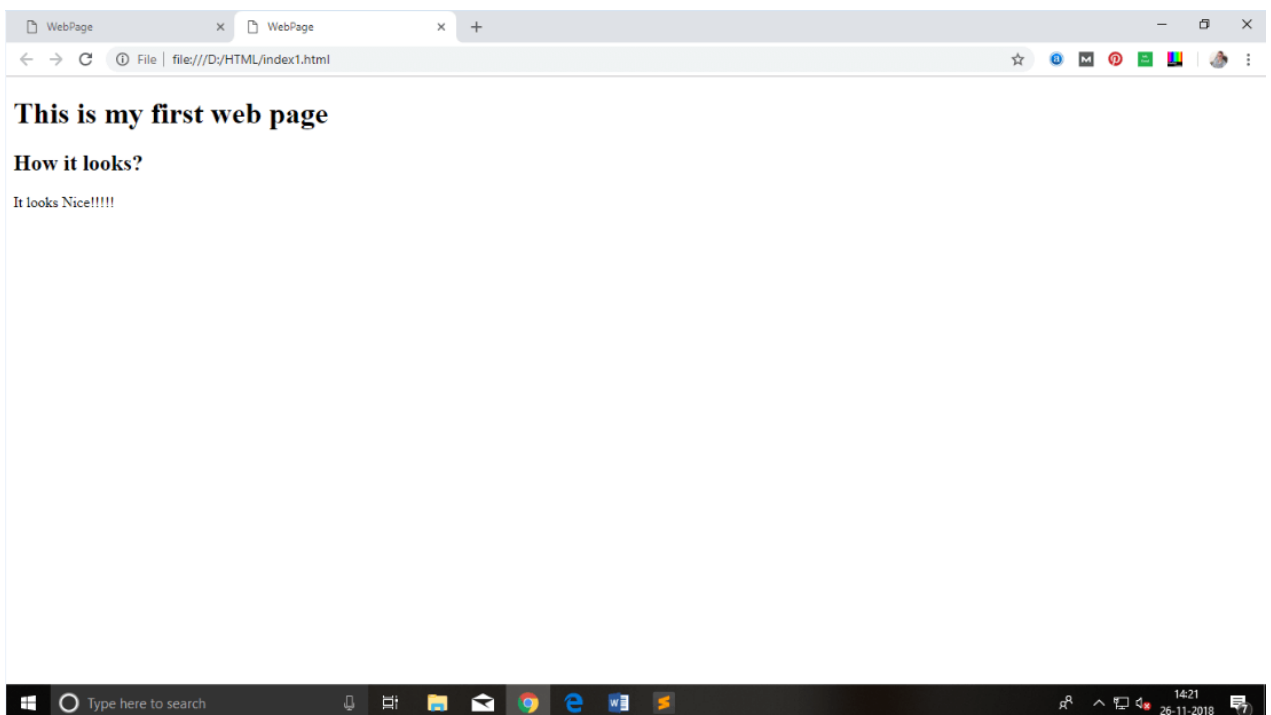
### Step 3: Write the code in Sublime Text editor



### Step 4: Open the HTML page in your Browser

To execute or open this page in Web browser just **right click** by mouse on sublime text page and click on **Open in Browser**.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>WebPage</title>
5 </head>
6 <body>
7 <h1>This is my first web page</h1>
8 <h2> How it looks?</h2>
9 <p>It looks Nice!!!!</p>
10 </body>
11 </html>
```



*Note: You can execute HTML file in any browser, but there are some tags which are not supported by Some Web browser.*

## Building blocks of HTML

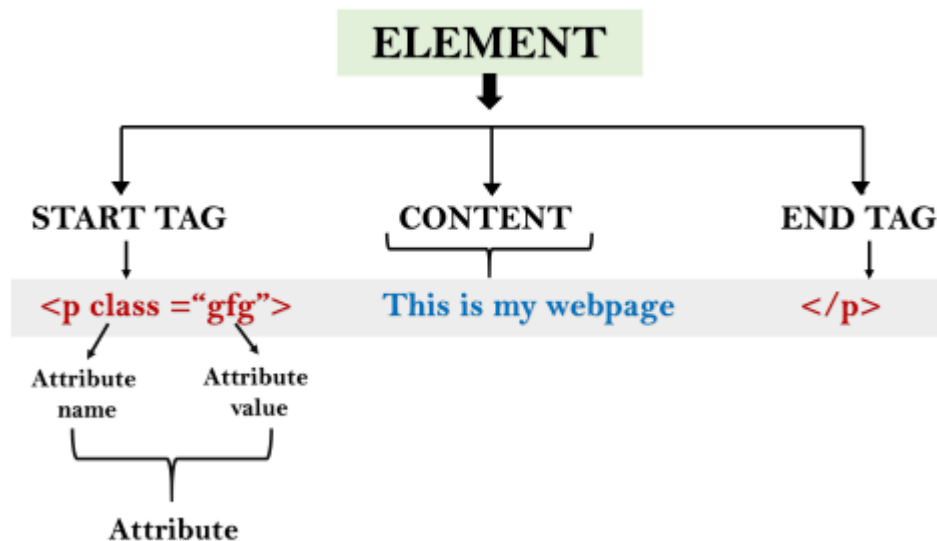
An HTML document consist of its basic building blocks which are:

- **Tags:** An HTML tag surrounds the content and apply meaning to it. It is written between < and > brackets.
- **Attribute:** An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value.



# Syntax

1. `<tag name attribute_name= " attr_value"> content </ tag name>`
  - **Elements:** An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags are termed as HTML elements.



## Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>` The basic building blocks of HTML `</title>`
5. `</head>`
6. `<body>`
7. `<h2>` The building blocks `</h2>`
8. `<p>` This is a paragraph tag `</p>`
9. `<p style="color: red">` The style is attribute of paragraph tag `</p>`
10. `<span>` The element contains tag, attribute and content `</span>`
11. `</body>`
12. `</html>`

[Test it Now](#)

**Output:**

# The building blocks

This is a paragraph tag

The style is attribute of paragraph tag

The element contains tag, attribute and content

## HTML Tags

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

- All HTML tags must enclosed within < > these brackets.
- Every tag in HTML perform different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

---

## Syntax

<tag> content </tag>

---

## HTML Tag Examples

*Note: HTML Tags are always written in lowercase letters. The basic HTML tags are given below:*

<p> Paragraph Tag </p>

## <h2> Heading Tag </h2>

<b> **Bold Tag** </b>

<i> *Italic Tag* </i>

<u> Underline Tag</u>

**Test it Now**

---

## Unclosed HTML Tags

Some HTML tags are not closed, for example br and hr.

<br> **Tag:** br stands for break line, it breaks the line of the code.

<hr> **Tag:** hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

---

## HTML Meta Tags

DOCTYPE, title, link, meta and style

---

## HTML Text Tags

<p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <strong>, <em>, <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <cite>, <q>, <code>, <ins>, <del>, <dfn>, <kbd>, <pre>, <samp>, <var> and <br>

---

## HTML Link Tags

<a> and <base>

---

## HTML Image and Object Tags

<img>, <area>, <map>, <param> and <object>

---

## HTML List Tags

<ul>, <ol>, <li>, <dl>, <dt> and <dd>

---

## HTML Table Tags

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

---

## HTML Form Tags

form, input, textarea, select, option, optgroup, button, label, fieldset and legend

---

## HTML Scripting Tags

script and noscript

*Note: We will see examples using these tags in later chapters.*

## HTML Tags List

Following is the complete list of HTML tags with the description which are arranged alphabetically.




*Note: Here  represents newly added Elements in HTML5.*

### HTML Tags by Alphabets


**ABCDEFGHIJKLMNOPQRSTUVWXYZ**

---

Tag name	Description
<u>&lt;!-- --&gt;</u>	This tag is used to apply comment in an HTML document.
<u>&lt;!DOCTYPE&gt;</u>	This tag is used to specify the version of HTML
A	


<a href="#"><u>&lt;a&gt;</u></a>	It is termed as anchor tag and it creates a hyperlink or link.
<a href="#"><u>&lt;abbr&gt;</u></a>	It defines an abbreviation for a phrase or longer word.
<a href="#"><u>&lt;acronym&gt;</u></a>	It defines acronym for a word. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;address&gt;</u></a>	It defines the author's contact information of the HTML article
<a href="#"><u>&lt;applet&gt;</u></a>	It defines an embedded Java applet. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;area&gt;</u></a>	It defines the area of an image map.
<a href="#"><u>&lt;article&gt;</u></a> 	It defines the self-contained content.
<a href="#"><u>&lt;aside&gt;</u></a> 	It defines content aside from main content. Mainly represented as sidebar.
<a href="#"><u>&lt;audio&gt;</u></a> 	It is used to embed sound content in HTML document.

## B



<a href="#"><u>&lt;b&gt;</u></a>	It is used to make a text bold.
<a href="#"><u>&lt;base&gt;</u></a>	This tag defines the base URL for all relative URL within the document.
<a href="#"><u>&lt;basefont&gt;</u></a>	This tag is used to set default font, size and color for all elements of document. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;bdi&gt;</u></a> 	This tag is used to provide isolation for that part of text which may be formatted in different directions from its surrounding text.
<a href="#"><u>&lt;bdo&gt;</u></a>	It is used to override the current text direction.
<a href="#"><u>&lt;big&gt;</u></a>	This tag is used to make font size one level larger than its surrounding content. <b>(Not supported in HTML5)</b>







<a href="#"><u>&lt;blockquote&gt;</u></a>	It is used to define a content which is taken from another source.
<a href="#"><u>&lt;body&gt;</u></a>	It is used to define the body section of an HTML document.
<a href="#"><u>&lt;br&gt;</u></a>	It is used to apply single line break.
<a href="#"><u>&lt;button&gt;</u></a>	It is used to represent a clickable button


## C

<a href="#"><u>&lt;canvas&gt;</u></a> 	It is used to provide a graphics space within a web document.
<a href="#"><u>&lt;caption&gt;</u></a>	It is used to define a caption for a table.
<a href="#"><u>&lt;center&gt;</u></a>	It is used to align the content in center. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;cite&gt;</u></a>	It is used to define the title of the work, book, website, etc.
<a href="#"><u>&lt;code&gt;</u></a>	It is used to display a part of programming code in an HTML document.
<a href="#"><u>&lt;col&gt;</u></a>	It defines a column within a table which represent common properties of columns and used with the <colgroup> element.
<a href="#"><u>&lt;colgroup&gt;</u></a>	It is used to define group of columns in a table.





## D





<a href="#"><u>&lt;data&gt;</u></a> 	It is used to link the content with the machine-readable translation.
<a href="#"><u>&lt;datalist&gt;</u></a> 	It is used to provide a predefined list for input option.
<a href="#"><u>&lt;dd&gt;</u></a>	It is used to provide definition/description of a term in description list.
<a href="#"><u>&lt;del&gt;</u></a>	It defines a text which has been deleted from the document.




<a href="#"><u>&lt;details&gt;</u></a> 	It defines additional details which user can either view or hide.
<a href="#"><u>&lt;dfn&gt;</u></a>	It is used to indicate a term which is defined within a sentence/phrase.
<a href="#"><u>&lt;dialog&gt;</u></a> 	It defines a dialog box or other interactive components.
<a href="#"><u>&lt;dir&gt;</u></a>	It is used as container for directory list of files. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;div&gt;</u></a>	It defines a division or section within HTML document.
<a href="#"><u>&lt;dl&gt;</u></a>	It is sued to define a description list.
<a href="#"><u>&lt;dt&gt;</u></a>	It is used to define a term in description list.
<b>E</b>	
<a href="#"><u>&lt;em&gt;</u></a>	It is used to emphasis the content applied within this element.
<a href="#"><u>&lt;embed&gt;</u></a> 	It is used as embedded container for external file/application/media, etc.
<b>F</b>	
<a href="#"><u>&lt;fieldset&gt;</u></a>	It is used to group related elements/labels within a web form.
<a href="#"><u>&lt;figcaption&gt;</u></a> 	It is used to add a caption or explanation for the <figure> element.
<a href="#"><u>&lt;figure&gt;</u></a> 	It is used to define the self-contained content, and s mostly refer as single unit.
<a href="#"><u>&lt;font&gt;</u></a>	It defines the font, size, color, and face for the content. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;footer&gt;</u></a> 	It defines the footer section of a webpage.
<a href="#"><u>&lt;form&gt;</u></a>	It is used to define an HTML form.
<a href="#"><u>&lt;frame&gt;</u></a>	It defines a particular area of webpage which can contain another HTML file.


	<b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;frameset&gt;</u></a>	It defines group of Frames. <b>(Not supported in HTML5)</b>
<b>H</b>	
<a href="#"><u>&lt;h1&gt; to &lt;h6&gt;</u></a>	It defines headings for an HTML document from level 1 to level 6.
<a href="#"><u>&lt;head&gt;</u></a>	It defines the head section of an HTML document.
<a href="#"><u>&lt;header&gt;</u></a> 	It defines the header of a section or webpage.
<a href="#"><u>&lt;hr&gt;</u></a>	It is used to apply thematic break between paragraph-level elements.
<a href="#"><u>&lt;html&gt;</u></a>	It represents root of an HTML document.
<b>I</b>	
<a href="#"><u>&lt;i&gt;</u></a>	It is used to represent a text in some different voice.
<a href="#"><u>&lt;iframe&gt;</u></a>	It defines an inline frame which can embed other content.
<a href="#"><u>&lt;img&gt;</u></a>	It is used to insert an image within an HTML document.
<a href="#"><u>&lt;input&gt;</u></a>	It defines an input field within an HTML form.
<a href="#"><u>&lt;ins&gt;</u></a>	It represent text that has been inserted within an HTML document.
<a href="#"><u>&lt;isindex&gt;</u></a>	It is used to display search string for current document. <b>(Not supported in HTML5)</b>
<b>K</b>	
<a href="#"><u>&lt;kbd&gt;</u></a>	It is used to define keyboard input.
<b>L</b>	





<a href="#"><u>&lt;label&gt;</u></a>	It defines a text label for the input field of form.
<a href="#"><u>&lt;legend&gt;</u></a>	It defines a caption for content of <fieldset>
<a href="#"><u>&lt;li&gt;</u></a>	It is used to represent items in list.
<a href="#"><u>&lt;link&gt;</u></a>	It represents a relationship between current document and an external resource.
<b>M</b>	
<a href="#"><u>&lt;main&gt;</u></a> 	It represents the main content of an HTML document.
<a href="#"><u>&lt;map&gt;</u></a>	It defines an image map with active areas.
<a href="#"><u>&lt;mark&gt;</u></a> 	It represents a highlighted text.
<a href="#"><u>&lt;marquee&gt;</u></a>	It is used to insert the scrolling text or an image either horizontally or vertically. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;menu&gt;</u></a>	It is used for creating a menu list of commands.
<a href="#"><u>&lt;meta&gt;</u></a>	It defines metadata of an HTML document.
<a href="#"><u>&lt;meter&gt;</u></a> 	It defines scalar measurement with known range or fractional value.
<b>N</b>	
<a href="#"><u>&lt;nav&gt;</u></a> 	It represents section of page to represent navigation links.
<a href="#"><u>&lt;noframes&gt;</u></a>	It provides alternate content to represent in browser which does not support the <frame> elements. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;noscript&gt;</u></a>	It provides an alternative content if a script type is not supported in browser.
<b>O</b>	

<a href="#"><u>&lt;object&gt;</u></a>	It is used to embed an object in HTML file.
<a href="#"><u>&lt;ol&gt;</u></a>	It defines an ordered list of items.
<a href="#"><u>&lt;optgroup&gt;</u></a>	It is used to group the options of a drop-down list.
<a href="#"><u>&lt;option&gt;</u></a>	It is used to define options or items in a drop-down list.
<a href="#"><u>&lt;output&gt;</u></a> 	It is used as container element which can show result of a calculation.
<b>P</b>	
<a href="#"><u>&lt;p&gt;</u></a>	It represents a paragraph in an HTML document.
<a href="#"><u>&lt;param&gt;</u></a>	It defines parameter for an <object> element
<a href="#"><u>&lt;picture&gt;</u></a> 	It defines more than one source element and one image element.
<a href="#"><u>&lt;pre&gt;</u></a>	It defines preformatted text in an HTML document.
<a href="#"><u>&lt;progress&gt;</u></a> 	It defines the progress of a task within HTML document.
<b>Q</b>	
<a href="#"><u>&lt;q&gt;</u></a>	It defines short inline quotation.
<b>R</b>	
<a href="#"><u>&lt;rp&gt;</u></a> 	It defines an alternative content if browser does not supports ruby annotations.
<a href="#"><u>&lt;rt&gt;</u></a>	It defines explanations and pronunciations in ruby annotations.
<a href="#"><u>&lt;ruby&gt;</u></a>	It is used to represent ruby annotations.
<b>S</b>	

<a href="#"><u>&lt;s&gt;</u></a>	It render text which is no longer correct or relevant.
<a href="#"><u>&lt;samp&gt;</u></a>	It is used to represent sample output of a computer program.
<a href="#"><u>&lt;script&gt;</u></a>	It is used to declare the JavaScript within HTML document.
<a href="#"><u>&lt;section&gt;</u></a> 	It defines a generic section for a document.
<a href="#"><u>&lt;select&gt;</u></a>	It represents a control which provides a menu of options.
<a href="#"><u>&lt;small&gt;</u></a>	It is used to make text font one size smaller than document's base font size.
<a href="#"><u>&lt;source&gt;</u></a> 	It defines multiple media recourses for different media element such as <picture>, <video>, and <audio> element.
<a href="#"><u>&lt;span&gt;</u></a>	It is used for styling and grouping inline.
<a href="#"><u>&lt;strike&gt;</u></a>	It is used to render strike through the text. <b>(Not supported in HTML5)</b>
<a href="#"><u>&lt;strong&gt;</u></a>	It is used to define important text.
<a href="#"><u>&lt;style&gt;</u></a>	It is used to contain style information for an HTML document.
<a href="#"><u>&lt;sub&gt;</u></a>	It defines a text which displays as a subscript text.
<a href="#"><u>&lt;summary&gt;</u></a> 	It defines summary which can be used with <details> tag.
<a href="#"><u>&lt;sup&gt;</u></a>	It defines a text which represent as superscript text.
<a href="#"><u>&lt;svg&gt;</u></a>	It is used as container of SVG (Scalable Vector Graphics).
<b>T</b>	
<a href="#"><u>&lt;table&gt;</u></a>	It is used to present data in tabular form or to create a table within HTML document.
<a href="#"><u>&lt;tbody&gt;</u></a>	It represents the body content of an HTML table and used along with <thead> and

	<tfoot>.
<a href="#"><u>&lt;td&gt;</u></a>	It is used to define cells of an HTML table which contains table data
<a href="#"><u>&lt;template&gt;</u></a>	It is used to contain the client side content which will not display at time of page load and may render later using JavaScript.
<a href="#"><u>&lt;textarea&gt;</u></a>	It is used to define multiple line input, such as comment, feedback, and review, etc.
<a href="#"><u>&lt;tfoot&gt;</u></a>	It defines the footer content of an HTML table.
<a href="#"><u>&lt;th&gt;</u></a>	It defines the head cell of an HTML table.
<a href="#"><u>&lt;thead&gt;</u></a>	It defines the header of an HTML table. It is used along with <tbody> and <tfoot> tags.
<a href="#"><u>&lt;time&gt;</u></a> 	It is used to define data/time within an HTML document.
<a href="#"><u>&lt;title&gt;</u></a>	It defines the title or name of an HTML document.
<a href="#"><u>&lt;tr&gt;</u></a>	It defines the row cells in an HTML table
<a href="#"><u>&lt;track&gt;</u></a>	It is used to define text tracks for <audio> and <video> elements.
<a href="#"><u>&lt;tt&gt;</u></a>	It is used to define teletype text. <b>(Not supported in HTML5)</b>
U	
<a href="#"><u>&lt;u&gt;</u></a>	It is used to render enclosed text with an underline.
<a href="#"><u>&lt;ul&gt;</u></a>	It defines unordered list of items.
V	
<a href="#"><u>&lt;var&gt;</u></a>	It defines variable name used in mathematical or programming context.

<code>&lt;video&gt;</code> 	It is used to embed a video content with an HTML document
<h2>W</h2>	
<code>&lt;wbr&gt;</code> 	It defines a position within text where break line is possible

## HTML Attribute

- HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element.
- Each element or tag can have attributes, which defines the behaviour of that element.
- Attributes should always be applied with start tag.
- The Attribute should always be applied with its name and value pair.
- The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.
- You can add multiple attributes in one HTML element, but need to give space between two attributes.

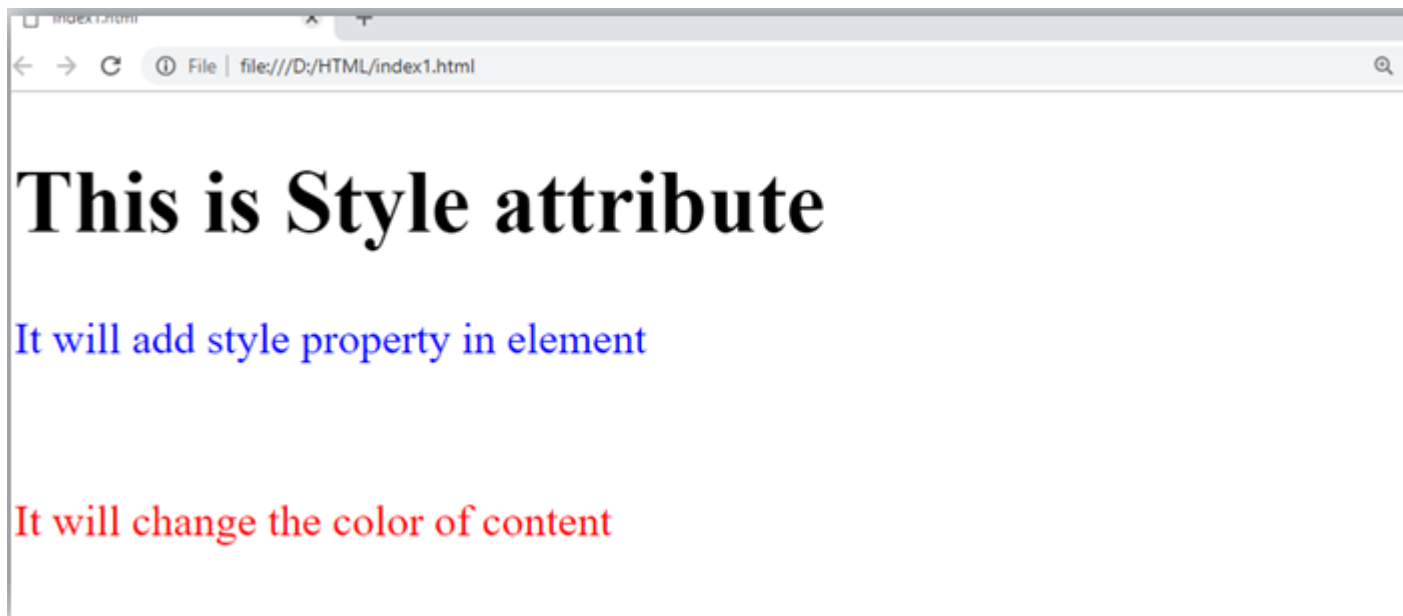
## Syntax

1. `<element attribute_name="value">content</element>`

## Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `</head>`
5. `<body>`
6. `<h1> This is Style attribute</h1>`
7. `<p style="height: 50px; color: blue">It will add style property in element</p>`
8. `<p style="color: red">It will change the color of content</p>`
9. `</body>`
10. `</html>`

**Output:**



### Explanation of above example:

1. `<p style="height: 50px; color: blue">` It will add style property in element `</p>`

In the above statement, we have used paragraph tags in which we have applied style attribute. This attribute is used for applying CSS property on any HTML element. It provides height to paragraph element of 50px and turns its colour to blue.

1. `<p style="color: red">` It will change the color of content `</p>`

In the above statement we have again used style attribute in paragraph tag, which turns its colour red.

*Note: There are some commonly used attributes are given below, and the complete list and explanation of all attributes are given in HTML attributes List.*

## The title attribute in HTML

**Description:** The title attribute is used as text tooltip in most of the browsers. It displays its text when the user moves the cursor over a link or any text. You can use it with any text or link to show the description about that link or text. In our example, we are taking this with paragraph tag and heading tag.

### Example

**With `<h1>` tag:**

1. `<h1 title="This is heading tag">` Example of title attribute `</h1>`

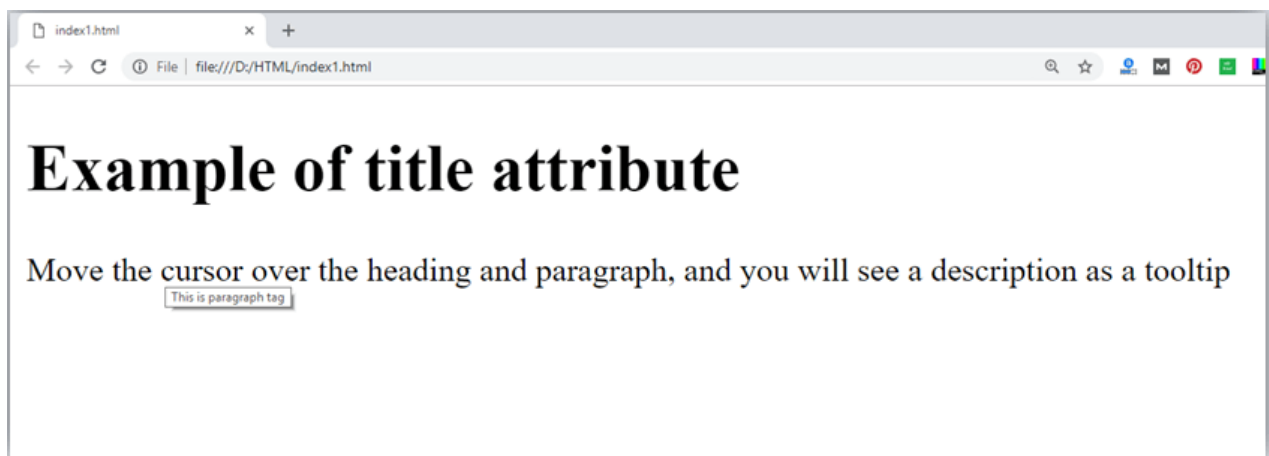
**With `<p>` tag:**

1. `<p title="This is paragraph tag">` Move the cursor over the heading and paragraph, and you will see a description as a tooltip `</p>`

#### Code:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `</head>`
5. `<body>`
- 6.
7. `<h1 title="This is heading tag">` Example of title attribute `</h1>`
8. `<p title="This is paragraph tag">` Move the cursor over the heading and paragraph, and you will see a description as a tooltip `</p>`
- 9.
10. `</body>`
11. `</html>`

#### Output:



---

## The href attribute in HTML

**Description:** The href attribute is the main attribute of `<a>` anchor tag. This attribute gives the link address which is specified in that link. **The href attribute provides the hyperlink, and if it is blank, then it will remain in same page.**

### Example

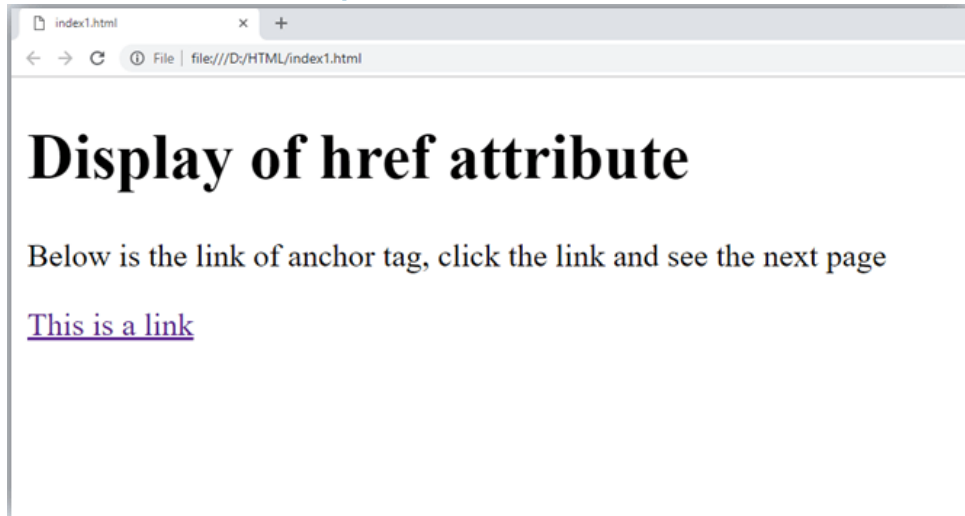
#### With link address:

1. `<a href="https://www.javatpoint.com/html-anchor">` This is a link `</a>`

## Test it Now

### Without link address:

1. `<a href="">This is a link</a>`



## The src Attribute

The **src** attribute is one of the important and required attribute of **<img>** element. It is source for the image which is required to display on browser. This attribute can contain image in same directory or another directory. The image name or source should be correct else browser will not display the image.

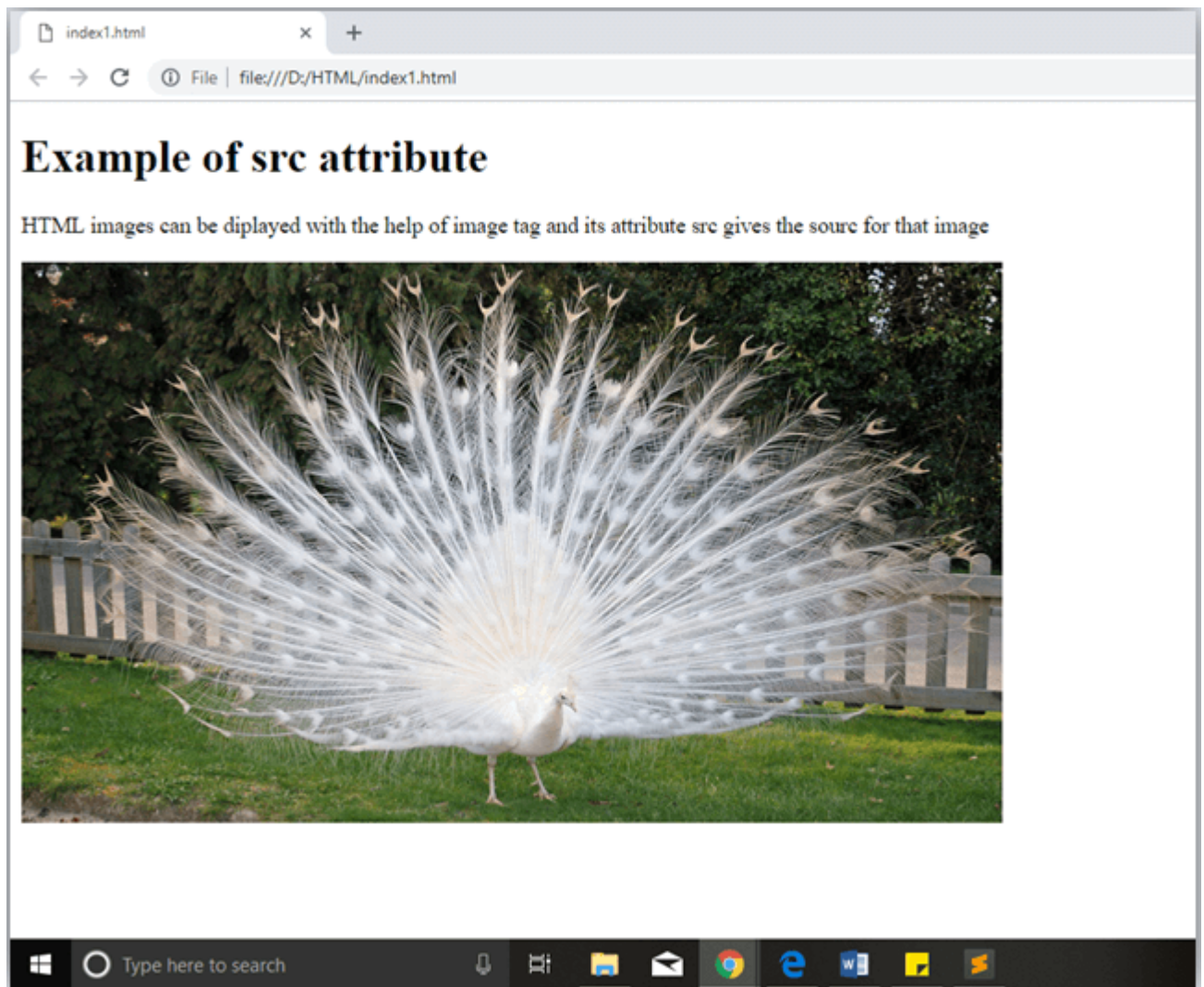
## Example

1. ``

*Note: The above example also have height and width attribute, which define the height and width of image on web page.*

### Output:





## Quotes: single quotes or double quotes?

In this chapter you have seen that, we have used attribute with double quotes, but some people might use single quotes in HTML. So use of single quotes with HTML attribute, is also allowed. The following both statements are absolutely fine.

1. `<a href="https://www.javatpoint.com">A link to HTML.</a>`
2. `<a href='https://www.javatpoint.com'>A link to HTML.</a>`

IN HTML5, you can also omit use of quotes around attribute values.

1. `<a href=https://www.javatpoint.com>A link to HTML.</a>`

## HTML Elements

An HTML file is made of elements. These elements are responsible for creating web pages and define content in that webpage. An element in HTML usually consist of a start tag `<tag name>`, close tag `</tag name>` and content inserted between

them. **Technically, an element is a collection of start tag, attributes, end tag, content between them.**

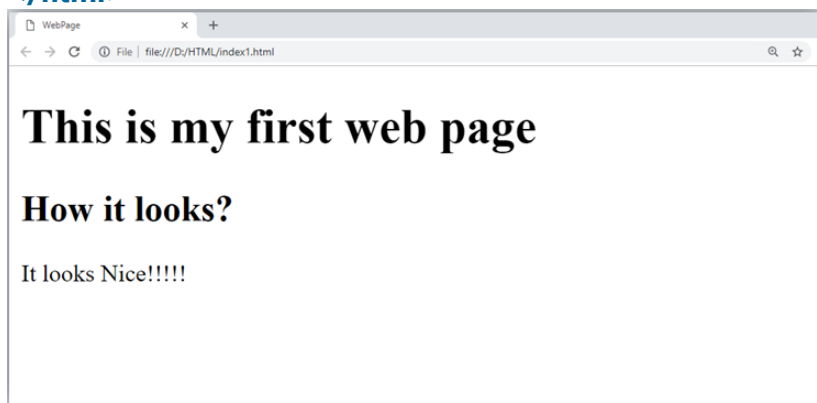
*Note: Some elements does not have end tag and content, these elements are termed as empty elements or self-closing element or void elements.*

Such as:

1. `<p> Hello world!!! </p>`

## Example

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>WebPage</title>`
5. `</head>`
6. `<body>`
7. `<h1>This is my first web page</h1>`
8. `<h2> How it looks?</h2>`
9. `<p>It looks Nice!!!!</p>`
10. `</body>`
11. `</html>`



- All the content written between body elements are visible on web page.

**Void element:** All the elements in HTML do not require to have start tag and end tag, some elements does not have content and end tag such elements are known as Void elements or empty elements. **These elements are also called as unpaired tag.**

**Some Void elements are** `<br>` (represents a line break) , `<hr>`(represents a horizontal line), etc.

**Nested HTML Elements:** HTML can be nested, which means an element can contain another element.

---

# Block-level and Inline HTML elements

For the default display and styling purpose in HTML, all the elements are divided into two categories:

- Block-level element
- Inline element

---

## Block-level element:

- These are the elements, which structure main part of web page, by dividing a page into coherent blocks.
- A block-level element always start with new line and takes the full width of web page, from left to right.
- These elements can contain block-level as well as inline elements.

Following are the block-level elements in HTML.

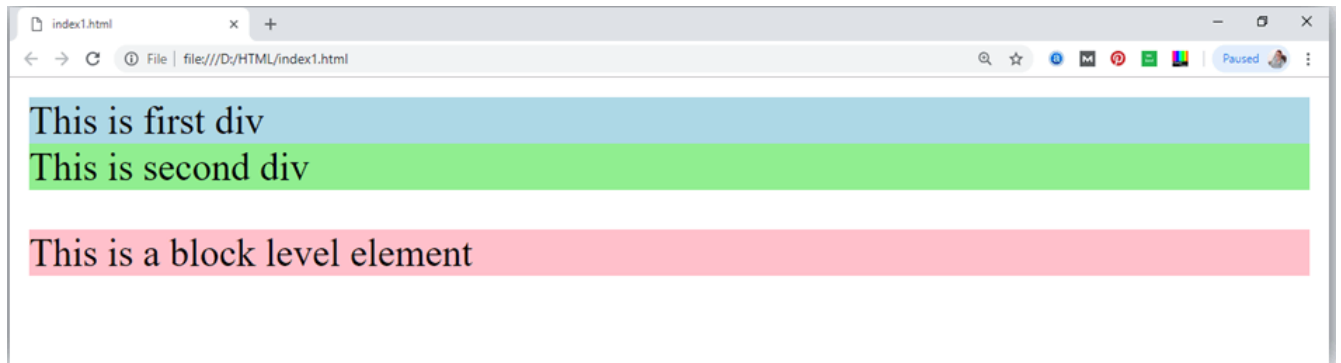
<address>, <article>, <aside>, <blockquote>, <canvas>, <dd>, <div>, <dl>, <dt>, <fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1>-<h6>, <header>, <hr>, <li>, <main>, <nav>, <noscript>, <ol>, <output>, <p>, <pre>, <section>, <table>, <tfoot>, <ul> and <video>.

*Note: All these elements are described in later chapters.*

## Example:

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.   </head>
5. <body>
6.   <div style="background-color: lightblue">This is first div</div>
7.   <div style="background-color: lightgreen">This is second div</div>
8.   <p style="background-color: pink">This is a block level element</p>
9. </body>
10. </html>

**Output:**



In the above example we have used

tag, which defines a section in a web page, and takes full width of page.

We have used style attribute which is used to styling the HTML content, and the background color are showing that it's a block level element.

---

## Inline elements:

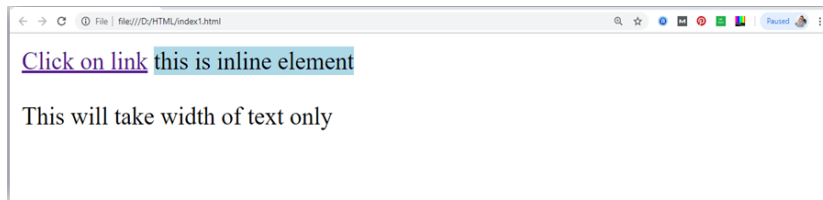
- Inline elements are those elements, which differentiate the part of a given text and provide it a particular function.
- These elements does not start with new line and take width as per requirement.
- The Inline elements are mostly used with other elements.

<a>, <abbr>, <acronym>, <b>, <bdo>, <big>, <br>, <button>, <cite>, <code>, <dfn>, <em>, <i>, <img>, <input>, <kbd>, <label>, <map>, <object>, <q>, <samp>, <script>, <select>, <small>, <span>, <strong>, <sub>, <sup>, <textarea>, <time>, <tt>, <var>.

## Example:

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.     </head>
5. <body>
6.     <a href="https://www.javatpoint.com/html-tutorial">Click on link</a>
7.     <span style="background-color: lightblue">this is inline element</span>
8.     <p>This will take width of text only</p>
9. </body>
10. </html>

**Output:**



Following is the list of the some main elements used in HTML:

Start tag	Content	End tag	Description
<code>&lt;h1&gt; ..... &lt;h6&gt;</code>	These are headings of HTML	<code>&lt;/h1&gt;??.. &lt;/h6&gt;</code>	These elements are used to provide the headings of page.
<code>&lt;p&gt;</code>	This is the paragraph	<code>&lt;/p&gt;</code>	This element is used to display a content in form of paragraph.
<code>&lt;div&gt;</code>	This is div section	<code>&lt;/div&gt;</code>	This element is used to provide a section in web page.
<code>&lt;br&gt;</code>			This element is used to provide a line break. ( void element)
<code>&lt;hr&gt;</code>			This element is used to provide a horizontal line. (void element)

## HTML Formatting

**HTML Formatting** is a process of formatting text for better look and feel. HTML provides us ability to format text without using CSS. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined. There are almost 14 options available that how text appears in HTML and XHTML.

In HTML the formatting tags are divided into two categories:

- Physical tag: These tags are used to provide the visual appearance to the text.
- Logical tag: These tags are used to add some logical or semantic value to the text.

*NOTE: There are some physical and logical tags which may give same visual appearance, but they will be different in semantics.*

Here, we are going to learn 14 HTML formatting tags. Following is the list of HTML formatting text.

Element name	Description
<b>	This is a physical tag, which is used to bold the text written between it.
<strong>	This is a logical tag, which tells the browser that the text is important.
<i>	This is a physical tag which is used to make text italic.
<em>	This is a logical tag which is used to display content in italic.
<mark>	This tag is used to highlight text.
<u>	This tag is used to underline text written between it.
<tt>	This tag is used to appear a text in teletype. (not supported in HTML5)
<strike>	This tag is used to draw a strikethrough on a section of text. (Not supported in HTML5)
<sup>	It displays the content slightly above the normal line.
<sub>	It displays the content slightly below the normal line.
<del>	This tag is used to display the deleted content.
<ins>	This tag displays the content which is added
<big>	This tag is used to increase the font size by one conventional unit.
<small>	This tag is used to decrease the font size by one unit from base font size.

## 1) Bold Text

HTML `<b>` and `<strong>` formatting elements

The HTML `<b>` element is a physical tag which display text in bold font, without any logical importance. If you write anything within `<b>.....</b>` element, is shown in bold letters.

See this example:

1. `<p> <b>`Write Your First Paragraph in bold text.`</b> </p>`

**Output:**

**Write Your First Paragraph in bold text.**

The HTML `<strong>` tag is a logical tag, which displays the content in bold font and informs the browser about its logical importance. If you write anything between `<strong>??????. </strong>`, is shown important text.

See this example:

1. `<p> <strong>`This is an important content`</strong>`, and this is normal content`</p>`

**Output:**

**This is an important content, and this is normal content**

## Example

1. `<!DOCTYPE html>`
  2. `<html>`
  3. `<head>`
  4. `<title>`formatting elements`</title>`
  5. `</head>`
  6. `<body>`
  7. `<h1>`Explanation of formatting element`</h1>`
  8. `<p> <strong>`This is an important content`</strong>`, and this is normal content`</p>`
  9. `</body>`
  10. `</html>`
- 

## 2) Italic Text

HTML `<i>` and `<em>` formatting elements

The HTML `<i>` element is physical element, which display the enclosed content in italic font, without any added importance. If you write anything within `<i>.....</i>` element, is shown in italic letters.

See this example:

1. `<p> <i>`Write Your First Paragraph in italic text.`</i> </p>`

**Output:**

*Write Your First Paragraph in italic text.*

The HTML `<em>` tag is a logical element, which will display the enclosed content in italic font, with added semantics importance.

**See this example:**

1. `<p> <em>`This is an important content`</em>`, which displayed in italic font.`</p>`

**Output:**

*This is an important content, which displayed in italic font.*

1. `<!DOCTYPE html>`
  2. `<html>`
  3. `<head>`
  4. `<title>`formatting elements`</title>`
  5. `</head>`
  6. `<body>`
  7. `<h1>`Explanation of italic formatting element`</h1>`
  8. `<p> <em>`This is an important content`</em>`, which displayed in italic font.`</p>`
  9. `</body>`
  10. `</html>`
- 

### 3) HTML Marked formatting

If you want to mark or highlight a text, you should write the content within `<mark>.....</mark>`.

See this example:

1. `<h2>` I want to put a `<mark>` Mark`</mark>` on your face`</h2>`

**Output:**



# I want to put a Mark on your face

---

## 4) Underlined Text

If you write anything within `<u>.....</u>` element, is shown in underlined text.

See this example:

1. `<p> <u>Write Your First Paragraph in underlined text.</u> </p>`

**Output:**

Write Your First Paragraph in underlined text.

---

## 5) Strike Text

Anything written within `<strike>.....</strike>` element is displayed with strikethrough. It is a thin line which cross the statement.

See this example:

1. `<p> <strike>Write Your First Paragraph with strikethrough</strike>.</p>`

**Output:**

~~Write Your First Paragraph with strikethrough.~~

---

## 6) Monospaced Font

If you want that each letter has the same width then you should write the content within `<tt>.....</tt>` element.

Note: We know that most of the fonts are known as variable-width fonts because different letters have different width. (for example: 'w' is wider than 'i'). Monospaced Font provides similar space among every letter.

See this example:

1. `<p>Hello <tt>Write Your First Paragraph in monospaced font.</tt> </p>`

**Output:**

Hello Write Your First Paragraph in monospaced font.

---

## 7) Superscript Text

If you put the content within `<sup>.....</sup>` element, is shown in superscript; means it is displayed half a character's height above the other characters.

See this example:

1. `<p>Hello <sup>Write Your First Paragraph in superscript.</sup> </p>`

**Output:**

Hello Write Your First Paragraph in superscript.

---

## 8) Subscript Text

If you put the content within `<sub>.....</sub>` element, is shown in subscript ; means it is displayed half a character's height below the other characters.

See this example:

1. `<p>Hello <sub>Write Your First Paragraph in subscript.</sub> </p>`

**Output:**

Hello Write Your First Paragraph in subscript.

---

## 9) Deleted Text

Anything that puts within `<del>.....</del>` is displayed as deleted text.

See this example:

1. `<p>Hello <del>Delete your first paragraph.</del> </p>`

**Output:**

Hello

---

## 10) Inserted Text

Anything that puts within `<ins>.....</ins>` is displayed as inserted text.

See this example:

1. `<p> <del>Delete your first paragraph.</del> <ins>Write another paragraph.</ins> </p>`

**Output:**

Write another paragraph.

---

## 11) Larger Text

If you want to put your font size larger than the rest of the text then put the content within `<big>.....</big>`. It increase one font size larger than the previous one.

See this example:

1. `<p>Hello <big>Write the paragraph in larger font.</big> </p>`

**Output:**

Hello Write the paragraph in larger font.

---

## 12) Smaller Text

If you want to put your font size smaller than the rest of the text then put the content within `<small>.....</small>` tag. It reduces one font size than the previous one.

See this example:

1. `<p>Hello <small>Write the paragraph in smaller font.</small> </p>`

**Output:**

Hello Write the paragraph in smaller font.

---

## HTML Heading

A HTML heading or HTML h tag can be defined as a title or a subtitle which you want to display on the webpage. When you place the text within the heading tags

`<h1>.....</h1>`, it is displayed on the browser in the bold format and size of the text depends on the number of heading.

There are six different HTML headings which are defined with the `<h1>` to `<h6>` tags, from highest level h1 (main heading) to the least level h6 (least important heading).

h1 is the largest heading tag and h6 is the smallest one. So h1 is used for most important heading and h6 is used for least important.

**Headings in HTML helps the search engine to understand and index the structure of web page.**

*Note: The main keyword of the whole content of a webpage should be display by h1 heading tag.*

See this example:

1. `<h1>`Heading no. 1`</h1>`
2. `<h2>`Heading no. 2`</h2>`
3. `<h3>`Heading no. 3`</h3>`
4. `<h4>`Heading no. 4`</h4>`
5. `<h5>`Heading no. 5`</h5>`
6. `<h6>`Heading no. 6`</h6>`

**Output:**

# Heading no. 1

## Heading no. 2

### Heading no. 3

#### Heading no. 4

##### Heading no. 5

###### Heading no. 6

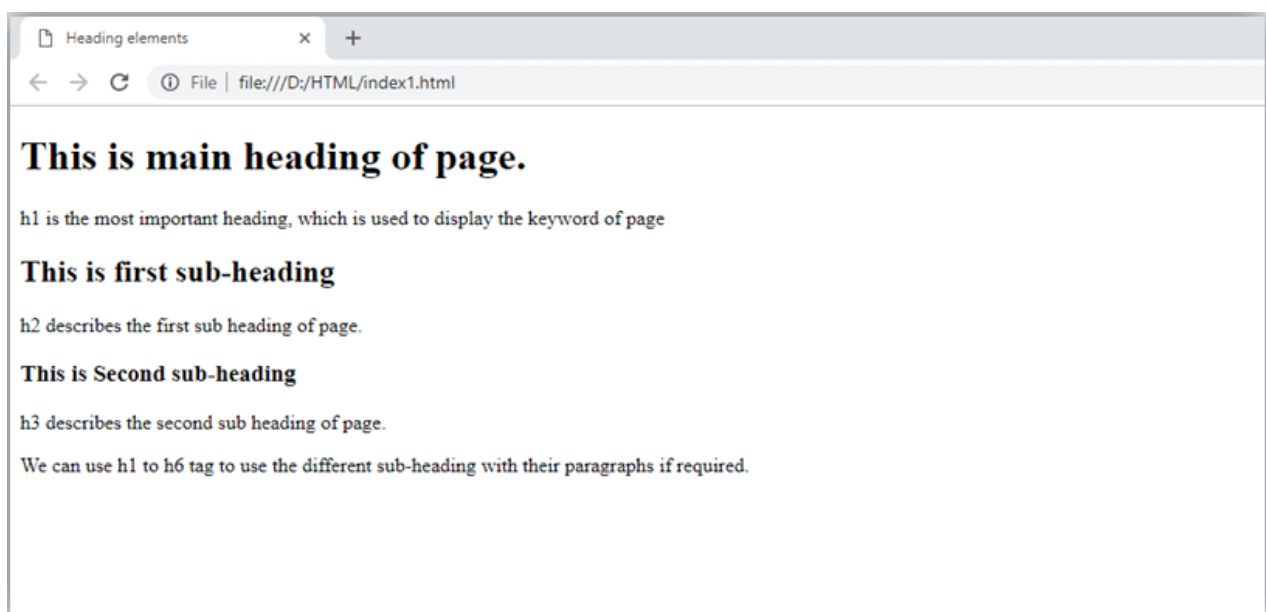
*Heading elements (h1....h6) should be used for headings only. They should not be used just to make text bold or big.*

- **HTML headings can also be used with nested elements. Following are different codes to display the way to use heading elements.**

## Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>`Heading elements`</title>`
5. `</head>`
6. `<body>`
7. `<h1>`This is main heading of page. `</h1>`
8. `<p>`h1 is the most important heading, which is used to display the keyword of page `</p>`
9. `<h2>`This is first sub-heading`</h2>`
10. `<p>`h2 describes the first sub heading of page. `</p>`
11. `<h3>`This is Second sub-heading`</h3>`
12. `<p>`h3 describes the second sub heading of page.`</p>`
13. `<p>`We can use h1 to h6 tag to use the different sub-heading with their paragraphs if
14. required.
15. `</p>`
16. `</body>`
17. `</html>`

## Output:



---

# HTML Paragraph

HTML paragraph or HTML p tag is used to define a paragraph in a webpage. Let's take a simple example to see how it work. It is a notable point that a browser itself add an empty line before and after a paragraph. An HTML <p> tag indicates starting of new paragraph.

*Note: If we are using various <p> tags in one HTML file then browser automatically adds a single blank line between the two paragraphs.*

See this example:

1. `<p>`This is first paragraph.`</p>`
2. `<p>`This is second paragraph.`</p>`
3. `<p>`This is third paragraph.`</p>`

Output:

This is first paragraph.

This is second paragraph.

This is third paragraph.

---

## Space inside HTML Paragraph

If you put a lot of spaces inside the HTML p tag, browser removes extra spaces and extra line while displaying the page. The browser counts number of spaces and lines as a single one.

1. `<p>`
2. I am
3. going to provide
4. you a tutorial on HTML
5. and hope that it will
6. be very beneficial for you.
7. `</p>`
8. `<p>`
9. Look, I put here a lot
10. of spaces                      but                      I know, Browser will ignore it.
11. `</p>`
12. `<p>`
13. You cannot determine the display of HTML`</p>`
14. `<p>`because resized windows may create different result.
15. `</p>`

Output:

I am going to provide you a tutorial on HTML and hope that it will be very beneficial for you.

Look, I put here a lot of spaces but I know, Browser will ignore it.

You cannot determine the display of HTML

because resized windows may create different result.

As you can see, all the extra lines and unnecessary spaces are removed by the browser.

---

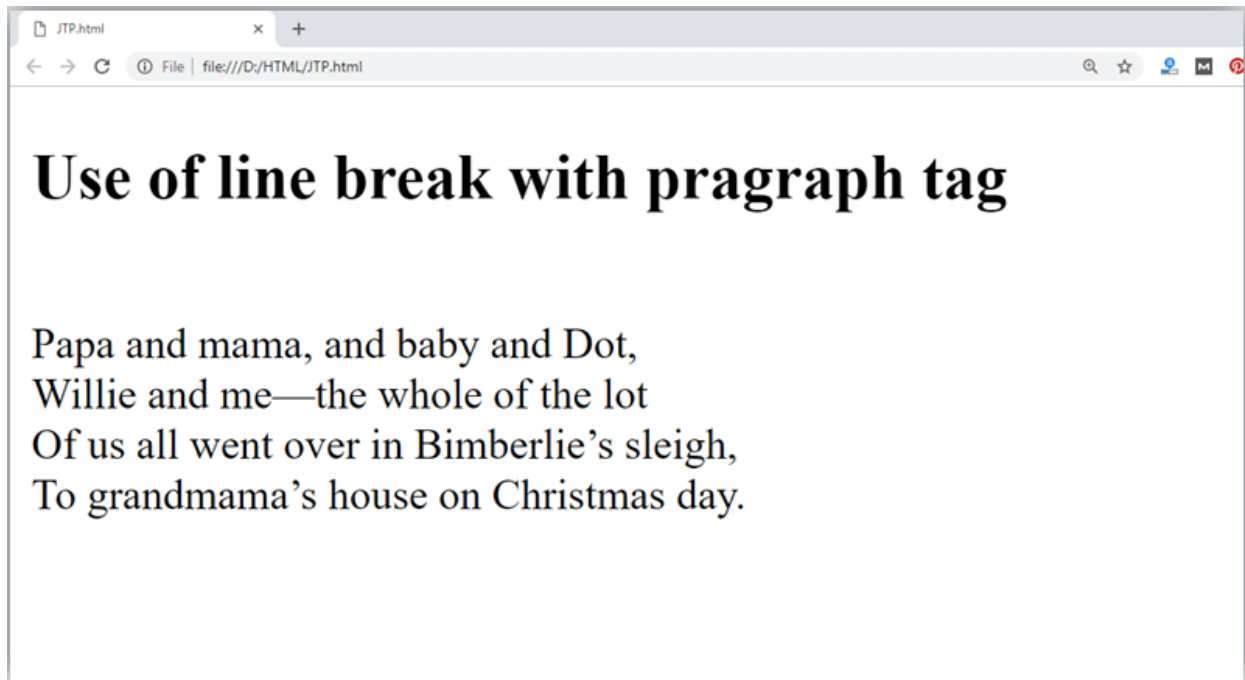
## How to Use <br> and <hr> tag with paragraph?

An HTML <br> tag is used for line break and it can be used with paragraph elements. Following is the example to show how to use <br> with <p> element.

### Example:

1. <!DOCTYPE html>
2. <html>
3.     <head>
4.     </head>
5.     <body>
6.         <h2> Use of line break with paragraph tag</h2>
7.         <p><br>Papa and mama, and baby and Dot,
8.         <br>Willie and me?the whole of the lot
9.         <br>Of us all went over in Bimberlie's sleigh,
10.         <br>To grandmama's house on Christmas day.
11.         </p>
12.     </body>
13. </html>

### Output:



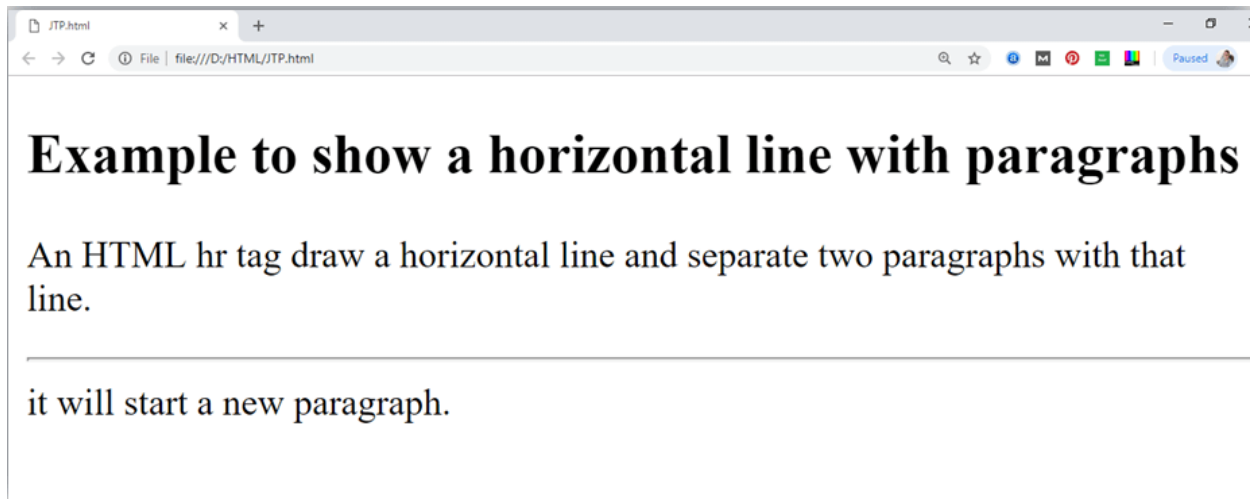
An HTML `<hr>` tag is used to apply a horizontal line between two statements or two paragraphs. Following is the example which is showing use of `<hr>` tag with paragraph.

**Example:**

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `</head>`
5. `<body>`
6. `<h2>` Example to show a horizontal line with paragraphs`</h2>`
7. `<p>` An HTML hr tag draw a horizontal line and separate two paragraphs with that line.`<hr>` it will start a new paragraph.
8. `</p>`
9. `</body>`
10. `</html>`

**Output:**





## What is CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

## What does CSS do

- You can add new looks to your old HTML documents.
- You can completely change the look of your website with only a few changes in CSS code.

## Why use CSS

These are the three major benefits of CSS:

### 1) Solves a big problem

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

## 2) Saves a lot of time

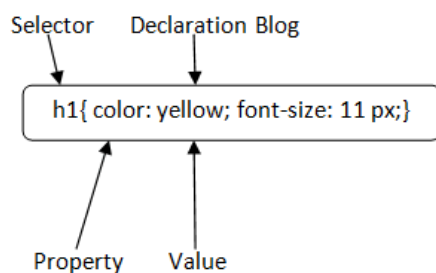
CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

## 3) Provide more attributes

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

# CSS Syntax

A CSS rule set contains a selector and a declaration block.



**Selector:** Selector indicates the HTML element you want to style. It could be any tag like `<h1>`, `<title>` etc.

**Declaration Block:** The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

1. `color: yellow;`
2. `font-size: 11 px;`

Each declaration contains a property name and value, separated by a colon.

**Property:** A Property is a type of attribute of HTML element. It could be color, border etc.

**Value:** Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

1. `Selector{Property1: value1; Property2: value2; .....;}`

## CSS Selector

**CSS selectors** are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

## 1) CSS Element Selector

The element selector selects the HTML element by name.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `p{`
6. `text-align: center;`
7. `color: blue;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<p>This style will be applied on every paragraph.</p>`
13. `<p id="para1">Me too!</p>`
14. `<p>And me!</p>`
15. `</body>`
16. `</html>`

Output:

This style will be applied on every paragraph.

Me too!

And me!

---

## 2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

Let's take an example with the id "para1".

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `#para1 {`
6. `text-align: center;`
7. `color: blue;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<p id="para1">Hello Javatpoint.com</p>`
13. `<p>This paragraph will not be affected.</p>`
14. `</body>`
15. `</html>`

Output:

Hello Javatpoint.com

This paragraph will not be affected.

---

## 3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

*Note: A class name should not be started with a number.*

Let's take an example with a class "center".

1. `<!DOCTYPE html>`
2. `<html>`

```
3. <head>
4. <style>
5. .center {
6.   text-align: center;
7.   color: blue;
8. }
9. </style>
10. </head>
11. <body>
12. <h1 class="center">This heading is blue and center-aligned.</h1>
13. <p class="center">This paragraph is blue and center-aligned.</p>
14. </body>
15. </html>
```

Output:

**This heading is blue and center-aligned.**

This paragraph is blue and center-aligned.

---

## CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

Let's see an example.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. p.center {
6.   text-align: center;
7.   color: blue;
8. }
9. </style>
10. </head>
11. <body>
12. <h1 class="center">This heading is not affected</h1>
13. <p class="center">This paragraph is blue and center-aligned.</p>
14. </body>
```

15. `</html>`

Output:

**This heading is not affected**

This paragraph is blue and center-aligned.

---

## 4) CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `* {`
6. `color: green;`
7. `font-size: 20px;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<h2>`This is heading`</h2>`
13. `<p>`This style will be applied on every paragraph.`</p>`
14. `<p id="para1">`Me too!`</p>`
15. `<p>`And me!`</p>`
16. `</body>`
17. `</html>`

Output:

**This is heading**

This style will be applied on every paragraph.

Me too!

And me!

---

## 5) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

Let's see the CSS code without group selector.

```
1. h1 {  
2.   text-align: center;  
3.   color: blue;  
4. }  
5. h2 {  
6.   text-align: center;  
7.   color: blue;  
8. }  
9. p {  
10.  text-align: center;  
11.  color: blue;  
12. }
```

As you can see, you need to define CSS properties for all the elements. It can be grouped in following ways:

```
1. h1,h2,p {  
2.   text-align: center;  
3.   color: blue;  
4. }
```

Let's see the full example of CSS group selector.

```
1. <!DOCTYPE html>  
2. <html>  
3. <head>  
4. <style>  
5. h1, h2, p {  
6.   text-align: center;  
7.   color: blue;  
8. }  
9. </style>  
10. </head>
```

11. `<body>`
12. `<h1>Hello Javatpoint.com</h1>`
13. `<h2>Hello Javatpoint.com (In smaller font)</h2>`
14. `<p>This is a paragraph.</p>`
15. `</body>`
16. `</html>`

Output:

**Hello Javatpoint.com**

**Hello Javatpoint.com (In smaller font)**

## How to add CSS

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

1. Inline CSS
2. Internal CSS
3. External CSS

---

### 1) Inline CSS

Inline CSS is used to apply CSS on a single line or element.

For example:

Competitive questions on Structures in Hindi

Keep Watching

1. `<p style="color:blue">Hello CSS</p>`

---

### 2) Internal CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

For example:

1. `<style>`
2. `p{color:blue}`
3. `</style>`



---

## 3) External CSS

External CSS is used to apply CSS on multiple pages or all pages. Here, we write all the CSS code in a css file. Its extension must be .css for example style.css.

For example:

1. `p{color:blue}`

You need to link this style.css file to your html pages like this:

1. `<link rel="stylesheet" type="text/css" href="style.css">`

The link tag must be used inside head section of html.

## Inline CSS

We can apply CSS in a single element by inline CSS technique.

The inline CSS is also a method to insert style sheets in HTML document. This method mitigates some advantages of style sheets so it is advised to use this method sparingly.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

Syntax:

1. `<htmltag style="cssproperty1:value; cssproperty2:value;"> </htmltag>`

Example:

1. `<h2 style="color:red;margin-left:40px;">Inline CSS is applied on this heading.</h2>`
2. `<p>This paragraph is not affected.</p>`

Output:

**Inline CSS is applied on this heading.**

This paragraph is not affected.

---

## Disadvantages of Inline CSS

- You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
- These styles cannot be reused anywhere else.
- These styles are tough to be edited because they are not stored at a single place.
- It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
- Inline CSS does not provide browser cache advantages.

## Internal CSS

The internal style sheet is used to add a unique style for a single document. It is defined in <head> section of the HTML page inside the <style> tag.

Example:

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6.     background-color: linen;
7. }
8. h1 {
9.     color: red;
10.    margin-left: 80px;
11. }
12. </style>
13. </head>
14. <body>
15. <h1>The internal style sheet is applied on this heading.</h1>
16. <p>This paragraph will not be affected.</p>
17. </body>
18. </html>

## External CSS

The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

Example:

1. `<head>`
2. `<link rel="stylesheet" type="text/css" href="mystyle.css">`
3. `</head>`

The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

Let's take an example of a style sheet file named "mystyle.css".

*File: mystyle.css*

1. `body {`
2. `background-color: lightblue;`
3. `}`
4. `h1 {`
5. `color: navy;`
6. `margin-left: 20px;`
7. `}`

Note: You should not use a space between the property value and the unit. For example: It should be `margin-left:20px` not `margin-left:20 px`.

## CSS Comments

CSS comments are generally written to explain your code. It is very helpful for the users who reads your code so that they can easily understand the code.

Comments are ignored by browsers.

Comments are single or multiple lines statement and written within `/*.....*/` .

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `p {`
6. `color: blue;`
7. `/* This is a single-line comment */`
8. `text-align: center;`
9. `}`
10. `/* This is`

11. a multi-line
12. comment \*/
13. `</style>`
14. `</head>`
15. `<body>`
16. `<p>`Hello Javatpoint.com`</p>`
17. `<p>`This statement is styled with CSS.`</p>`
18. `<p>`CSS comments are ignored by the browsers and not shown in the output.`</p>`
19. `</body>`
20. `</html>`

Output:

Hello Javatpoint.com

This statement is styled with CSS.

CSS comments are ignored by the browsers and not shown in the output.

## CSS Background

CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

---

### 1) CSS background-color

The background-color property is used to specify the background color of the element.

You can set the background color like this:

1. `<!DOCTYPE html>`
2. `<html>`

```
3. <head>
4. <style>
5. h2,p{
6.     background-color: #b0d4de;
7. }
8. </style>
9. </head>
10. <body>
11. <h2>My first CSS page.</h2>
12. <p>Hello Javatpoint. This is an example of CSS background-color.</p>
13. </body>
14. </html>
```

Output:

## My first CSS page.

Hello Javatpoint. This is an example of CSS background-color.

---

## 2) CSS background-image

The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6.     background-image: url("paper1.gif");
7.     margin-left:100px;
8. }
9. </style>
10. </head>
11. <body>
12. <h1>Hello Javatpoint.com</h1>
13. </body>
14. </html>
```

Note: The background image should be chosen according to text color. The bad combination of text and background image may be a cause of poor designed and not readable webpage.

---

### 3) CSS background-repeat

By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

The background looks better if the image repeated horizontally only.

#### **background-repeat: repeat-x;**

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6.     background-image: url("gradient_bg.png");
7.     background-repeat: repeat-x;
8. }
9. </style>
10. </head>
11. <body>
12. <h1>Hello Javatpoint.com</h1>
13. </body>
14. </html>
```

#### **background-repeat: repeat-y;**

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6.     background-image: url("gradient_bg.png");
7.     background-repeat: repeat-y;
8. }
9. </style>
10. </head>
11. <body>
```

12. `<h1>Hello Javatpoint.com</h1>`
  13. `</body>`
  14. `</html>`
- 

## 4) CSS background-attachment

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser. Let's take an example with fixed background image.

1. `background: white url('bbb.gif');`
2. `background-repeat: no-repeat;`
3. `background-attachment: fixed;`

## 5) CSS background-position

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

You can set the following positions:

1. center
  2. top
  3. bottom
  4. left
  5. right
1. `background: white url('good-morning.jpg');`
  2. `background-repeat: no-repeat;`
  3. `background-attachment: fixed;`
  4. `background-position: center;`

## CSS Border

The CSS border is a shorthand property used to set the border on an element.

The [CSS](#) border properties are use to specify the style, color and size of the border of an element. The CSS border properties are given below

- border-style
- border-color

- border-width
- border-radius

## 1) CSS border-style

The Border style property is used to specify the border type which you want to display on the web page.

There are some border style values which are used with border-style property to define a border.

Value	Description
none	It doesn't define any border.
dotted	It is used to define a dotted border.
dashed	It is used to define a dashed border.
solid	It is used to define a solid border.
double	It defines two borders with the same border-width value.
groove	It defines a 3d grooved border. effect is generated according to border-color value.
ridge	It defines a 3d ridged border. effect is generated according to border-color value.
inset	It defines a 3d inset border. effect is generated according to border-color value.
outset	It defines a 3d outset border. effect is generated according to border-color value.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. p.none {border-style: none;}
6. p.dotted {border-style: dotted;}
7. p.dashed {border-style: dashed;}
8. p.solid {border-style: solid;}
9. p.double {border-style: double;}



```
10. p.groove {border-style: groove;}
11. p.ridge {border-style: ridge;}
12. p.inset {border-style: inset;}
13. p.outset {border-style: outset;}
14. p.hidden {border-style: hidden;}
15. </style>
16. </head>
17. <body>
18. <p class="none">No border.</p>
19. <p class="dotted">A dotted border.</p>
20. <p class="dashed">A dashed border.</p>
21. <p class="solid">A solid border.</p>
22. <p class="double">A double border.</p>
23. <p class="groove">A groove border.</p>
24. <p class="ridge">A ridge border.</p>
25. <p class="inset">An inset border.</p>
26. <p class="outset">An outset border.</p>
27. <p class="hidden">A hidden border.</p>
28. </body>
29. </html>
```

Output:

No border.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

A hidden border.

## 2) CSS border-width

The border-width property is used to set the border's width. It is set in pixels. You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border.

*Note: The border-width property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.*

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5.   p.one {
6.     border-style: solid;
7.     border-width: 5px;
8.   }
9.   p.two {
10.    border-style: solid;
11.    border-width: medium;
12.  }
13. p.three {
14.   border-style: solid;
15.   border-width: 1px;
16. }
17. </style>
18. </head>
19. <body>
20. <p class="one">Write your text here.</p>
21. <p class="two">Write your text here.</p>
22. <p class="three">Write your text here.</p>
23. </body>
24. </html>
```

## 3) CSS border-color

There are three methods to set the color of the border.

- Name: It specifies the color name. For example: "red".
- RGB: It specifies the RGB value of the color. For example: "rgb(255,0,0)".
- Hex: It specifies the hex value of the color. For example: "#ff0000".

There is also a border color named "transparent". If the border color is not set it is inherited from the color property of the element.

*Note: The border-color property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.*

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. p.one {
6.   border-style: solid;
7.   border-color: red;
8. }
9. p.two {
10.   border-style: solid;
11.   border-color: #98bf21;
12. }
13. </style>
14. </head>
15. <body>
16. <p class="one">This is a solid red border</p>
17. <p class="two">This is a solid green border</p>
18. </body>
19. </html>

## CSS Background

CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

---

### 1) CSS background-color

The background-color property is used to specify the background color of the element.

You can set the background color like this:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `h2,p{`
6. `background-color: #b0d4de;`
7. `}`
8. `</style>`
9. `</head>`
10. `<body>`
11. `<h2>My first CSS page.</h2>`
12. `<p>Hello Javatpoint. This is an example of CSS background-color.</p>`
13. `</body>`
14. `</html>`

Output:

## My first CSS page.

Hello Javatpoint. This is an example of CSS background-color.

---

## 2) CSS background-image

The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `body {`
6. `background-image: url("paper1.gif");`
7. `margin-left:100px;`
8. `}`
9. `</style>`
10. `</head>`
11. `<body>`
12. `<h1>Hello Javatpoint.com</h1>`

13. `</body>`

14. `</html>`

Note: The background image should be chosen according to text color. The bad combination of text and background image may be a cause of poor designed and not readable webpage.

---

### 3) CSS background-repeat

By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

The background looks better if the image repeated horizontally only.

**background-repeat: repeat-x;**

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6.     background-image: url("gradient_bg.png");
7.     background-repeat: repeat-x;
8. }
9. </style>
10. </head>
11. <body>
12. <h1>Hello Javatpoint.com</h1>
13. </body>
14. </html>
```

**background-repeat: repeat-y;**

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6.     background-image: url("gradient_bg.png");
7.     background-repeat: repeat-y;
8. }
```

9. `</style>`
  10. `</head>`
  11. `<body>`
  12. `<h1>Hello Javatpoint.com</h1>`
  13. `</body>`
  14. `</html>`
- 

## 4) CSS background-attachment

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser. Let's take an example with fixed background image.

1. `background: white url('bbb.gif');`
2. `background-repeat: no-repeat;`
3. `background-attachment: fixed;`

## 5) CSS background-position

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

You can set the following positions:

1. center
  2. top
  3. bottom
  4. left
  5. right
1. `background: white url('good-morning.jpg');`
  2. `background-repeat: no-repeat;`
  3. `background-attachment: fixed;`
  4. `background-position: center;`

## CSS Border

The CSS border is a shorthand property used to set the border on an element.

The [CSS](#) border properties are used to specify the style, color and size of the border of an element. The CSS border properties are given below

- border-style
- border-color
- border-width
- border-radius

## 1) CSS border-style

The Border style property is used to specify the border type which you want to display on the web page.

There are some border style values which are used with border-style property to define a border.

Value	Description
none	It doesn't define any border.
dotted	It is used to define a dotted border.
dashed	It is used to define a dashed border.
solid	It is used to define a solid border.
double	It defines two borders with the same border-width value.
groove	It defines a 3d grooved border. effect is generated according to border-color value.
ridge	It defines a 3d ridged border. effect is generated according to border-color value.
inset	It defines a 3d inset border. effect is generated according to border-color value.
outset	It defines a 3d outset border. effect is generated according to border-color value.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<style>`
5. `p.none {border-style: none;}`
6. `p.dotted {border-style: dotted;}`

```
7. p.dashed {border-style: dashed;}
8. p.solid {border-style: solid;}
9. p.double {border-style: double;}
10. p.groove {border-style: groove;}
11. p.ridge {border-style: ridge;}
12. p.inset {border-style: inset;}
13. p.outset {border-style: outset;}
14. p.hidden {border-style: hidden;}
15. </style>
16. </head>
17. <body>
18. <p class="none">No border.</p>
19. <p class="dotted">A dotted border.</p>
20. <p class="dashed">A dashed border.</p>
21. <p class="solid">A solid border.</p>
22. <p class="double">A double border.</p>
23. <p class="groove">A groove border.</p>
24. <p class="ridge">A ridge border.</p>
25. <p class="inset">An inset border.</p>
26. <p class="outset">An outset border.</p>
27. <p class="hidden">A hidden border.</p>
28. </body>
29. </html>
```

Output:

No border.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.



An outset border.

A hidden border.

## 2) CSS border-width

The border-width property is used to set the border's width. It is set in pixels. You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border.

*Note: The border-width property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.*

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5.   p.one {
6.     border-style: solid;
7.     border-width: 5px;
8.   }
9.   p.two {
10.    border-style: solid;
11.    border-width: medium;
12.  }
13.  p.three {
14.    border-style: solid;
15.    border-width: 1px;
16.  }
17. </style>
18. </head>
19. <body>
20. <p class="one">Write your text here.</p>
21. <p class="two">Write your text here.</p>
22. <p class="three">Write your text here.</p>
23. </body>
24. </html>
```

## 3) CSS border-color

There are three methods to set the color of the border.

- Name: It specifies the color name. For example: "red".

- RGB: It specifies the RGB value of the color. For example: "rgb(255,0,0)".
- Hex: It specifies the hex value of the color. For example: "#ff0000".

There is also a border color named "transparent". If the border color is not set it is inherited from the color property of the element.

*Note: The border-color property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.*

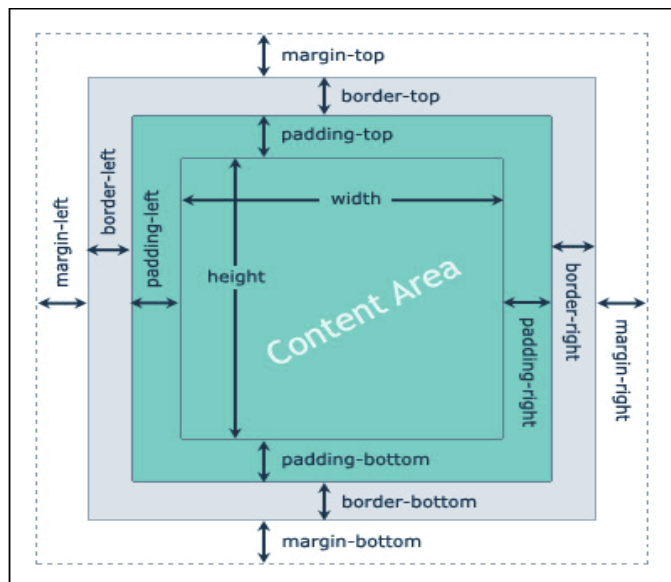
```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5.   p.one {
6.     border-style: solid;
7.     border-color: red;
8.   }
9.   p.two {
10.    border-style: solid;
11.    border-color: #98bf21;
12.  }
13. </style>
14. </head>
15. <body>
16. <p class="one">This is a solid red border</p>
17. <p class="two">This is a solid green border</p>
18. </body>
19. </html>
```

## CSS Box Model

The components that can be depicted on the web page consist of one or more than one rectangular box.

A CSS box model is a compartment that includes numerous assets, such as edge, border, padding and material. It is used to develop the design and structure of a web page. It can be used as a set of tools to personalize the layout of different components. According to the CSS box model, the web browser supplies each element as a square prism.

The following diagram illustrates how the CSS properties of width, height, padding, border and margin dictate that how much space an attribute will occupy on a web page.



The [CSS](#) box model contains the different properties in CSS. These are listed below.

- **Border**
- **Margin**
- **Padding**
- **Content**

Now, we are going to determine the properties one by one in detail.

### **Border Field**

It is a region between the padding-box and the margin. Its proportions are determined by the width and height of the boundary.

### **Margin Field**

This segment consists of the area between the boundary and the edge of the border.

The proportion of the margin region is equal to the margin-box width and height. It is better to separate the product from its neighbor nodes.

### **Padding Field**

This field requires the padding of the component. In essence, this area is the space around the subject area and inside the border-box. The height and the width of the padding box decide its proportions.

### **Content Field**

Material such as text, photographs, or other digital media is included in this area.

It is constrained by the information edge, and its proportions are dictated by the width and height of the content enclosure.

## Elements of the width and height

Typically, when you assign the width and height of an attribute using the CSS width and height assets, it means you just positioned the height and width of the subject areas of that component. The additional height and width of the unit box is based on a range of influences.

The specific area that an element box may occupy on a web page is measured as follows-

Size of the box	Properties of CSS
Height	height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom
Width	width + padding-left + padding-right + border-left + border-right + margin-left + margin-right

### Example 1

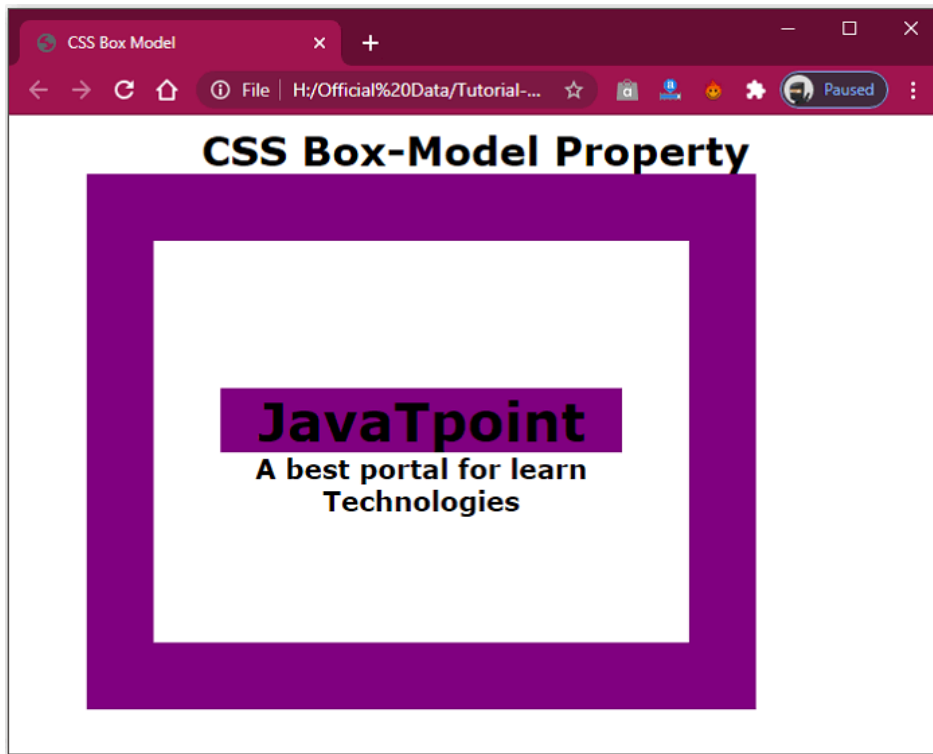
Here, to explain the CSS box model, we have an instance.

```
1. <!DOCTYPE html>
2. <head>
3. <title>CSS Box Model</title>
4. <style>
5.     .main
6. {
7.     font-size:30px;
8.     font-weight:bold;
9.     Text-align:center;
10. }
11. .gfg
12. {
13.     margin-left:50px;
14.     border:50px solid Purple;
15.     width:300px;
```

```
16.         height:200px;
17.         text-align:center;
18.         padding:50px;
19.     }
20.     .gfg1
21. {
22.     font-size:40px;
23.     font-weight:bold;
24.     color:black;
25.     margin-top:60px;
26.     background-color:purple;
27. }
28.     .gfg2
29. {
30.     font-size:20px;
31.     font-weight:bold;
32.     background-color:white;
33. }
34. </style>
35. </head>
36. <body>
37. <div class = "main">CSS Box-Model Property</div>
38.     <div class = "gfg">
39.     <div class = "gfg1">JavaTpoint</div>
40.     <div class = "gfg2">A best portal for learn Technologies</div>
41. </div>
42. </body>
43. </html>
```

## Output

After the compilation of the above code, you get the following output.



## Example 2

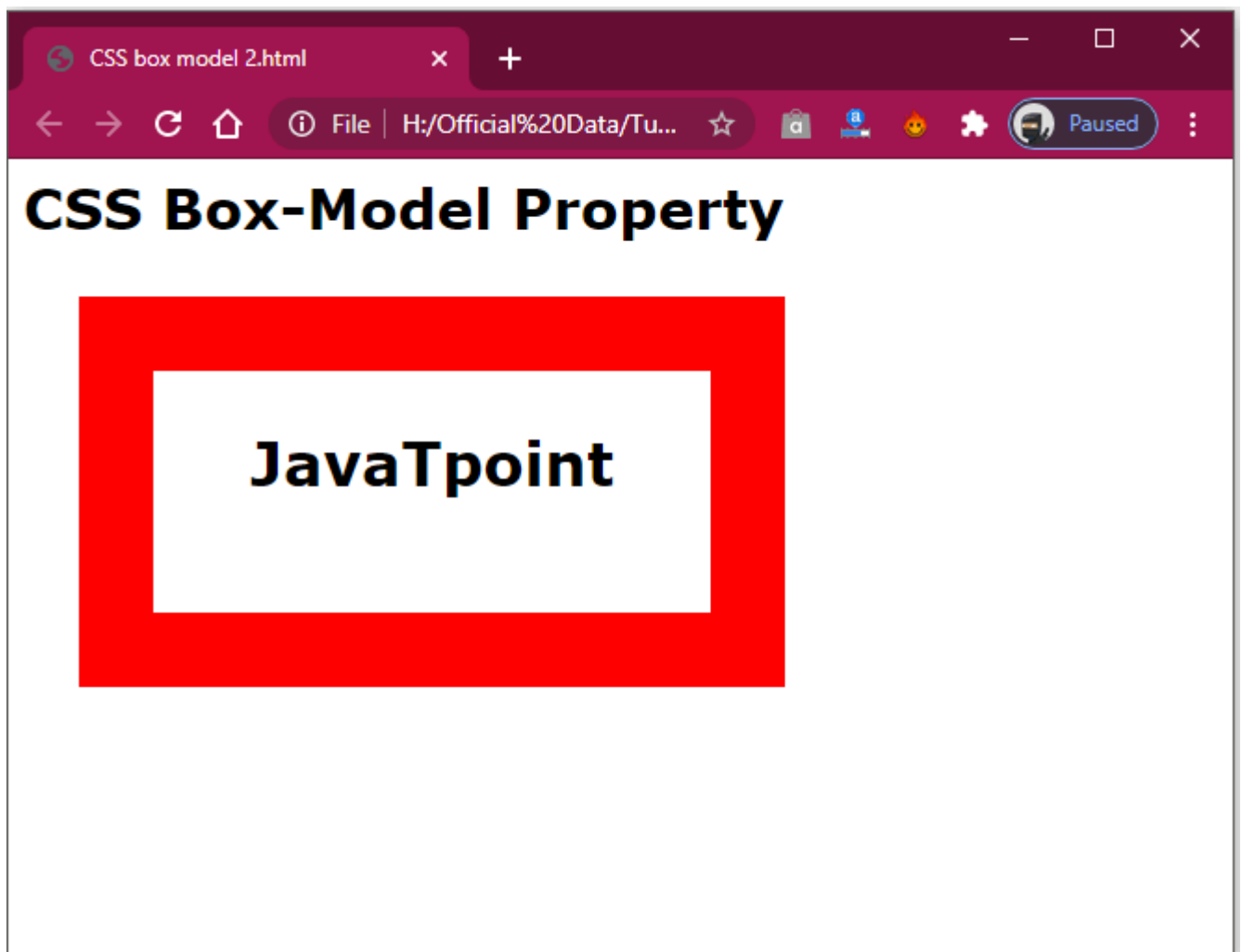
Here, we also have an illustration to describe the CSS box model.

```
1. <!DOCTYPE html>
2. <head>
3. <style>
4.     .main
5.     {
6.         font-size:30px;
7.         font-weight:bold;
8.         text-align:left;
9.     }
10. #box
11. {
12.     padding-top:30px;
13.     width: 300px;
14.     height: 100px;
15.     border: 40px solid red;
16.     margin: 30px;
17.     text-align:center;
18.     font-size:32px;
19.     font-weight:bold;
20. }
21. </style>
```

```
22. </head>
23. <body>
24.     <div class="main">CSS Box-Model Property</div>
25.     <div id="box">JavaTpoint</div>
26. </body>
27. </html>
```

## Output

After the execution of the code, you get the following output:



**Important Point:** In the CSS box model, the subject area of an entity box is the region where the content, such as image, text, video, etc., initially appeared. It may also retain boxes of decedent elements.

## What is JavaScript

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to

the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

## Features of JavaScript

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

## History of JavaScript

In 1993, **Mosaic**, the first popular web browser, came into existence. In the **year 1994**, **Netscape** was founded by **Marc Andreessen**. He realized that the web needed to become more dynamic. Thus, a 'glue language' was believed to be provided to HTML to make web designing easy for designers and part-time programmers. Consequently, in 1995, the company recruited **Brendan Eich** intending to implement and embed Scheme programming language to the browser. But, before Brendan could start, the company merged with **Sun Microsystems** for adding Java into its Navigator so that it could compete with Microsoft over the web technologies and platforms. Now, two languages were there: Java and the scripting language. Further, Netscape decided to give a similar name to the scripting language as Java's. It led to 'Javascrpt'. Finally, in May 1995, Marc Andreessen coined the first code of Javascript named '**Mocha**'. Later, the marketing team replaced the name with '**LiveScript**'. But, due to trademark reasons and certain other reasons, in December



1995, the language was finally renamed to 'JavaScript'. From then, JavaScript came into existence.

## Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- Displaying clocks etc.

## JavaScript Example

1. `<script>`
2. `document.write("Hello JavaScript by JavaScript");`
3. `</script>`

## JavaScript Example

1. [JavaScript Example](#)
2. [Within body tag](#)
3. [Within head tag](#)

Javascript example is easy to code. JavaScript provides 3 places to put the JavaScript code: within body tag, within head tag and external JavaScript file.

Let's create the first JavaScript example.

1. `<script type="text/javascript">`
2. `document.write("JavaScript is a simple language for javatpoint learners");`
3. `</script>`

The **script** tag specifies that we are using JavaScript.

The **text/javascript** is the content type that provides information to the browser about the data.

The **document.write()** function is used to display dynamic content through JavaScript. We will learn about document object in detail later.

---

## 3 Places to put JavaScript code

1. Between the body tag of html
  2. Between the head tag of html
  3. In .js file (external javascript)
- 

### 1) JavaScript Example : code between the body tag

In the above example, we have displayed the dynamic content using JavaScript. Let's see the simple example of JavaScript that displays alert dialog box.

1. `<script type="text/javascript">`
2. `alert("Hello Javatpoint");`
3. `</script>`

### 2) JavaScript Example : code between the head tag

Let's see the same example of displaying alert dialog box of JavaScript that is contained inside the head tag.

In this example, we are creating a function msg(). To create function in JavaScript, you need to write function with function\_name as given below.

To call function, you need to work on event. Here we are using onclick event to call msg() function.

1. `<html>`
2. `<head>`
3. `<script type="text/javascript">`
4. `function msg(){`
5. `alert("Hello Javatpoint");`
6. `}`
7. `</script>`

8. `</head>`
9. `<body>`
10. `<p>Welcome to JavaScript</p>`
11. `<form>`
12. `<input type="button" value="click" onclick="msg()"/>`
13. `</form>`
14. `</body>`
15. `</html>`

## External JavaScript file

We can create external JavaScript file and embed it in many html page.

It provides **code re usability** because single JavaScript file can be used in several html pages.

An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.

Let's create an external [JavaScript](#) file that prints Hello Javatpoint in a alert dialog box.

### message.js

1. `function msg(){`
2. `alert("Hello Javatpoint");`
3. `}`

Let's include the JavaScript file into [html](#) page. It calls the [JavaScript function](#) on button click.

### index.html

1. `<html>`
2. `<head>`
3. `<script type="text/javascript" src="message.js"></script>`
4. `</head>`
5. `<body>`
6. `<p>Welcome to JavaScript</p>`
7. `<form>`
8. `<input type="button" value="click" onclick="msg()"/>`
9. `</form>`
10. `</body>`

## 11. `</html>`

### Advantages of External JavaScript

There will be following benefits if a user creates an external javascript:

1. It helps in the reusability of code in more than one HTML file.
2. It allows easy code readability.
3. It is time-efficient as web browsers cache the external js files, which further reduces the page loading time.
4. It enables both web designers and coders to work with html and js files parallelly and separately, i.e., without facing any code conflictions.
5. The length of the code reduces as only we need to specify the location of the js file.

### Disadvantages of External JavaScript

There are the following disadvantages of external files:

1. The stealer may download the coder's code using the url of the js file.
2. If two js files are dependent on one another, then a failure in one file may affect the execution of the other dependent file.
3. The web browser needs to make an additional http request to get the js code.
4. A tiny to a large change in the js code may cause unexpected results in all its dependent files.
5. We need to check each file that depends on the commonly created external javascript file.
6. If it is a few lines of code, then better to implement the internal javascript code.

## JavaScript Comment

1. [JavaScript comments](#)
2. [Advantage of javascript comments](#)
3. [Single-line and Multi-line comments](#)

The **JavaScript comments** are meaningful way to deliver message. It is used to add information about the code, warnings or suggestions so that end user can easily interpret the code.

The JavaScript comment is ignored by the JavaScript engine i.e. embedded in the browser.

## Advantages of JavaScript comments

There are mainly two advantages of JavaScript comments.

1. **To make code easy to understand** It can be used to elaborate the code so that end user can easily understand the code.
  2. **To avoid the unnecessary code** It can also be used to avoid the code being executed. Sometimes, we add the code to perform some action. But after sometime, there may be need to disable the code. In such case, it is better to use comments.
- 

## Types of JavaScript Comments

There are two types of comments in JavaScript.

1. Single-line Comment
  2. Multi-line Comment
- 

## JavaScript Single line Comment

It is represented by double forward slashes (//). It can be used before and after the statement.

Let's see the example of single-line comment i.e. added before the statement.

1. `<script>`
2. `// It is single line comment`
3. `document.write("hello javascript");`
4. `</script>`

Let's see the example of single-line comment i.e. added after the statement.

1. `<script>`
2. `var a=10;`
3. `var b=20;`
4. `var c=a+b;//It adds values of a and b variable`
5. `document.write(c);//prints sum of 10 and 20`
6. `</script>`

# JavaScript Multi line Comment

It can be used to add single as well as multi line comments. So, it is more convenient.

It is represented by forward slash with asterisk then asterisk with forward slash. For example:

1. `/* your code here */`

It can be used before, after and middle of the statement.

1. `<script>`
2. `/* It is multi line comment.`
3. `It will not be displayed */`
4. `document.write("example of javascript multiline comment");`
5. `</script>`

# JavaScript Variable

1. [JavaScript variable](#)
2. [JavaScript Local variable](#)
3. [JavaScript Global variable](#)

A **JavaScript variable** is simply a name of storage location. There are two types of variables in JavaScript : local variable and global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

1. Name must start with a letter (a to z or A to Z), underscore( \_ ), or dollar( \$ ) sign.
2. After first letter we can use digits (0 to 9), for example value1.
3. JavaScript variables are case sensitive, for example x and X are different variables.

---

## Correct JavaScript variables

1. `var x = 10;`
2. `var _value="sonoo";`

---

## Incorrect JavaScript variables

1. `var 123=30;`
  2. `var *aa=320;`
-

# Example of JavaScript variable

Let's see a simple example of JavaScript variable.

1. `<script>`
2. `var x = 10;`
3. `var y = 20;`
4. `var z=x+y;`
5. `document.write(z);`
6. `</script>`

*Output of the above example*

30

---

## JavaScript local variable

A JavaScript local variable is declared inside block or function. It is accessible within the function or block only. For example:

1. `<script>`
2. `function abc(){`
3. `var x=10;//local variable`
4. `}`
5. `</script>`

Or,

1. `<script>`
2. `If(10<13){`
3. `var y=20;//JavaScript local variable`
4. `}`
5. `</script>`

---

## JavaScript global variable

A **JavaScript global variable** is accessible from any function. A variable i.e. declared outside the function or declared with window object is known as global variable. For example:

1. `<script>`

```
2. var data=200;//global variable
3. function a(){
4. document.writeln(data);
5. }
6. function b(){
7. document.writeln(data);
8. }
9. a();//calling JavaScript function
10. b();
11. </script>
```

## JavaScript Global Variable

A **JavaScript global variable** is declared outside the function or declared with window object. It can be accessed from any function.

Let's see the simple example of global variable in JavaScript.

```
1. <script>
2. var value=50;//global variable
3. function a(){
4. alert(value);
5. }
6. function b(){
7. alert(value);
8. }
9. </script>
```

### *Declaring JavaScript global variable within function*

To declare JavaScript global variables inside function, you need to use **window object**. For example:

```
1. window.value=90;
```

Now it can be declared inside any function and can be accessed from any function. For example:

```
1. function m(){
2. window.value=100;//declaring global variable by window object
3. }
4. function n(){
```



5. `alert(window.value);`//accessing global variable from other function
6. `}`

## Internals of global variable in JavaScript

When you declare a variable outside the function, it is added in the window object internally. You can access it through window object also. For example:

1. `var value=50;`
2. `function a(){`
3. `alert(window.value);`//accessing global variable
4. `}`

## Javascript Data Types

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

JavaScript is a **dynamic type language**, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use **var** here to specify the data type. It can hold any type of values such as numbers, strings etc. For example:

1. `var a=40;`//holding number
2. `var b="Rahul";`//holding string

## JavaScript primitive data types

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	Description
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

# JavaScript non-primitive data types

The non-primitive data types are as follows:

Data Type	Description
Object	represents instance through which we can access members
Array	represents group of similar values
RegExp	represents regular expression

## JavaScript Operators

JavaScript operators are symbols that are used to perform operations on operands. For example:

1. `var sum=10+20;`

Here, + is the arithmetic operator and = is the assignment operator.

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

---

## JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
----------	-------------	---------

+	Addition	10+20 = 30
-	Subtraction	20-10 = 10
*	Multiplication	10*20 = 200
/	Division	20/10 = 2
%	Modulus (Remainder)	20%10 = 0
++	Increment	var a=10; a++; Now a = 11
--	Decrement	var a=10; a--; Now a = 9

## JavaScript Comparison Operators

The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

Operator	Description	Example
==	Is equal to	10==20 = false
===	Identical (equal and of same type)	10===20 = false
!=	Not equal to	10!=20 = true
!==	Not Identical	20!==20 = false
>	Greater than	20>10 = true
>=	Greater than or equal to	20>=10 = true
<	Less than	20<10 = false
<=	Less than or equal to	20<=10 = false

## JavaScript Bitwise Operators

The bitwise operators perform bitwise operations on operands. The bitwise operators are as follows:

Operator	Description	Example
&	Bitwise AND	(10==20 & 20==33) = false
	Bitwise OR	(10==20   20==33) = false

<code>^</code>	Bitwise XOR	$(10 == 20 \wedge 20 == 33) = \text{false}$
<code>~</code>	Bitwise NOT	$(\sim 10) = -10$
<code>&lt;&lt;</code>	Bitwise Left Shift	$(10 < < 2) = 40$
<code>&gt;&gt;</code>	Bitwise Right Shift	$(10 > > 2) = 2$
<code>&gt;&gt;&gt;</code>	Bitwise Right Shift with Zero	$(10 > > > 2) = 2$

## JavaScript Logical Operators

The following operators are known as JavaScript logical operators.

Operator	Description	Example
<code>&amp;&amp;</code>	Logical AND	$(10 == 20 \&\& 20 == 33) = \text{false}$
<code>  </code>	Logical OR	$(10 == 20    20 == 33) = \text{false}$
<code>!</code>	Logical Not	$!(10 == 20) = \text{true}$

## JavaScript Assignment Operators

The following operators are known as JavaScript assignment operators.

Operator	Description	Example
<code>=</code>	Assign	$10 + 10 = 20$
<code>+=</code>	Add and assign	<code>var a=10; a+=20; Now a = 30</code>
<code>-=</code>	Subtract and assign	<code>var a=20; a-=10; Now a = 10</code>
<code>*=</code>	Multiply and assign	<code>var a=10; a*=20; Now a = 200</code>
<code>/=</code>	Divide and assign	<code>var a=10; a/=2; Now a = 5</code>
<code>%=</code>	Modulus and assign	<code>var a=10; a%=2; Now a = 0</code>

## JavaScript Special Operators

The following operators are known as JavaScript special operators.

Operator	Description
----------	-------------

(?:)	Conditional Operator returns value based on the condition. It is like if-else.
,	Comma Operator allows multiple expressions to be evaluated as single statement.
delete	Delete Operator deletes a property from the object.
in	In Operator checks if object has the given property
instanceof	checks if the object is an instance of given type
new	creates an instance (object)
typeof	checks the type of object.
void	it discards the expression's return value.
yield	checks what is returned in a generator by the generator's iterator.

## JavaScript If-else

The **JavaScript if-else statement** is used to *execute the code whether condition is true or false*. There are three forms of if statement in JavaScript.

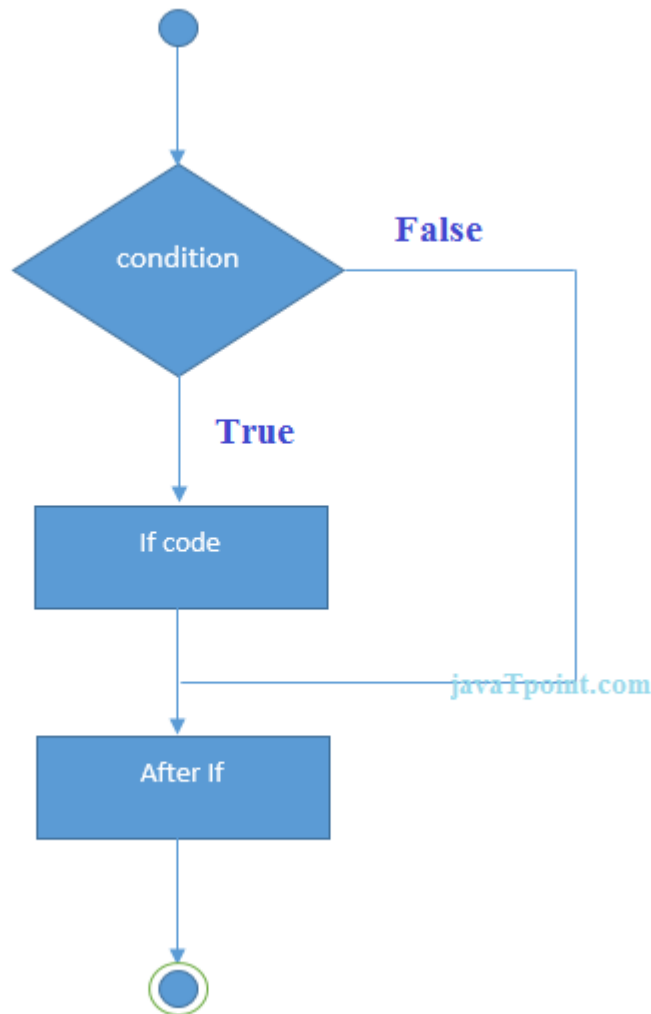
1. If Statement
2. If else statement
3. if else if statement

## JavaScript If statement

It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

1. if(expression){
2. //content to be evaluated
3. }

## Flowchart of JavaScript If statement



Let's see the simple example of if statement in javascript.

1. `<script>`
2. `var a=20;`
3. `if(a>10){`
4. `document.write("value of a is greater than 10");`
5. `}`
6. `</script>`

*Output of the above example*

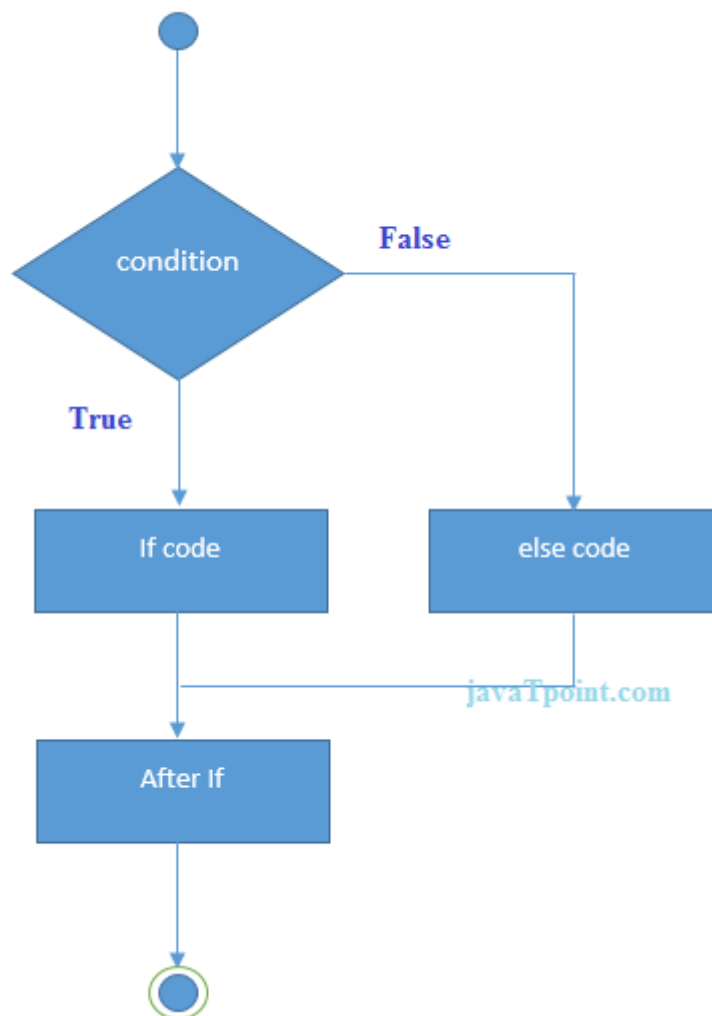
value of a is greater than 10

## JavaScript If...else Statement

It evaluates the content whether condition is true or false. The syntax of JavaScript if-else statement is given below.

1. `if(expression){`
2. `//content to be evaluated if condition is true`
3. `}`
4. `else{`
5. `//content to be evaluated if condition is false`
6. `}`

### Flowchart of JavaScript If...else statement



Let's see the example of if-else statement in JavaScript to find out the even or odd number.

1. `<script>`
2. `var a=20;`
3. `if(a%2==0){`
4. `document.write("a is even number");`
5. `}`
6. `else{`
7. `document.write("a is odd number");`

8. }
9. `</script>`

*Output of the above example*

```
a is even number
```

---

## JavaScript If...else if statement

It evaluates the content only if expression is true from several expressions. The signature of JavaScript if else if statement is given below.

1. if(expression1){
2. //content to be evaluated if expression1 is true
3. }
4. else if(expression2){
5. //content to be evaluated if expression2 is true
6. }
7. else if(expression3){
8. //content to be evaluated if expression3 is true
9. }
10. else{
11. //content to be evaluated if no expression is true
12. }

Let's see the simple example of if else if statement in javascript.

1. `<script>`
2. var `a=20`;
3. if(`a==10`){
4. document.write("a is equal to 10");
5. }
6. else if(`a==15`){
7. document.write("a is equal to 15");
8. }
9. else if(`a==20`){
10. document.write("a is equal to 20");
11. }
12. else{
13. document.write("a is not equal to 10, 15 or 20");
14. }
15. `</script>`



# JavaScript Switch

The **JavaScript switch statement** is used to *execute one code from multiple expressions*. It is just like else if statement that we have learned in previous page. But it is convenient than *if..else..if* because it can be used with numbers, characters etc.

The signature of JavaScript switch statement is given below.

1. switch(expression){
2. case value1:
3. code to be executed;
4. break;
5. case value2:
6. code to be executed;
7. break;
8. ....
- 9.
10. default:
11. code to be executed if above values are not matched;
12. }

Let's see the simple example of switch statement in javascript.

1. `<script>`
2. `var grade='B';`
3. `var result;`
4. `switch(grade){`
5. `case 'A':`
6. `result="A Grade";`
7. `break;`
8. `case 'B':`
9. `result="B Grade";`
10. `break;`
11. `case 'C':`
12. `result="C Grade";`
13. `break;`
14. `default:`
15. `result="No Grade";`
16. `}`
17. `document.write(result);`
18. `</script>`

### Output of the above example

B Grade

*The switch statement is fall-through i.e. all the cases will be evaluated if you don't use break statement.*

Let's understand the behaviour of switch statement in JavaScript.

Competitive questions on Structures in Hindi

Keep Watching

1. `<script>`
2. `var grade='B';`
3. `var result;`
4. `switch(grade){`
5. `case 'A':`
6. `result+=" A Grade";`
7. `case 'B':`
8. `result+=" B Grade";`
9. `case 'C':`
10. `result+=" C Grade";`
11. `default:`
12. `result+=" No Grade";`
13. `}`
14. `document.write(result);`
15. `</script>`

## JavaScript Loops

The **JavaScript loops** are used to *iterate the piece of code* using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop
4. for-in loop

---

### 1) JavaScript For loop

The **JavaScript for loop** *iterates the elements for the fixed number of times*. It should be used if number of iteration is known. The syntax of for loop is given below.

1. for (initialization; condition; increment)
2. {
3.     code to be executed
4. }

Let's see the simple example of for loop in javascript.

1. **<script>**
2. for (**i=1**; i<=5; i++)
3. {
4.     document.write(i + "**<br/>**")
5. }
6. **</script>**

Output:

```
1
2
3
4
5
```

## 2) JavaScript while loop

The **JavaScript while loop** *iterates the elements for the infinite number of times*. It should be used if number of iteration is not known. The syntax of while loop is given below.

1. while (condition)
2. {
3.     code to be executed
4. }

Let's see the simple example of while loop in javascript.

1. **<script>**
2. var **i=11**;
3. while (i<=15)
4. {
5.     document.write(i + "**<br/>**");

6. `i++;`
7. `}`
8. `</script>`

Output:

```
11
12
13
14
15
```

### 3) JavaScript do while loop

The **JavaScript do while loop** *iterates the elements for the infinite number of times* like while loop. But, code is *executed at least* once whether condition is true or false. The syntax of do while loop is given below.

1. `do{`
2.     code to be executed
3. `}while (condition);`

Let's see the simple example of do while loop in javascript.

1. `<script>`
2. `var i=21;`
3. `do{`
4. `document.write(i + "<br/>");`
5. `i++;`
6. `}while (i<=25);`
7. `</script>`

Output:

```
21
22
23
24
25
```

## JavaScript Functions

**JavaScript functions** are used to perform operations. We can call JavaScript function many times to reuse the code.

## Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. **Code reusability:** We can call a function several times so it save coding.
  2. **Less coding:** It makes our program compact. We don't need to write many lines of code each time to perform a common task.
- 

## JavaScript Function Syntax

The syntax of declaring function is given below.

1. `function functionName([arg1, arg2, ...argN]){`
2. `//code to be executed`
3. `}`

JavaScript Functions can have 0 or more arguments.

## JavaScript Function Example

Let's see the simple example of function in JavaScript that does not has arguments.

1. `<script>`
2. `function msg(){`
3. `alert("hello! this is message");`
4. `}`
5. `</script>`
6. `<input type="button" onclick="msg()" value="call function"/>`

*Output of the above example*

## JavaScript Function Arguments

We can call function by passing arguments. Let's see the example of function that has one argument.

1. `<script>`
2. `function getcube(number){`
3. `alert(number*number*number);`
4. `}`
5. `</script>`

6. `<form>`
7. `<input type="button" value="click" onclick="getcube(4)"/>`
8. `</form>`

*Output of the above example*

## Function with Return Value

We can call function that returns a value and use it in our program. Let's see the example of function that returns value.

1. `<script>`
2. `function getInfo(){`
3. `return "hello javatpoint! How r u?";`
4. `}`
5. `</script>`
6. `<script>`
7. `document.write(getInfo());`
8. `</script>`

*Output of the above example*

```
hello javatpoint! How r u?
```

---

## JavaScript Function Object

In JavaScript, the purpose of **Function constructor** is to create a new Function object. It executes the code globally. However, if we call the constructor directly, a function is created dynamically but in an unsecured way.

## Syntax

1. `new Function ([arg1[, arg2[, ....argn]],] functionBody)`

## Parameter

**arg1, arg2, .... , argn** - It represents the argument used by function.

**functionBody** - It represents the function definition.

## JavaScript Function Methods

Let's see function methods with description.

Method	Description
<a href="#">apply()</a>	It is used to call a function contains this value and a single array of arguments.
<a href="#">bind()</a>	It is used to create a new function.
<a href="#">call()</a>	It is used to call a function contains this value and an argument list.
<a href="#">toString()</a>	It returns the result in a form of a string.

## JavaScript Function Object Examples

### Example 1

Let's see an example to display the sum of given numbers.

1. `<script>`
2. `var add=new Function("num1","num2","return num1+num2");`
3. `document.writeln(add(2,5));`
4. `</script>`

**Output:**

```
7
```

### Example 2

Let's see an example to display the power of provided value.

1. `<script>`
2. `var pow=new Function("num1","num2","return Math.pow(num1,num2)");`
3. `document.writeln(pow(2,3));`
4. `</script>`

## JavaScript Objects

A javascript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.

JavaScript is an object-based language. Everything is an object in JavaScript.

JavaScript is template based not class based. Here, we don't create class to get the object. But, we direct create objects.

---

## Creating Objects in JavaScript

There are 3 ways to create objects.

1. By object literal
  2. By creating instance of Object directly (using new keyword)
  3. By using an object constructor (using new keyword)
- 

### 1) JavaScript Object by object literal

The syntax of creating object using object literal is given below:

1. `object={property1:value1,property2:value2.....propertyN:valueN}`

As you can see, property and value is separated by : (colon).

Let's see the simple example of creating object in JavaScript.

1. `<script>`
2. `emp={id:102,name:"Shyam Kumar",salary:40000}`
3. `document.write(emp.id+" "+emp.name+" "+emp.salary);`
4. `</script>`

*Output of the above example*

```
102 Shyam Kumar 40000
```

---

### 2) By creating instance of Object

The syntax of creating object directly is given below:

1. `var objectname=new Object();`

Here, **new keyword** is used to create object.

Let's see the example of creating object directly.

1. `<script>`



2. `var emp=new Object();`
3. `emp.id=101;`
4. `emp.name="Ravi Malik";`
5. `emp.salary=50000;`
6. `document.write(emp.id+" "+emp.name+" "+emp.salary);`
7. `</script>`

*Output of the above example*

```
101 Ravi 50000
```

---

### 3) By using an Object constructor

Here, you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword.

The **this keyword** refers to the current object.

The example of creating object by object constructor is given below.

1. `<script>`
2. `function emp(id,name,salary){`
3. `this.id=id;`
4. `this.name=name;`
5. `this.salary=salary;`
6. `}`
7. `e=new emp(103,"Vimal Jaiswal",30000);`
- 8.
9. `document.write(e.id+" "+e.name+" "+e.salary);`
10. `</script>`

*Output of the above example*

```
103 Vimal Jaiswal 30000
```

---

## Defining method in JavaScript Object

We can define method in JavaScript object. But before defining method, we need to add property in the function with same name as method.

The example of defining method in object is given below.

```

1. <script>
2. function emp(id,name,salary){
3.   this.id=id;
4.   this.name=name;
5.   this.salary=salary;
6.
7.   this.changeSalary=changeSalary;
8.   function changeSalary(otherSalary){
9.     this.salary=otherSalary;
10.  }
11. }
12. e=new emp(103,"Sonoo Jaiswal",30000);
13. document.write(e.id+" "+e.name+" "+e.salary);
14. e.changeSalary(45000);
15. document.write("<br>" +e.id+" "+e.name+" "+e.salary);
16. </script>

```

*Output of the above example*

```

103                Sonoo                Jaiswal                30000
103 Sonoo Jaiswal 45000

```

## JavaScript Object Methods

The various methods of Object are as follows:

S.No	Methods	Description
1	<a href="#">Object.assign()</a>	This method is used to copy enumerable and own properties from a source object to a target object
2	<a href="#">Object.create()</a>	This method is used to create a new object with the specified prototype object and properties.
3	<a href="#">Object.defineProperty()</a>	This method is used to describe some behavioral attributes of the property.
4	<a href="#">Object.defineProperties()</a>	This method is used to create or configure multiple object properties.

	<a href="#"><u>rties()</u></a>	
5	<a href="#"><u>Object.entries()</u></a>	This method returns an array with arrays of the key, value pairs.
6	<a href="#"><u>Object.freeze()</u></a>	This method prevents existing properties from being removed.
7	<a href="#"><u>Object.getOwnPro pertyDescriptor()</u></a>	This method returns a property descriptor for the specified property of the specified object.
8	<a href="#"><u>Object.getOwnPro pertyDescriptors()</u></a>	This method returns all own property descriptors of a given object.
9	<a href="#"><u>Object.getOwnPro pertyNames()</u></a>	This method returns an array of all properties (enumerable or not) found.
10	<a href="#"><u>Object.getOwnPro pertySymbols()</u></a>	This method returns an array of all own symbol key properties.
11	<a href="#"><u>Object.getPrototyp eOf()</u></a>	This method returns the prototype of the specified object.
12	<a href="#"><u>Object.is()</u></a>	This method determines whether two values are the same value.
13	<a href="#"><u>Object.isExtensible( )</u></a>	This method determines if an object is extensible
14	<a href="#"><u>Object.isFrozen()</u></a>	This method determines if an object was frozen.
15	<a href="#"><u>Object.isSealed()</u></a>	This method determines if an object is sealed.
16	<a href="#"><u>Object.keys()</u></a>	This method returns an array of a given object's own property names.
17	<a href="#"><u>Object.preventExte nsions()</u></a>	This method is used to prevent any extensions of an object.
18	<a href="#"><u>Object.seal()</u></a>	This method prevents new properties from being added and marks all existing properties as non-configurable.

19	<a href="#">Object.setPrototypeOf()</a>	This method sets the prototype of a specified object to another object.
20	<a href="#">Object.values()</a>	This method returns an array of values.

## Document Object Model

1. [Document Object](#)
2. [Properties of document object](#)
3. [Methods of document object](#)
4. [Example of document object](#)

The **document object** represents the whole html document.

When html document is loaded in the browser, it becomes a document object. It is the **root element** that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

As mentioned earlier, it is the object of window. So

1. window.document

Is same as

1. document

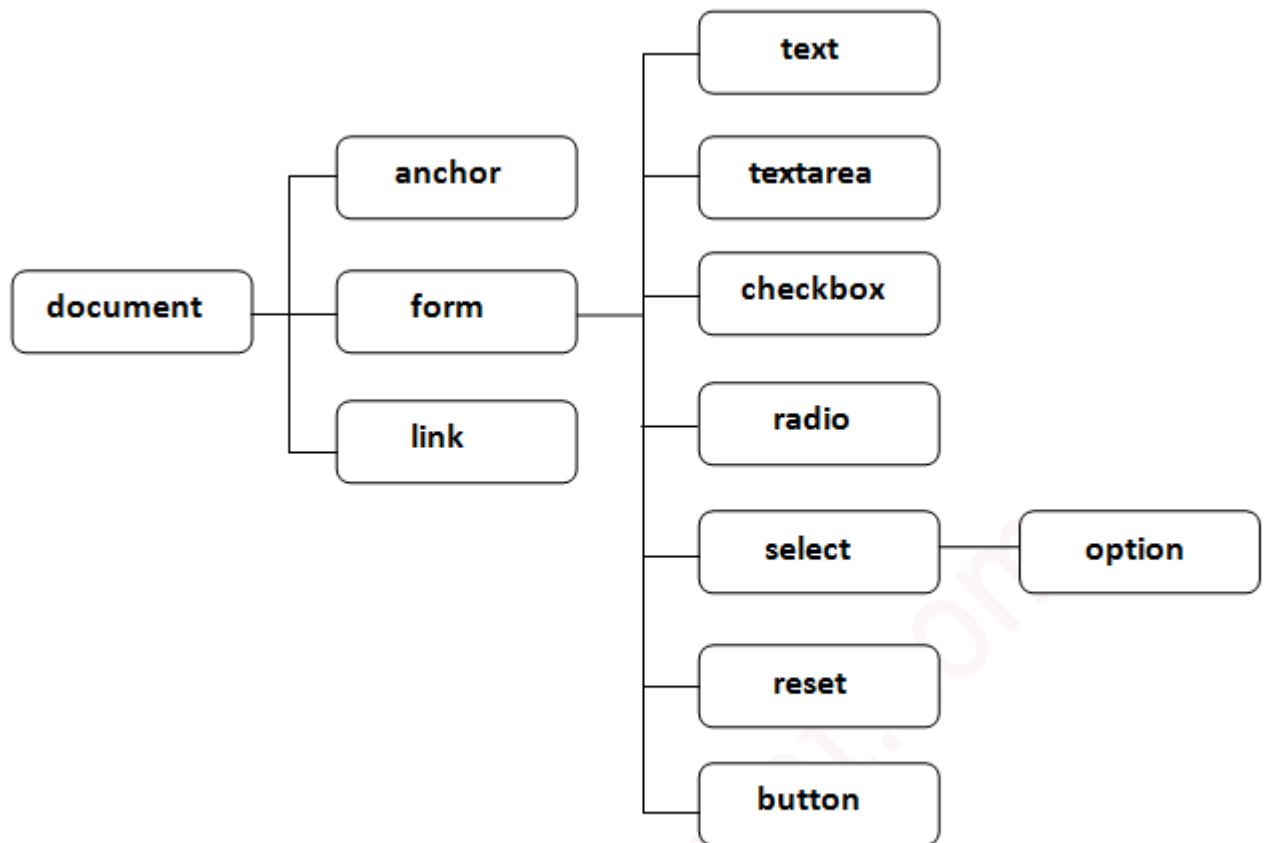
According to W3C - *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

---

## Properties of document object

Let's see the properties of document object that can be accessed and modified by the document

object.



## Methods of document object

We can access and change the contents of document by its methods.

The important methods of document object are as follows:

Method	Description
write("string")	writes the given string on the document.
writeln("string")	writes the given string on the document with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.

getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.

## Accessing field value by document object

In this example, we are going to get the value of input text by user. Here, we are using **document.form1.name.value** to get the value of name field.

Here, **document** is the root element that represents the html document.

**form1** is the name of the form.

**name** is the attribute name of the input text.

**value** is the property, that returns the value of the input text.

Let's see the simple example of document object that prints name with welcome message.

1. `<script type="text/javascript">`
2. `function printvalue(){`
3. `var name=document.form1.name.value;`
4. `alert("Welcome: "+name);`
5. `}`
6. `</script>`
- 7.
8. `<form name="form1">`
9. `Enter Name:<input type="text" name="name"/>`
10. `<input type="button" onclick="printvalue()" value="print name"/>`
11. `</form>`

*Output of the above example*

Enter Name:

## Document Object Model

1. [Document Object](#)
2. [Properties of document object](#)
3. [Methods of document object](#)
4. [Example of document object](#)

The **document object** represents the whole html document.

When html document is loaded in the browser, it becomes a document object. It is the **root element** that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

As mentioned earlier, it is the object of window. So

1. window.document

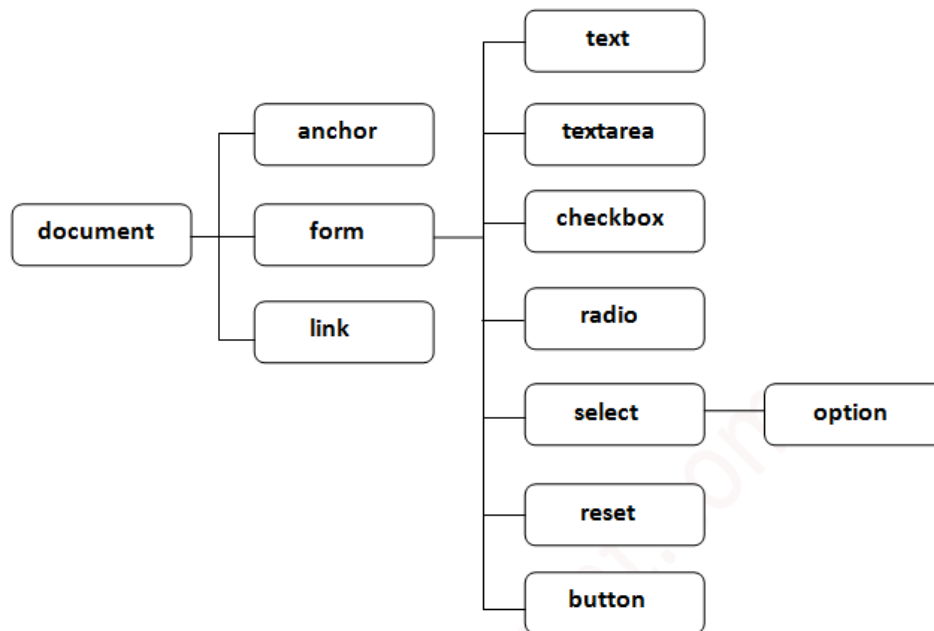
Is same as

1. document

According to W3C - *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*

## Properties of document object

Let's see the properties of document object that can be accessed and modified by the document object.



## Methods of document object

We can access and change the contents of document by its methods.

The important methods of document object are as follows:

Method	Description
write("string")	writes the given string on the document.
writeln("string")	writes the given string on the document with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.
getElementsByTagName()	returns all the elements having the given tag name.
getElementsByClassName()	returns all the elements having the given class name.



## Accessing field value by document object

In this example, we are going to get the value of input text by user. Here, we are using **document.form1.name.value** to get the value of name field.

Here, **document** is the root element that represents the html document.

**form1** is the name of the form.

**name** is the attribute name of the input text.

**value** is the property, that returns the value of the input text.

Let's see the simple example of document object that prints name with welcome message.

1. `<script type="text/javascript">`
2. `function printvalue(){`
3. `var name=document.form1.name.value;`
4. `alert("Welcome: "+name);`
5. `}`
6. `</script>`
- 7.
8. `<form name="form1">`
9. Enter Name:`<input type="text" name="name"/>`
10. `<input type="button" onclick="printvalue()" value="print name"/>`
11. `</form>`

---

*Output of the above example*

The screenshot shows a dark-themed web interface. On the left, the text 'Enter Name:' is displayed in a light color. To its right is a white rectangular text input field. Further to the right is a light-colored button with the text 'print name' on it.

## How to call JavaScript function in html?

There are many ways to call a JavaScript function in the HTML document, and it is also not a difficult task. First, we have used one of the easiest ways to call a [JavaScript](#)

function in HTML document:

In this method, we will create and define a function in the [HTML](#)

document's head section. To invoke this function in the html document, we have to create a simple button and using the **onclick** event attribute (which is an event handler) along with it, we can call the function by clicking on the button.

### To understand it more clearly let's see the given program

1. `<html>`
2. `<head>`
3. `<script type = "text/javascript">`
4. `function myfunction() {`
5. `alert("how are you");`
6. `}`
7. `</script>`
8. `</head>`
9. `<body>`
10. `<p>Click the following button to see the function in action</p>`
11. `<input type = "button" onclick = "myfunction()" value = "Display">`
12. `</body>`
13. `</html>`

### Explanation of program

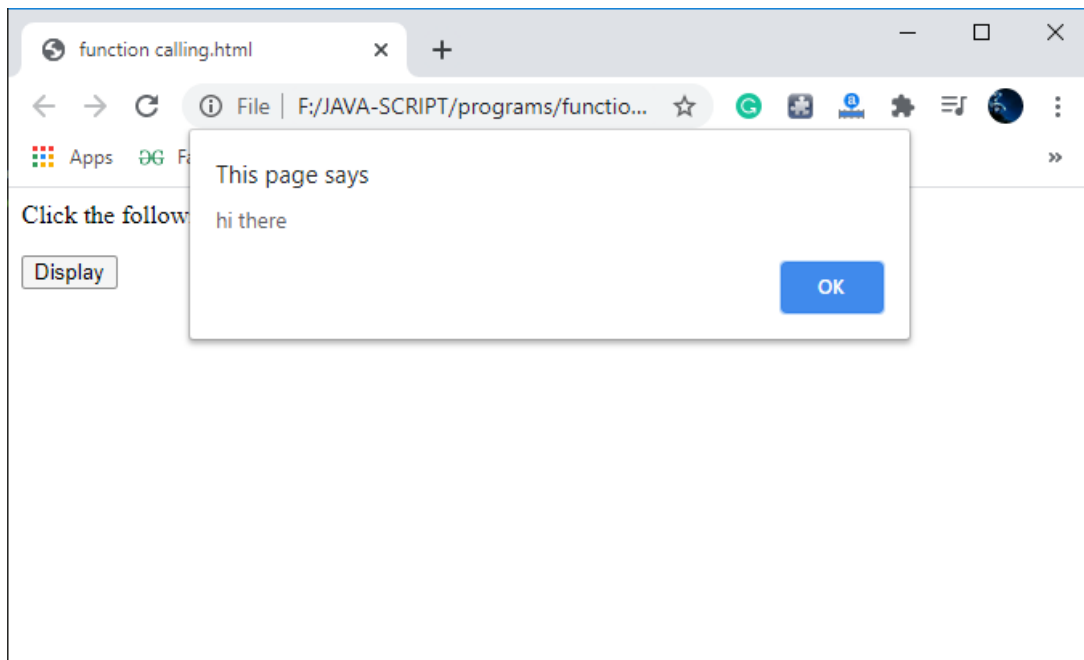
In the above-given program, we have created a simple HTML document. Inside the head section of the HTML document, we have defined a function (**e.g myfunction();**) inside the script tags `<script>...</script>`.

1. `<html>`
2. `<head>`
3. `<script type = "text/javascript">`
4. `function myfunction() {`
5. `alert("how are you");`
6. `}`
7. `</script>`

On the other hand, inside the body section, we displayed some text and created a button. To call our function, we have used the **onclick** attribute along with the button and when the user clicks on that button our function gets executed and displays an alert message, as you can see in the output.

1. `<body>`
2. `<p>Click the following button to see the function in action</p>`
3. `<input type = "button" onclick = "myfunction()" value = "Display">`
4. `</body>`

## Output



## Calling a function using external JavaScript file

We can also call JavaScript functions using an external JavaScript file attached to our HTML document. To do this, first we have to create a JavaScript file and define our function in it and save it with (.Js) extension.

Once the JavaScript file is created, we need to create a simple HTML document. To include our JavaScript file in the HTML document, we have to use the script tag `<script type = "text/javascript" src = "function.js">` and in the "src" attribute we have to provide the path to our JavaScript file where it is stored. After linking the external JavaScript file to the HTML document, we can create a button and call the function

using the "**onclick**" attribute with it.

### Let's understand it with help of a program:

Program

1. `<html>`
2. `<head>`
3. `<script type = "text/javascript" src="function.js"> </script>`
4. `</head>`
5. `<body>`
6. `<p>Click the following button to see the function in action</p>`
7. `<input type = "button" onclick = "myfunction()" value = "Display">`
8. `</body>`
9. `</html>`

## Explanation of program

In the above program first, we have created a JavaScript file and defined our function in it and saved it with the **.js** extension.

Function.js

1. `functionmyfunction()`
2. `{`
3. `document.write("welcome to Javatpoint");`
4. `}`

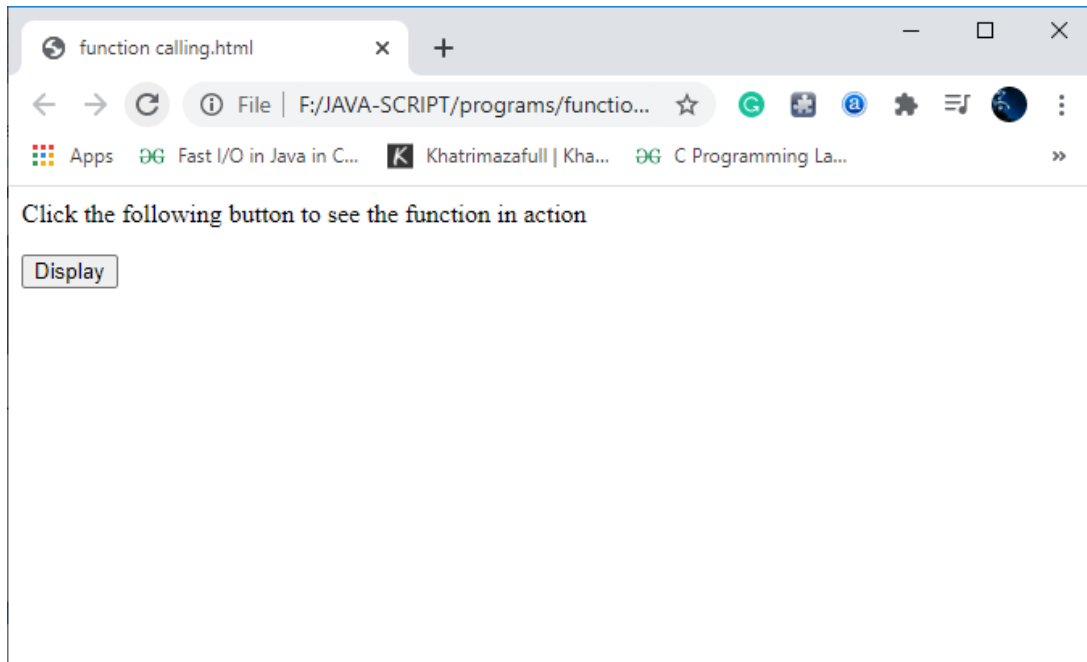
After creating the JavaScript file, we have created an HTML document and linked our JavaScript file using `<script type = "text/javascript" src="function.js"></script>`. Because we have stored our HTML document and JavaScript file in the same folder, we have just named our JavaScript file in the "src" attribute instead of providing the full path in the head section.

1. `<head>`
2. `<script type = "text/javascript" src="function.js"></script>`
3. `</head>`

Inside the body section, we displayed some text and created a button. To call our function, we have used the **onclick** attribute along with the button and when the user clicks on that button our function gets executes and display an alert message, as you can see in the output.

1. `<body>`
2. `<p>Click the following button to see the function in action</p>`
3. `<input type = "button" onclick = "myfunction()" value = "Display">`
4. `</body>`

## Output



## What is jQuery

- jQuery is a small and lightweight JavaScript library.
- jQuery is cross-platform.
- jQuery means "write less do more".
- jQuery simplifies AJAX call and DOM manipulation.

## jQuery Example

In this tutorial, you will get a lot of jQuery examples to understand the topic well. Let's see a simple jQuery example.

*File: firstjquery.html*

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>First jQuery Example</title>`
5. `<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">`
6. `</script>`

```
7. <script type="text/javascript" language="javascript">
8. $(document).ready(function() {
9.   $("p").css("background-color", "pink");
10. });
11. </script>
12. </head>
13. <body>
14. <p>This is first paragraph.</p>
15. <p>This is second paragraph.</p>
16. <p>This is third paragraph.</p>
17. </body>
18. </html>
```

Output:

This is first paragraph.

This is second paragraph.

This is third paragraph.

## jQuery

jQuery is a fast, small, cross-platform and feature-rich JavaScript library. It is designed to simplify the client-side scripting of HTML. It makes things like HTML document traversal and manipulation, animation, event handling, and AJAX very simple with an easy-to-use API that works on a lot of different type of browsers.

The main purpose of jQuery is to provide an easy way to use JavaScript on your website to make it more interactive and attractive. It is also used to add animation.

## What is jQuery

jQuery is a small, light-weight and fast JavaScript library. It is cross-platform and supports different types of browsers. It is also referred as "write less do more" because it takes a lot of common tasks that requires many lines of JavaScript code to accomplish, and binds them into methods that can be called with a single line of code whenever needed. It is also very useful to simplify a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

- jQuery is a small, fast and lightweight JavaScript library.
- jQuery is platform-independent.
- jQuery means "write less do more".
- jQuery simplifies AJAX call and DOM manipulation.

# jQuery Features

Following are the important features of jQuery.

- HTML manipulation
- DOM manipulation
- DOM element selection
- CSS manipulation
- Effects and Animations
- Utilities
- AJAX
- HTML event methods
- JSON Parsing
- Extensibility through plug-ins

## Why jQuery is required

Sometimes, a question can arise that what is the need of jQuery or what difference it makes on bringing jQuery instead of AJAX/ JavaScript? If jQuery is the replacement of AJAX and JavaScript? For all these questions, you can state the following answers.

- It is very fast and extensible.
- It facilitates the users to write UI related function codes in minimum possible lines.
- It improves the performance of an application.
- Browser's compatible web applications can be developed.
- It uses mostly new features of new browsers.

So, you can say that out of the lot of JavaScript frameworks, jQuery is the most popular and the most extendable. Many of the biggest companies on the web use jQuery.

Some of these companies are:

- Microsoft
- Google
- IBM
- Netflix

# jQuery Example

jQuery is developed by Google. To create the first jQuery example, you need to use JavaScript file for jQuery. You can download the jQuery file from [jquery.com](http://jquery.com) or use the absolute URL of jQuery file.

In this jQuery example, we are using the absolute URL of jQuery file. The jQuery example is written inside the script tag.

Let's see a simple example of jQuery.

*File: firstjquery.html*

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>First jQuery Example</title>`
5. `<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">`
6. `</script>`
7. `<script type="text/javascript" language="javascript">`
8. `$(document).ready(function() {`
9. `$("#p").css("background-color", "cyan");`
10. `});`
11. `</script>`
12. `</head>`
13. `<body>`
14. `<p>The first paragraph is selected.</p>`
15. `<p>The second paragraph is selected.</p>`
16. `<p>The third paragraph is selected.</p>`
17. `</body>`
18. `</html>`

Output:

The first paragraph is selected.

The second paragraph is selected.

The third paragraph is selected.

`$(document).ready()` and `$()`



The code inserted between `$(document).ready()` is executed only once when page is ready for JavaScript code to execute.

In place of `$(document).ready()`, you can use shorthand notation `$()` only.

1. `$(document).ready(function() {`
2. `$("#p").css("color", "red");`
3. `});`

The above code is equivalent to this code.

1. `$(function() {`
2. `$("#p").css("color", "red");`
3. `});`

Let's see the full example of jQuery using shorthand notation `$()`.

*File: shortjquery.html*

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Second jQuery Example</title>`
5. `<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">`
6. `</script>`
7. `<script type="text/javascript" language="javascript">`
8. `$(function() {`
9. `$("#p").css("color", "red");`
10. `});`
11. `</script>`
12. `</head>`
13. `<body>`
14. `<p>The first paragraph is selected.</p>`
15. `<p>The second paragraph is selected.</p>`
16. `<p>The third paragraph is selected.</p>`
17. `</body>`
18. `</html>`

Output:

The first paragraph is selected.

The second paragraph is selected.

The third paragraph is selected.

```
function() { $("p").css("background-color", "cyan"); }
```

It changes the background-color of all <p> tag or paragraph to cyan.

## How to add jQuery to Html

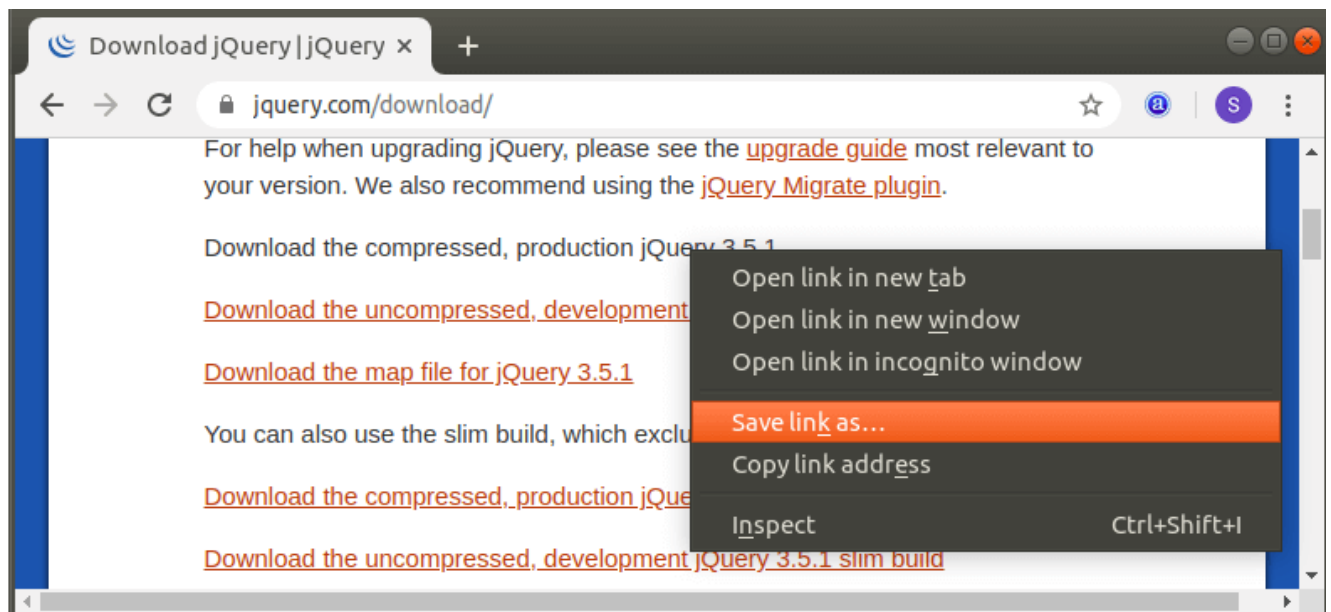
Following are the two different ways for adding the jQuery to Html page:

1. Download and Include jQuery file
2. Include the jQuery by CDN.

### Download and Include jQuery File

If we want to add the jQuery to [Html](#) page by downloading the jQuery file, then we have to follow the steps which are given below. Using these steps, any user can easily add the jQuery.

**Step 1:** Firstly, we have to download the [jquery](#) js file from the following official site of jQuery. <https://jquery.com/download/>



**Step 2:** When we have downloaded the file, then we have to open that Html file in which we want to add the jquery.

Competitive questions on Structures in Hindi

Keep Watching

1. <!Doctype Html>

2. `<Html>`
3. `<Head>`
4. `<Title>`
5. Add the jQuery file into Html by downloading and Including file
6. `</Title>`
7. `</Head>`
8. `<Body>`
9. Hello User!... `<br> <center>`
10. ` </center>`
11. `</Body>`
12. `</Html>`

**Step 3:** After then, we have to place the cursor between the head tag just before the title tag. And, then we have to use the `<script>` tag, which specify the src attribute for adding the jQuery file.

1. `<!Doctype Html>`
2. `<Html>`
3. `<Head>`
4. `<script type="text/javascript" src="jquery-3.5.1.min.js">`
5. `</script>`
6. `<Title>`
7. Add the jQuery file into Html by downloading and Including file
8. `</Title>`
9. `</Head>`
10. `<Body>`
11. Hello User!... `<br> <center>`
12. ` </center>`
13. `</Body>`
14. `</Html>`

**Step 4:** And, at last, save the Html file and the jQuery file is successfully added into our Html page.

## Include the jQuery by CDN

If we want to add the jQuery to Html page by the help of CDN, then we have to follow the steps which are given below. Using these steps, any user can easily add the jQuery.

**Step 1:** Firstly, we have to open that Html file in which we want to add the jQuery using CDN.

1. `<!Doctype Html>`
2. `<Html>`
3. `<Head>`
4. `<Title>`
5. Add the jQuery file into Html by including CDN
6. `</Title>`
7. `</Head>`
8. `<Body>`
9. Hello User!... `<br>` `<center>`
10. `` `</center>`
11. `</Body>`
12. `</Html>`

**Step 2:** After then, we have to place the cursor between the head tag just before the title tag. And, then we have to use the `<script>` tag, which specify the src attribute for adding.

1. `<!Doctype Html>`
2. `<Html>`
3. `<Head>`
4. `<script type="text/javascript" src=>`
5. `</script>`
6. `<Title>`
7. Add the jQuery file into Html by including CDN
8. `</Title>`
9. `</Head>`
10. `<Body>`
11. Hello User!... `<br>` `<center>`
12. `` `</center>`
13. `</Body>`
14. `</Html>`

**Step 3:** And then, we have to type the following path in the src attribute.

1. `http://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js`

**Step 4:** And, at last, save the Html file and the jQuery file is successfully added into our Html page.

1. `<!Doctype Html>`
2. `<Html>`
3. `<Head>`
4. `<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">`
5. `</script>`
6. `<Title>`
7. Add the jQuery file into Html by including CDN
8. `</Title>`
9. `</Head>`
10. `<Body>`
11. Hello User!... `<br>` `<center>`
12. `` `</center>`
13. `</Body>`
14. `</Html>`

## jQuery Selectors

jQuery Selectors are used to select and manipulate HTML elements. They are very important part of jQuery library.

With jQuery selectors, you can find or select HTML elements based on their id, classes, attributes, types and much more from a DOM.

In simple words, you can say that selectors are used to select one or more HTML elements using jQuery and once the element is selected then you can perform various operation on that.

All jQuery selectors start with a dollar sign and parenthesis e.g. `$()`. It is known as the factory function.

## The `$()` factory function

Every jQuery selector start with this sign `$()`. This sign is known as the factory function. It uses the three basic building blocks while selecting an element in a given document.

S.No.	Selector	Description
1)	Tag Name:	It represents a tag name available in the

		DOM. For example: \$('p') selects all paragraphs 'p' in the document.
2)	Tag ID:	It represents a tag available with a specific ID in the DOM. For example: \$('#real-id') selects a specific element in the document that has an ID of real-id.
3)	Tag Class:	It represents a tag available with a specific class in the DOM. For example: \$('.real-class') selects all elements in the document that have a class of real-class.

Let's take a simple example to see the use of Tag selector. This would select all the elements with a tag name

and the background color is set to "pink".

*File: firstjquery.html*

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>First jQuery Example</title>
5. <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
6. </script>
7. <script type="text/javascript" language="javascript">
8. \$(document).ready(function() {
9. \$("p").css("background-color", "pink");
10. });
11. </script>
12. </head>
13. <body>
14. <p>This is first paragraph.</p>
15. <p>This is second paragraph.</p>
16. <p>This is third paragraph.</p>
17. </body>
18. </html>

Output:

This is first paragraph.

This is second paragraph.

This is third paragraph.

*Note: 1. All of the above discussed selectors can be used alone or with the combination of other selectors.*

*Note: 2. If you have any conflict with the use of dollar sign \$ in any JavaScript library then you can use jQuery() function instead of factory function \$(). The factory function \$() and the jQuery function is the same.*

## How to use Selectors

The jQuery selectors can be used single or with the combination of other selectors. They are required at every steps while using jQuery. They are used to select the exact element that you want from your HTML document.

S.No.	Selector	Description
1)	Name:	It selects all elements that match with the given element name.
2)	#ID:	It selects a single element that matches with the given id.
3)	.Class:	It selects all elements that matches with the given class.
4)	Universal(*)	It selects all elements available in a DOM.
5)	Multiple Elements A,B,C	It selects the combined results of all the specified selectors A,B and C.

## Different jQuery Selectors

Selector	Example	Description
*	\$("#*")	It is used to select all elements.

#id	\$("#firstname")	It will select the element with id="firstname"
.class	\$(".primary")	It will select all elements with class="primary"
class,.class	\$(".primary,.secondary")	It will select all elements with the class "primary" or "secondary"
element	\$( <code>"p"</code> )	It will select all p elements.
el1,el2,el3	\$( <code>"h1,div,p"</code> )	It will select all h1, div, and p elements.
:first	\$( <code>"p:first"</code> )	This will select the first p element
:last	\$( <code>"p:last"</code> )	This will select the last p element
:even	\$( <code>"tr:even"</code> )	This will select all even tr elements
:odd	\$( <code>"tr:odd"</code> )	This will select all odd tr elements
:first-child	\$( <code>"p:first-child"</code> )	It will select all p elements that are the first child of their parent
:first-of-type	\$( <code>"p:first-of-type"</code> )	It will select all p elements that are the first p element of their parent
:last-child	\$( <code>"p:last-child"</code> )	It will select all p elements that are the last child of their parent
:last-of-type	\$( <code>"p:last-of-type"</code> )	It will select all p elements that are the last p element of their parent
:nth-child(n)	\$( <code>"p:nth-child(2)"</code> )	This will select all p elements that are the 2nd child of their parent
:nth-last-child(n)	\$( <code>"p:nth-last-</code>	This will select all p elements that are the 2nd child of their



	child(2)")	parent, counting from the last child
:nth-of-type(n)	\$("p:nth-of-type(2)")	It will select all p elements that are the 2nd p element of their parent
:nth-last-of-type(n)	\$("p:nth-last-of-type(2)")	This will select all p elements that are the 2nd p element of their parent, counting from the last child
:only-child	\$("p:only-child")	It will select all p elements that are the only child of their parent
:only-of-type	\$("p:only-of-type")	It will select all p elements that are the only child, of its type, of their parent
parent > child	\$("div > p")	It will select all p elements that are a direct child of a div element
parent descendant	\$("div p")	It will select all p elements that are descendants of a div element
element + next	\$("div + p")	It selects the p element that are next to each div elements
element ~ siblings	\$("div ~ p")	It selects all p elements that are siblings of a div element
:eq(index)	\$("ul li:eq(3)")	It will select the fourth element in a list (index starts at 0)
:gt(no)	\$("ul li:gt(3)")	Select the list elements with an index greater than 3
:lt(no)	\$("ul li:lt(3)")	Select the list elements with an index less than 3
:not(selector)	\$("input:not(:empty)")	Select all input elements that are not empty
:header	\$(":header")	Select all header elements h1, h2 ...

:animated	\$(":animated")	Select all animated elements
:focus	\$(":focus")	Select the element that currently has focus
:contains(text)	\$(":contains('Hello')")	Select all elements which contains the text "Hello"
:has(selector)	\$("div:has(p)")	Select all div elements that have a p element
:empty	\$(":empty")	Select all elements that are empty
:parent	\$(":parent")	Select all elements that are a parent of another element
:hidden	\$("p:hidden")	Select all hidden p elements
:visible	\$("table:visible")	Select all visible tables
:root	\$(":root")	It will select the document's root element
:lang(language)	\$("p:lang(de)")	Select all p elements with a lang attribute value starting with "de"
[attribute]	\$("[href]")	Select all elements with a href attribute
[attribute=value]	\$("[href='default.htm']")	Select all elements with a href attribute value equal to "default.htm"
[attribute!=value]	\$("[href!='default.htm']")	It will select all elements with a href attribute value not equal to "default.htm"
[attribute\$=value]	\$("[href\$='.jpg']")	It will select all elements with a href attribute value ending with ".jpg"
[attribute =value]	\$("[title='Tomorrow']")	Select all elements with a title attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen

[attribute^=value]	\$("[title^='Tom']")	Select all elements with a title attribute value starting with "Tom"
[attribute~=value]	\$("[title~='hello']")	Select all elements with a title attribute value containing the specific word "hello"
[attribute*=value]	\$("[title*='hello']")	Select all elements with a title attribute value containing the word "hello"
:input	\$(":input")	It will select all input elements
:text	\$(":text")	It will select all input elements with type="text"
:password	\$(":password")	It will select all input elements with type="password"
:radio	\$(":radio")	It will select all input elements with type="radio"
:checkbox	\$(":checkbox")	It will select all input elements with type="checkbox"
:submit	\$(":submit")	It will select all input elements with type="submit"
:reset	\$(":reset")	It will select all input elements with type="reset"
:button	\$(":button")	It will select all input elements with type="button"
:image	\$(":image")	It will select all input elements with type="image"
:file	\$(":file")	It will select all input elements with type="file"
:enabled	\$(":enabled")	Select all enabled input elements
:disabled	\$(":disabled")	It will select all disabled input elements
:selected	\$(":selected")	It will select all selected input elements
:checked	\$(":checked")	It will select all checked input elements

# jQuery Event Methods

This section contains a comprehensive list of event methods belonging to the latest jQuery JavaScript library. All the methods are grouped into categories.

## Mouse Events

Method	Description
<code>click()</code>	Bind an event handler to be fired when the element is clicked, or trigger that handler on an element.
<code>dblclick()</code>	Bind an event handler to be fired when the element is double-clicked, or trigger that event on an element.
<code>hover()</code>	Bind one or two handlers to the selected elements, to be executed when the mouse pointer enters and leaves the elements.
<code>mousedown()</code>	Bind an event handler to be fired when the mouse button is pressed within the element, or trigger that event on an element.
<code>mouseenter()</code>	Bind an event handler to be fired when the mouse enters an element, or trigger that handler on an element.
<code>mouseleave()</code>	Bind an event handler to be fired when the mouse leaves an element, or trigger that handler on an element.
<code>mousemove()</code>	Bind an event handler to be fired when the mouse pointer moves within the element, or trigger that event on an element.
<code>mouseout()</code>	Bind an event handler to be fired when the mouse pointer leaves the element, or trigger that event on an element.
<code>mouseover()</code>	Bind an event handler to be fired when the mouse pointer enters the element, or trigger that event on an element.

<code>mouseup()</code>	Bind an event handler to be fired when the mouse button is released within the element, or trigger that event on an element.
<code>toggle()</code>	Bind two or more handlers to the selected elements, to be executed on alternate clicks.  Deprecated in favor of <code>toggle()</code> animation method.

## Keyboard Events

Method	Description
<code>keydown()</code>	Bind an event handler to be fired when a key is pressed and the element has keyboard focus, or trigger that event on an element.
<code>keypress()</code>	Bind an event handler to be fired when a keystroke occurs and the element has keyboard focus, or trigger that event on an element.
<code>keyup()</code>	Bind an event handler to be fired when a key is released and the element has keyboard focus , or trigger that event on an element.

## Form Events

Method	Description
<code>blur()</code>	Bind an event handler to be fired when the element loses keyboard focus, or trigger that event on an element.
<code>change()</code>	Bind an event handler to be fired when the element's value changes, or trigger that event on an element.
<code>focus()</code>	Bind an event handler to be fired when the element gains keyboard focus, or trigger that event on an element.
<code>focusin()</code>	Bind an event handler to be fired when the element, or a descendant, gains keyboard focus.

<code>focusout()</code>	Bind an event handler to be fired when the element, or a descendant, loses keyboard focus.
<code>select()</code>	Bind an event handler to be fired when text in the element is selected, or trigger that event on an element.
<code>submit()</code>	Bind an event handler to be fired when the form element is submitted, or trigger that event on an element.

## Document/Browser Events

Method	Description
<code>error()</code>	Bind an event handler to be fired if the element was not loaded correctly.
<code>load()</code>	Bind an event handler to be fired when the element finishes loading. Deprecated in favor of Ajax <code>load()</code> method.
<code>ready()</code>	Bind an event handler to be fired when the DOM is fully loaded.
<code>resize()</code>	Bind an event handler to be fired when the element is resized, or trigger that event on an element.
<code>scroll()</code>	Bind an event handler to be fired when the window's or element's scroll position changes, or trigger that event on an element.
<code>unload()</code>	Bind an event handler to be fired when the user navigates away from the page.

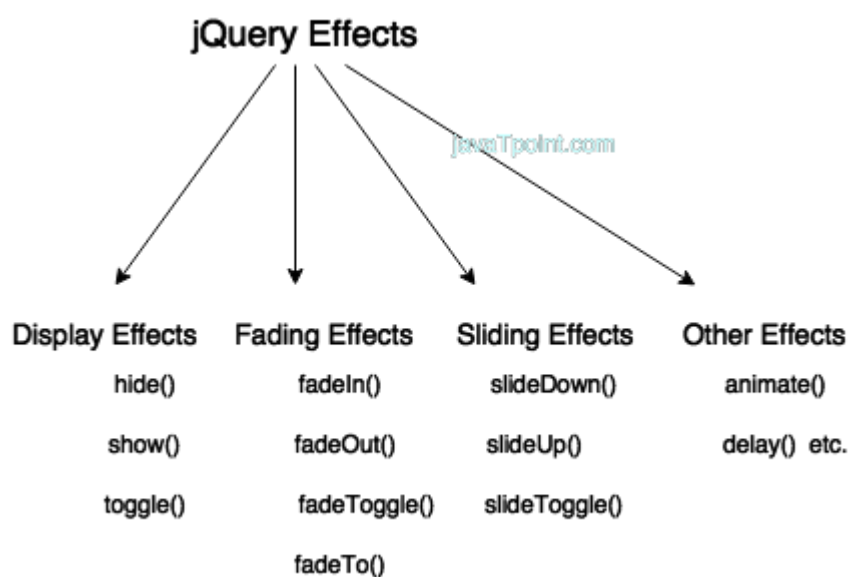
## Event Handler Attachment

Method	Description
<code>bind()</code>	Bind an event handler to be fired when the given type of event is sent to the element.  In general, use the <code>on()</code> method instead.
<code>delegate()</code>	Bind one or more event handlers to be fired when the given type of event is/are sent to a

	descendant element matching selector.
<code>jQuery.proxy()</code>	Takes an existing function and returns a new one that will always have a particular context.
<code>on()</code>	Attaches event handlers to the selected elements.
<code>off()</code>	Removes event handlers that were with the <code>on()</code> method.
<code>one()</code>	Attaches one or more event handlers to selected elements. The handler is executed at most once per element.
<code>trigger()</code>	Execute all handlers and behaviors attached to the selected elements for the given event type.
<code>triggerHandler()</code>	Execute all handlers attached to the selected elements for an event.
<code>unbind()</code>	Remove a previously-attached event handler from the elements.
<code>undelegate()</code>	Removes the event handlers bindings on the element that have been previously bound with <code>delegate()</code> method.

## jQuery Effects

jQuery enables us to add effects on a web page. jQuery effects can be categorized into fading, sliding, hiding/showing and animation effects.



jQuery provides many methods for effects on a web page. A complete list of jQuery effect methods are given below:

No.	Method	Description
1)	animate()	performs animation.
2	clearQueue()	It is used to remove all remaining queued functions from the selected elements.
3)	delay()	sets delay execution for all the queued functions on the selected elements.
4	dequeue()	It is used to remove the next function from the queue, and then execute the function.
5)	fadeIn()	shows the matched elements by fading it to opaque. In other words, it fades in the selected elements.
6)	fadeOut()	shows the matched elements by fading it to transparent. In other words, it fades out the selected elements.
7)	fadeTo()	adjusts opacity for the matched element. In other words, it fades in/out the selected elements.
8)	fadeToggle()	shows or hides the matched element. In other words, toggles between the fadeIn() and fadeOut() methods.
9)	finish()	It stops, removes and complete all queued animation for the selected elements.
10)	hide()	hides the matched or selected elements.
11)	queue()	shows or manipulates the queue of methods i.e. to be executed on the selected elements.
12)	show()	displays or shows the selected elements.



13)	slideDown()	shows the matched elements with slide.
14)	slideToggle()	shows or hides the matched elements with slide. In other words, it is used to toggle between the slideUp() and slideDown() methods.
15)	slideUp()	hides the matched elements with slide.
16)	stop()	stops the animation which is running on the matched elements.
17)	toggle()	shows or hides the matched elements. In other words, it toggles between the hide() and show() methods.

## jQuery hide()

The jQuery hide() method is used to hide the selected elements.

### Syntax:

1. \$(selector).hide();
2. \$(selector).hide(speed, callback);
3. \$(selector).hide(speed, easing, callback);

**speed:** It is an optional parameter. It specifies the speed of the delay. Its possible values are slow, fast and milliseconds.

**easing:** It specifies the easing function to be used for transition.

**callback:** It is also an optional parameter. It specifies the function to be called after completion of hide() effect.

Let's take an example to see the jQuery hide effect.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"> </script>
5. <script>
6. \$(document).ready(function(){
7.     \$("#hide").click(function(){
8.         \$("p").hide();

9. `});`
10. `});`
11. `</script>`
12. `</head>`
13. `<body>`
14. `<p>`
15. `<b>`This is a little poem: `</b><br/>`
16. Twinkle, twinkle, little star`<br/>`
17. How I wonder what you are`<br/>`
18. Up above the world so high`<br/>`
19. Like a diamond in the sky`<br/>`
20. Twinkle, twinkle little star`<br/>`
21. How I wonder what you are
22. `</p>`
23. `<button id="hide">`Hide`</button>`
24. `</body>`
25. `</html>`

Output: This is a little poem

Twinkle, twinkle, little star  
How I wonder what you are  
Up above the world so high  
Like a diamond in the sky  
Twinkle, twinkle little star  
How I wonder what you are

## jQuery show()

The jQuery show() method is used to show the selected elements.

### Syntax:

1. `$(selector).show();`
2. `$(selector).show(speed, callback);`
3. `$(selector).show(speed, easing, callback);`

**speed:** It is an optional parameter. It specifies the speed of the delay. Its possible values are slow, fast and milliseconds.

**easing:** It specifies the easing function to be used for transition.

**callback:** It is also an optional parameter. It specifies the function to be called after completion of show() effect.

Let's take an example to see the jQuery show effect.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"> </script>
5. <script>
6. $(document).ready(function(){
7.     $("#hide").click(function(){
8.         $("p").hide();
9.     });
10.    $("#show").click(function(){
11.        $("p").show();
12.    });
13. });
14. </script>
15. </head>
16. <body>
17. <p>
18. <b>This is a little poem: </b><br/>
19. Twinkle, twinkle, little star<br/>
20. How I wonder what you are<br/>
21. Up above the world so high<br/>
22. Like a diamond in the sky<br/>
23. Twinkle, twinkle little star<br/>
24. How I wonder what you are
25. </p>
26. <button id="hide">Hide</button>
27. <button id="show">Show</button>
28. </body>
29. </html>
```

Output: This is a little poem

Twinkle, twinkle, little star  
How I wonder what you are

Up above the world so high  
Like a diamond in the sky  
Twinkle, twinkle little star  
How I wonder what you are

## jQuery show() effect with speed parameter

Let's see the example of jQuery show effect with 1500 milliseconds speed.

```
1. $(document).ready(function(){  
2.     $("#hide").click(function(){  
3.         $("p").hide(1000);  
4.     });  
5.     $("#show").click(function(){  
6.         $("p").show(1500);  
7.     });  
8. });
```

## jQuery fadeIn()

jQuery fadeIn() method is used to fade in the element.

### Syntax:

```
1. $(selector).fadeIn();  
2. $(selector).fadeIn(speed,callback);  
3. $(selector).fadeIn(speed, easing, callback);
```

**speed:** It is an optional parameter. It specifies the speed of the delay. Its possible values are slow, fast and milliseconds.

**easing:** It specifies the easing function to be used for transition.

**callback:** It is also an optional parameter. It specifies the function to be called after completion of fadeIn() effect.

Let's take an example to demonstrate jQuery fadeIn() effect.

```
1. <!DOCTYPE html>  
2. <html>  
3. <head>  
4. <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"> </script>  
5. <script>
```

```

6. $(document).ready(function(){
7.     $("button").click(function(){
8.         $("#div1").fadeIn();
9.         $("#div2").fadeIn("slow");
10.        $("#div3").fadeIn(3000);
11.    });
12. });
13. </script>
14. </head>
15. <body>
16. <p>See the fadeIn() method example with different parameters.</p>
17. <button>Click to fade in boxes</button><br><br>
18. <div id="div1" style="width:80px;height:80px;display:none;background-
    color:red;"></div><br>
19. <div id="div2" style="width:80px;height:80px;display:none;background-
    color:green;"></div><br>
20. <div id="div3" style="width:80px;height:80px;display:none;background-
    color:blue;"></div>
21. </body>
22. </html>

```

Output:

See the fadeIn() method example with different parameters.

## jQuery fadeOut()

The jQuery fadeOut() method is used to fade out the element.

### Syntax:

1. \$(selector).fadeOut();
2. \$(selector).fadeOut(speed,callback);
3. \$(selector).fadeOut(speed, easing, callback);

**speed:** It is an optional parameter. It specifies the speed of the delay. Its possible values are slow, fast and milliseconds.

**easing:** It specifies the easing function to be used for transition.

**callback:** It is also an optional parameter. It specifies the function to be called after completion of fadeOut() effect.

Let's take an example to demonstrate jQuery fadeOut() effect.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
5. <script>
6. \$(document).ready(function(){
7.     \$("#button").click(function(){
8.         \$("#div1").fadeOut();
9.         \$("#div2").fadeOut("slow");
10.        \$("#div3").fadeOut(3000);
11.     });
12. });
13. </script>
14. </head>
15. <body>
16. <p>See the fadeOut() method example with different parameters.</p>
17. <button>Click to fade out boxes</button> <br> <br>
18. <div id="div1" style="width:80px;height:80px;background-color:red;"></div> <br>
19. <div id="div2" style="width:80px;height:80px;background-color:green;"></div> <br>
20. <div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
21. </body>
22. </html>

Output:

See the fadeOut() method example with different parameters.

## jQuery slideDown()

jQuery slideDown() method is used to slide down an element.

### Syntax:

1. \$(selector).slideDown(speed);
2. \$(selector).slideDown(speed, callback);
3. \$(selector).slideDown(speed, easing, callback);

**speed:** It specifies the speed of the delay. Its possible values are slow, fast and milliseconds.

**easing:** It specifies the easing function to be used for transition.

**callback:** It is also an optional parameter. It specifies the function to be called after completion of slideDown() effect.

Let's take an example to demonstrate jQuery slideDown() effect.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></scri
   pt>
5. <script>
6. $(document).ready(function(){
7.     $("#flip").click(function(){
8.         $("#panel").slideDown("slow");
9.     });
10. });
11. </script>
12. <style>
13. #panel, #flip {
14.     padding: 5px;
15.     text-align: center;
16.     background-color: #00FFFF;
17.     border: solid 1px #c3c3c3;
18. }
19. #panel {
20.     padding: 50px;
21.     display: none;
22. }
23. </style>
24. </head>
25. <body>
26. <div id="flip">Click to slide down panel</div>
27. <div id="panel">Hello javatpoint.com!
28. It is the best tutorial website to learn jQuery and other languages.</div>
29. </body>
30. </html>
```

## jQuery slideUp()

jQuery slideDown() method is used to slide up an element.

**Syntax:**

1. `$(selector).slideUp(speed);`
2. `$(selector).slideUp(speed, callback);`
3. `$(selector).slideUp(speed, easing, callback);`

**speed:** It specifies the speed of the delay. Its possible values are slow, fast and milliseconds.

**easing:** It specifies the easing function to be used for transition.

**callback:** It is also an optional parameter. It specifies the function to be called after completion of `slideUp()` effect.

Let's take an example to demonstrate jQuery `slideUp()` effect.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>`
5. `<script>`
6. `$(document).ready(function(){`
7.  `$("#flip").click(function(){`
8.  `$("#panel").slideUp("slow");`
9.  `});`
10. `});`
11. `</script>`
12. `<style>`
13. `#panel, #flip {`
14.  `padding: 5px;`
15.  `text-align: center;`
16.  `background-color: #00FFFF;`
17.  `border: solid 1px #c3c3c3;`
18. `}`
19. `#panel {`
20.  `padding: 50px;`
21. `}`
22. `</style>`
23. `</head>`
24. `<body>`
25. `<div id="flip">Click to slide up panel</div>`
26. `<div id="panel">Hello javatpoint.com!`
27. `It is the best tutorial website to learn jQuery and other languages.</div>`
28. `</body>`



29. `</html>`

Output:

Click to slide up panel

Hello javatpoint.com! It is the best tutorial website to learn jQuery and other languages.

## What is Bootstrap

- Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
- It is absolutely free to download and use.
- It is a front-end framework used for easier and faster web development.
- It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.
- It can also use JavaScript plug-ins.
- It facilitates you to create responsive designs.



---

## History of Bootstrap

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source product in August 2011 on GitHub.

In June 2014 Bootstrap was the No.1 project on GitHub.

---

## Why use Bootstrap

Following are the main advantage of Bootstrap:

- It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
  - It facilitates users to develop a responsive website.
  - It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.
- 

## What is a responsive website

A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc.

---

## What Bootstrap package contains

**Scaffolding:** Bootstrap provides a basic structure with Grid System, link styles, and background.

**CSS:** Bootstrap comes with the feature of global CSS settings, fundamental HTML elements style and an advanced grid system.

**Components:** Bootstrap contains a lot of reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.

**JavaScript Plugins:** Bootstrap also contains a lot of custom jQuery plugins. You can easily include them all, or one by one.

**Customize:** Bootstrap components are customizable and you can customize Bootstrap's components, LESS variables, and jQuery plugins to get your own style.

---

## What is Bootstrap 4?

Bootstrap is the newest and latest version of Bootstrap. It is the most popular HTML, CSS, JavaScript framework for developing responsive, mobile first websites.

---

## Bootstrap 3 vs. Bootstrap 4

Bootstrap 4 has some new components, faster stylesheet, more buttons, effects and more responsiveness.

Bootstrap 4 supports some the latest, stable releases of all major browsers and platforms.

## Is Bootstrap Best?

Bootstrap is more than efficient to create a responsive and mobile first website but it is not the best in the industry. There is an alternative of Bootstrap named W3.CSS which is smaller, faster, and easier to use.

## Colors

By using utility classes you can change the color of the text. It can also be used to links and will darken on hover just like our default link styles.

### Syntax:

#### Muted Class:

1. `<p class="text-muted">Fusce dapibus, tellus ac cursus commodo, tortor mauris nibh.</p>`

#### Primary Class:

1. `<p class="text-primary">Nullam id dolor id nibh ultricies vehicula ut id elit.</p>`

#### Success:

1. `<p class="text-success">Duis mollis, est non commodo luctus, nisi erat porttitor ligula.</p>`

#### Info:

1. `<p class="text-info">Maecenas sed diam eget risus varius blandit sit amet non magna.</p>`

### Warning:

1. `<p class="text-warning">Etiam porta sem malesuada magna mollis euismod.</p>`

### Danger:

1. `<p class="text-danger">Donec ullamcorper nulla non metus auctor fringilla.</p>`

### White:

- 1.
2. `<p class="text-white">Etiam porta sem malesuada ultricies vehicula.</p>`

Contextual text classes can also be used with anchors tags with the provided hover and focus states.

*Note: The .text-white class has no link styling.*

### Example:

- 1.
2. `<a href="#" class="text-muted">Muted link</a>`
3. `<a href="#" class="text-primary">Primary link</a>`
4. `<a href="#" class="text-success">Success link</a>`
5. `<a href="#" class="text-info">Info link</a>`
6. `<a href="#" class="text-warning">Warning link</a>`
7. `<a href="#" class="text-danger">Danger link</a>`

## Bootstrap Inputs

---

### Bootstrap Input Types

Input type: checkbox

Input type: radio button

---

### Bootstrap Checkbox

Checkbox facilitates you to select any number of options from a list of present options.

**See this example:**

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
   ap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13.   <h2>Input type: checkbox</h2>
14.   <form role="form">
15.     <div class="checkbox">
16.       <label><input type="checkbox" value="">Option 1</label>
17.     </div>
18.     <div class="checkbox">
19.       <label><input type="checkbox" value="">Option 2</label>
20.     </div>
21.   </div>
22. </form>
23. </div>
24. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
25. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scri
   pt>
26. </body>
27. </html>
```

Use the **.checkbox-inline** class if you want the checkboxes to appear on the same line:

**See this example:**

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
```

7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
- 8.
9. `</head>`
10. `<body>`
- 11.
12. `<div class="container">`
13. `<h2>Inline checkbox</h2>`
14. `<p>The following checkboxes appear in the same line :</p>`
15. `<form role="form">`
16. `<label class="checkbox-inline">`
17. `<input type="checkbox" value="">Option 1`
18. `</label>`
19. `<label class="checkbox-inline">`
20. `<input type="checkbox" value="">Option 2`
21. `</label>`
22. `</form>`
23. `</div>`
24. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
25. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
26. `</body>`
27. `</html>`

---

## Bootstrap Radio Buttons

Radio buttons also facilitates you to select any number of options from a list of present options.

**See this example:**

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
- 8.
9. `</head>`

10. `<body>`
11. `<div class="container">`
12. `<h2>Input type: Radio buttons</h2>`
- 13.
14. `<form role="form">`
15. `<div class="radio">`
16. `<label><input type="radio" name="optradio">Option 1</label>`
17. `</div>`
18. `<div class="radio">`
19. `<label><input type="radio" name="optradio">Option 2</label>`
20. `</div>`
21. `</form>`
22. `</div>`
23. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
24. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
25. `</body>`
26. `</html>`

Use the **.radio-inline class** if you want the radio buttons to appear on the same line:

**See this example:**

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
- 8.
9. `</head>`
10. `<body>`
- 11.
12. `<div class="container">`
13. `<h2>Inline radio buttons</h2>`
14. `<p>The following radio boxes appear in the same line:</p>`
15. `<form role="form">`
16. `<label class="radio-inline">`
17. `<input type="radio" name="optradio">Option 1`

```
18. </label>
19. <label class="radio-inline">
20.   <input type="radio" name="optradio">Option 2
21. </label>
22. </form>
23. </div>
24. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
25. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
26. </body>
27. </html>
```

## Bootstrap 4 Inputs

Bootstrap 4 supports following types of form controls:

- input
- textarea
- checkbox
- radio
- select

## Bootstrap Input

Bootstrap supports all the HTML5 input types: text, password, datetime, datetime-local, date, month, time, week, number, email, url, search, tel, and color.

*Note: Inputs will NOT be fully styled if their type is not properly declared!*

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
```



7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
13. `<div class="container">`
14. `<h2>Form control: input example</h2>`
15. `<form>`
16. `<div class="form-group">`
17. `<label for="usr">Name:</label>`
18. `<input type="text" class="form-control" id="usr">`
19. `</div>`
20. `<div class="form-group">`
21. `<label for="pwd">Password:</label>`
22. `<input type="password" class="form-control" id="pwd">`
23. `</div>`
24. `</form>`
25. `</div>`
26. `</body>`
27. `</html>`

## Bootstrap Textarea

Use textarea with "form-control" to create a Bootstrap textarea.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`

9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
13. `<div class="container">`
14. `<h2>Form control: textarea</h2>`
15. `<p>The form below contains a textarea for comments:</p>`
16. `<form>`
17. `<div class="form-group">`
18. `<label for="comment">Comment:</label>`
19. `<textarea class="form-control" rows="5" id="comment"></textarea>`
20. `</div>`
21. `</form>`
22. `</div>`
23. `</body>`
24. `</html>`

## Bootstrap Select List

Bootstrap Select List is used to select single or multiple list. Select list is used if you want to allow the user to pick from multiple options.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`

```
12. <body>
13. <div class="container">
14. <h2>Form control: select</h2>
15. <p>The form below contains two dropdown menus (select lists):</p>
16. <form>
17.   <div class="form-group">
18.     <label for="sel1">Select list (select one):</label>
19.     <select class="form-control" id="sel1">
20.       <option>1</option>
21.       <option>2</option>
22.       <option>3</option>
23.       <option>4</option>
24.     </select>
25.     <br>
26.     <label for="sel2">Mutiple select list (hold shift to select more than one):</label>
27.     <select multiple class="form-control" id="sel2">
28.       <option>1</option>
29.       <option>2</option>
30.       <option>3</option>
31.       <option>4</option>
32.       <option>5</option>
33.     </select>
34.   </div>
35. </form>
36. </div>
37. </body>
38. </html>
```

## Bootstrap Tables

We can create different types of Bootstrap tables by using different classes to style them.

---

### Bootstrap Basic Table:

The basic Bootstrap table has a light padding and only horizontal dividers. The **.table** class is used to add basic styling to a table.

#### Example:

```
1. <!DOCTYPE html>
```

```

2. <html lang="en">
3. <head>
4. <title>Job</title>
5. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
  trap.min.css"/>
6. </head>
7. <body>
8.
9. <div class="container">
10. <h1>Basic Table Example</h1>
11.
12. <table class="table">
13. <tr><th>Id</th><th>Name</th><th>Age</th></tr>
14. <tr><td>101</td><td>Rahul</td><td>23</td></tr>
15. <tr><td>102</td><td>Umesh</td><td>22</td></tr>
16. <tr><td>103</td><td>Max</td><td>29</td></tr>
17. <tr><td>104</td><td>Ajeet</td><td>21</td></tr>
18. </table>
19.
20. </div>
21.
22. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
23. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scri
  pt>
24. </body>
25. </html>

```

---

## Bootstrap Striped Rows Table:

The **.table-striped class** is used to add zebra-stripes to a table:

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Job</title>
5. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
  trap.min.css"/>
6. </head>
7. <body>
8.

```

```
9. <div class="container">
10. <h1>Striped Table Example</h1>
11.
12. <table class="table table-striped">
13. <tr><th>Id</th><th>Name</th><th>Age</th></tr>
14. <tr><td>101</td><td>Rahul</td><td>23</td></tr>
15. <tr><td>102</td><td>Umesh</td><td>22</td></tr>
16. <tr><td>103</td><td>Max</td><td>29</td></tr>
17. <tr><td>104</td><td>Ajeet</td><td>21</td></tr>
18. </table>
19.
20. </div>
21.
22. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
23. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
24. </body>
25. </html>
```

---

## Bootstrap Bordered table:

The **.table-bordered class** is used to add borders on all sides of the table and cells:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Job</title>
5. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>
6. </head>
7. <body>
8.
9. <div class="container">
10. <h1>Bordered Table Example</h1>
11.
12. <table class="table table-striped table-bordered">
13. <tr><th>Id</th><th>Name</th><th>Age</th></tr>
14. <tr><td>101</td><td>Rahul</td><td>23</td></tr>
15. <tr><td>102</td><td>Umesh</td><td>22</td></tr>
16. <tr><td>103</td><td>Max</td><td>29</td></tr>
```

```

17. <tr><td>104</td><td>Ajeet</td><td>21</td></tr>
18. </table>
19.
20. </div>
21.
22. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
23. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scri
    pt>
24. </body>
25. </html>

```

---

## Bootstrap Hover rows Table:

The **.table-hover class** is used to enable a hover state on table rows:

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Job</title>
5.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
    trap.min.css"/>
6. </head>
7. <body>
8.
9. <div class="container">
10.  <h1>Hower rows Table Example</h1>
11.
12. <table class="table table-hover">
13.  <tr><th>Id</th><th>Name</th><th>Age</th></tr>
14.  <tr><td>101</td><td>Rahul</td><td>23</td></tr>
15.  <tr><td>102</td><td>Umesh</td><td>22</td></tr>
16.  <tr><td>103</td><td>Max</td><td>29</td></tr>
17.  <tr><td>104</td><td>Ajeet</td><td>21</td></tr>
18. </table>
19.
20. </div>
21.
22. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
23. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scri
    pt>

```

24. `</body>`

25. `</html>`

---

## Bootstrap Condensed table:

The **.table-condensed class** is used to make a table more compact by cutting cell padding in half:

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <title>Job</title>
5.     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boot
trap.min.css"/>
6.   </head>
7.   <body>
8.
9.     <div class="container">
10.    <h1>Condensed Table Example</h1>
11.
12.    <table class="table table-condensed">
13.      <tr><th>Id</th><th>Name</th><th>Age</th></tr>
14.      <tr><td>101</td><td>Rahul</td><td>23</td></tr>
15.      <tr><td>102</td><td>Umesh</td><td>22</td></tr>
16.      <tr><td>103</td><td>Max</td><td>29</td></tr>
17.      <tr><td>104</td><td>Ajeet</td><td>21</td></tr>
18.    </table>
19.
20.  </div>
21.
22.
23.  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
24.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scri
pt>
25. </body>
26. </html>
```

---

## Bootstrap Contextual classes:.

Contextual classes are used to color table rows (<tr>) or table cells (<td>):

Following are the different contextual classes:

Class	Description
.active	It is used to apply the hover color to the table row or table cell
.success	It indicates a successful or positive action
.info	It indicates a neutral informative change or action
.warning	It specifies a warning that might need attention
.danger	It indicates a dangerous or potentially negative action

**Example:**

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <title>Job</title>
5.     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>
6.   </head>
7.   <body>
8.
9.     <div class="container">
10.    <h1>Contextual classes</h1>
11.
12.    <table class="table">
13.      <tr class="success"><th>Id</th><th>Name</th><th>Age</th></tr>
14.      <tr class="active"><td>101</td><td>Rahul</td><td>23</td></tr>
15.      <tr class="danger"><td>102</td><td>Umesh</td><td>22</td></tr>
16.      <tr class="info"><td>103</td><td>Max</td><td>29</td></tr>
17.      <tr class="warning"><td>104</td><td>Ajeet</td><td>21</td></tr>
18.    </table>
19.
20.  </div>
21.
22. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
```



23. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>`
  24. `</body>`
  25. `</html>`
- 

## Responsive tables:

The **.table-responsive class** is used to create a responsive table. You can open the responsive table even on small devices (under 768px). Then the table will be scrolled horizontally. Displays larger than 768px wide, there is no difference.

### See this example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Job</title>`
5. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>`
6. `</head>`
7. `<body>`
- 8.
9. `<div class="container">`
10. `<h1>Contextual classes</h1>`
11. `<div class="table-responsive">`
- 12.
13. `<table class="table">`
14. `<tr class="success"><th>Id</th><th>Name</th><th>Age</th></tr>`
15. `<tr class="active"><td>101</td><td>Rahul</td><td>23</td></tr>`
16. `<tr class="danger"><td>102</td><td>Umesh</td><td>22</td></tr>`
17. `<tr class="info"><td>103</td><td>Max</td><td>29</td></tr>`
18. `<tr class="warning"><td>104</td><td>Ajeet</td><td>21</td></tr>`
19. `</table>`
- 20.
21. `</div>`
- 22.
23. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>`
24. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>`
25. `</body>`
26. `</html>`

---

Some newly added tables in Bootstrap 4:

## Black/Dark Table

The .table-dark class is used to add a black background to the table:

### Example

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="container">`
15. `<h2>Black/Dark Table</h2>`
16. `<p>The .table-dark class is used to add a black background to the table:</p>`
17. `<table class="table table-dark">`
18. `<thead>`
19. `<tr>`
20. `<th>Firstname</th>`
21. `<th>Lastname</th>`
22. `<th>Email</th>`
23. `</tr>`
24. `</thead>`
25. `<tbody>`
26. `<tr>`
27. `<td>Ajeet</td>`
28. `<td>Kumar</td>`
29. `<td>ajeet@jtp.com</td>`
30. `</tr>`

```
31. <tr>
32.   <td>Mahesh</td>
33.   <td>Sharma</td>
34.   <td>mahesh@jtp.com</td>
35. </tr>
36. <tr>
37.   <td>Sonoo</td>
38.   <td>Jaiswal</td>
39.   <td>sonoo@jtp.com</td>
40. </tr>
41. </tbody>
42. </table>
43. </div>
44.
45. </body>
46. </html>
```

---

## Dark Striped Table

Combine the .table-dark class and .table-striped class to create a dark, striped table:

### Example

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
13.
14. <div class="container">
15.   <h2>Dark Striped Table</h2>
```

```
16. <p>Combine the .table-dark class and .table-  
    striped class to create a dark, striped table: </p>  
17. <table class="table table-dark table-striped">  
18.     <thead>  
19.     <tr>  
20.         <th>Firstname</th>  
21.         <th>Lastname</th>  
22.         <th>Email</th>  
23.     </tr>  
24. </thead>  
25. <tbody>  
26.     <tr>  
27.         <td>Ajeet</td>  
28.         <td>Kumar</td>  
29.         <td>ajeet@jtp.com</td>  
30.     </tr>  
31.     <tr>  
32.         <td>Mahesh</td>  
33.         <td>Sharma</td>  
34.         <td>mahesh@jtp.com</td>  
35.     </tr>  
36.     <tr>  
37.         <td>Sonoo</td>  
38.         <td>Jaiswal</td>  
39.         <td>sonoo@jtp.com</td>  
40.     </tr>  
41. </tbody>  
42. </table>  
43. </div>  
44.  
45. </body>  
46. </html>
```

---

## Hoverable Dark Table

The .table-hover class is used to add a hover effect (grey background color) on table rows:

### Example

```
1. <!DOCTYPE html>
```

```
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
13.
14. <div class="container">
15. <h2>Hoverable Dark Table</h2>
16. <p>The .table-
  hover class is used to add a hover effect (grey background color) on table rows:</p>
```

```
17. <table class="table table-dark table-hover">
18. <thead>
19. <tr>
20. <th>Firstname</th>
21. <th>Lastname</th>
22. <th>Email</th>
23. </tr>
24. </thead>
25. <tbody>
26. <tr>
27. <td>Ajeet</td>
28. <td>Kumar</td>
29. <td>ajeet@jtp.com</td>
30. </tr>
31. <tr>
32. <td>Mahesh</td>
33. <td>Sharma</td>
34. <td>mahesh@jtp.com</td>
35. </tr>
36. <tr>
```

```
37.     <td>Sonoo</td>
38.     <td>Jaiswal</td>
39.     <td>sonoo@jtp.com</td>
40. </tr>
41. </tbody>
42. </table>
43. </div>
44.
45. </body>
46. </html>
```

---

## Bootstrap 4 Contextual Table

Contextual classes can be used to color the whole table (<table>), the table rows (<tr>) or table cells (<td>).

The classes that can be used are:

.table-primary, .table-success, .table-info, .table-warning, .table-danger, .table-active, .table-secondary, .table-light and .table-dark:

Let's take an example to see the usage of all contextual classes in a Bootstrap 4 table.

### Example

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
13.
14. <div class="container">
15.   <h2>Contextual Classes Example</h2>
```

```
16. <table class="table">
17.   <thead>
18.     <tr>
19.       <th>Firstname</th>
20.       <th>Lastname</th>
21.       <th>Email</th>
22.     </tr>
23.   </thead>
24.   <tbody>
25.     <tr>
26.       <td>Default</td>
27.       <td>A</td>
28.       <td>a@jtp.com</td>
29.     </tr>
30.     <tr class="table-primary">
31.       <td>Primary</td>
32.       <td>B</td>
33.       <td>b@jtp.com</td>
34.     </tr>
35.     <tr class="table-success">
36.       <td>Success</td>
37.       <td>C</td>
38.       <td>c@jtp.com</td>
39.     </tr>
40.     <tr class="table-danger">
41.       <td>Danger</td>
42.       <td>D</td>
43.       <td>d@jtp.com</td>
44.     </tr>
45.     <tr class="table-info">
46.       <td>Info</td>
47.       <td>E</td>
48.       <td>e@jtp.com</td>
49.     </tr>
50.     <tr class="table-warning">
51.       <td>Warning</td>
52.       <td>F</td>
53.       <td>f@jtp.com</td>
54.     </tr>
55.     <tr class="table-active">
```

```
56.     <td>Active</td>
57.     <td>G</td>
58.     <td>g@jtp.com</td>
59. </tr>
60. <tr class="table-secondary">
61.     <td>Secondary</td>
62.     <td>H</td>
63.     <td>h@jtp.com</td>
64. </tr>
65. <tr class="table-light">
66.     <td>Light</td>
67.     <td>I</td>
68.     <td>j@jtp.com</td>
69. </tr>
70. <tr class="table-dark text-dark">
71.     <td>Dark</td>
72.     <td>K</td>
73.     <td>k@jtp.com</td>
74. </tr>
75. </tbody>
76. </table>
77. </div>
78.
79. </body>
80. </html>
```

---

## Table Head Colors

You can change the background color of the table header by using `.thead-dark` class to add a black background to table headers, and the `.thead-light` class to add a grey background to table headers.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <title>Bootstrap Example</title>
5.     <meta charset="utf-8">
6.     <meta name="viewport" content="width=device-width, initial-scale=1">
```



```
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-  
beta.2/css/bootstrap.min.css">  
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  
9. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><  
/script>  
10. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-  
beta.2/js/bootstrap.min.js"></script>  
11. </head>  
12. <body>  
13.  
14. <div class="container">  
15. <h2>Table Head Colors</h2>  
16. <table class="table">  
17. <thead class="thead-dark">  
18. <tr>  
19. <th>Firstname</th>  
20. <th>Lastname</th>  
21. <th>Email</th>  
22. </tr>  
23. </thead>  
24. <tbody>  
25. <tr>  
26. <th>Firstname</th>  
27. <th>Lastname</th>  
28. <th>Email</th>  
29. </tr>  
30. </thead>  
31. <tbody>  
32. <tr>  
33. <td>Ajeet</td>  
34. <td>Kumar</td>  
35. <td>ajeet@jtp.com</td>  
36. </tr>  
37. <tr>  
38. <td>Mahesh</td>  
39. <td>Sharma</td>  
40. <td>mahesh@jtp.com</td>  
41. </tr>  
42. <tr>  
43. <td>Sonoo</td>
```

```
44.     <td>Jaiswal</td>
45.     <td>sonoo@jtp.com</td>
46. </tr>
47. </tbody>
48. </table>
49. <table class="table">
50.   <thead class="thead-light">
51.     <tr>
52.       <th>Firstname</th>
53.       <th>Lastname</th>
54.       <th>Email</th>
55.     </tr>
56.   </thead>
57.   <tbody>
58.     <tr>
59.       <th>Firstname</th>
60.       <th>Lastname</th>
61.       <th>Email</th>
62.     </tr>
63.   </thead>
64.   <tbody>
65.     <tr>
66.       <td>Ajeet</td>
67.       <td>Kumar</td>
68.       <td>ajeet@jtp.com</td>
69.     </tr>
70.     <tr>
71.       <td>Mahesh</td>
72.       <td>Sharma</td>
73.       <td>mahesh@jtp.com</td>
74.     </tr>
75.     <tr>
76.       <td>Sonoo</td>
77.       <td>Jaiswal</td>
78.       <td>sonoo@jtp.com</td>
79.     </tr>
80.
81.   </tbody>
82. </table>
83. </div>
```

- 84.
85. `</body>`
86. `</html>`

---

## Small Table

The `.table-sm` class is used to make the table smaller by cutting cell padding in half.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="container">`
15. `<h2>Small Table Example</h2>`
16. `<p>`The `.table-sm` class is used to make the table smaller by cutting cell padding in half.`</p>`
17. `<table class="table table-bordered table-sm">`
18. `<thead>`
19. `<tr>`
20. `<th>Firstname</th>`
21. `<th>Lastname</th>`
22. `<th>Email</th>`
23. `</tr>`
24. `</thead>`
25. `<tbody>`
26. `<tr>`

```
27.     <td>Ajeet</td>
28.     <td>Kumar</td>
29.     <td>ajeet@jtp.com</td>
30. </tr>
31. <tr>
32.     <td>Mahesh</td>
33.     <td>Sharma</td>
34.     <td>mahesh@jtp.com</td>
35. </tr>
36. <tr>
37.     <td>Sonoo</td>
38.     <td>Jaiswal</td>
39.     <td>sonoo@jtp.com</td>
40. </tr>
41.
42. </tbody>
43. </table>
44. </div>
45.
46. </body>
47. </html>
```

## Bootstrap Dropdowns

Dropdown menus are toggleable, contextual menus, used for displaying links in a list format. It facilitates users to choose one value from a predefined list. This can be made interactive with the dropdown JavaScript plugin.

You have to wrap dropdown menu within the class `.dropdown` to create Bootstrap Dropdown.

## Bootstrap Dropdown Example

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta name="viewport" content="width=device-width, initial-scale=1">
5.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
ap.min.css">
6.
7. </head>
8. <body>
```

- 9.
10. `<div class="container">`
11. `<h2>Dropdowns</h2>`
12. `<p>The .dropdown class is used to indicate a dropdown menu.</p>`
13. `<p>Use the .dropdown-menu class to actually build the dropdown menu.</p>`
14. `<p>To open the dropdown menu, use a button or a link with a class of .dropdown-toggle and data-toggle="dropdown".</p>`
15. `<div class="dropdown">`
16. `<button class="btn btn-primary dropdown-toggle" type="button" data-toggle="dropdown">Dropdown Example`
17. `<span class="caret"></span></button>`
18. `<ul class="dropdown-menu">`
19. `<li><a href="#">HTML</a></li>`
20. `<li><a href="#">CSS</a></li>`
21. `<li><a href="#">JavaScript</a></li>`
22. `</ul>`
23. `</div>`
24. `</div>`
- 25.
26. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
27. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
- 28.
29. `</body>`
30. `</html>`

## Bootstrap Dropdown Divider

The **class .divider** is used to separate links inside the dropdown menu:

**See this example:**

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<meta name="viewport" content="width=device-width, initial-scale=1">`
5. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
- 6.
7. `</head>`
8. `<body>`

- 9.
10. `<div class="container">`
11. `<h2>Dropdowns</h2>`
12. `<p>The .divider class is used to separate links inside the dropdown menu:</p>`
13. `<div class="dropdown">`
14. `<button class="btn btn-default dropdown-toggle" type="button" data-`  
`toggle="dropdown">Tutorials`
15. `<span class="caret"></span></button>`
16. `<ul class="dropdown-menu">`
17. `<li><a href="#">HTML</a></li>`
18. `<li><a href="#">CSS</a></li>`
19. `<li><a href="#">JavaScript</a></li>`
20. `<li class="divider"></li>`
21. `<li><a href="#">About Us</a></li>`
22. `</ul>`
23. `</div>`
24. `</div>`
- 25.
26. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
27. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scri`  
`pt>`
- 28.
29. `</body>`
30. `</html>`

## Bootstrap Dropdown Header

The **class .dropdown-header** is used to add headers inside the dropdown menu.

**See this example:**

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<meta name="viewport" content="width=device-width, initial-scale=1">`
5. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr`  
`ap.min.css">`
- 6.
7. `</head>`
8. `<body>`
- 9.

```

10. <div class="container">
11. <h2>Dropdowns</h2>
12. <p>The .dropdown-header class is used to add headers inside the dropdown menu:</p>
13. <div class="dropdown">
14.   <button class="btn btn-default dropdown-toggle" type="button" data-
toggle="dropdown">Tutorials
15.   <span class="caret"></span></button>
16.   <ul class="dropdown-menu">
17.     <li class="dropdown-header">Dropdown header 1</li>
18.     <li><a href="#">HTML</a></li>
19.     <li><a href="#">CSS</a></li>
20.     <li><a href="#">JavaScript</a></li>
21.     <li class="divider"></li>
22.     <li class="dropdown-header">Dropdown header 2</li>
23.     <li><a href="#">About Us</a></li>
24.   </ul>
25. </div>
26. </div>
27.
28. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
29. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scri
pt>
30.
31. </body>
32. </html>

```

## Bootstrap Dropdown: Disable an item

Use the **class .disabled** to disable an item in the dropdown menu.

**See this example:**

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta name="viewport" content="width=device-width, initial-scale=1">
5.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
ap.min.css">
6.
7. </head>
8. <body>

```

- 9.
10. `<div class="container">`
11. `<h2>Dropdowns</h2>`
12. `<p>Here, CSS is disable.</p>`
13. `<div class="dropdown">`
14. `<button class="btn btn-default dropdown-toggle" type="button" data-`  
`toggle="dropdown">Tutorials`
15. `<span class="caret"></span></button>`
16. `<ul class="dropdown-menu">`
17. `<li><a href="#">HTML</a></li>`
18. `<li class="disabled"><a href="#">CSS</a></li>`
19. `<li><a href="#">JavaScript</a></li>`
20. `<li class="divider"></li>`
21. `<li><a href="#">About Us</a></li>`
22. `</ul>`
23. `</div>`
24. `</div>`
- 25.
26. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
27. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scrip`  
`t>`
- 28.
29. `</body>`
30. `</html>`

---

## Bootstrap 4 Disable and Active Items

To make an item active, highlight a specific dropdown item with the `.active` class (active item is appeared with a blue background color).

To disable an item in the dropdown menu, you have to use the `.disabled` class (It makes a light-grey text color and a "no-parking-sign" icon on hover).

### Example:

1. `<!DOCTYPE html>`
2. `<html>`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta name="viewport" content="width=device-width, initial-scale=1">`
6. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-`  
`beta.2/css/bootstrap.min.css">`



```
7. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
8. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
9. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
10. </head>
11. <body>
12.
13. <div class="container">
14. <h2>Dropdowns Active and Disable Example</h2>
15. <div class="dropdown">
16. <button type="button" class="btn btn-primary dropdown-toggle" data-
  toggle="dropdown">
17.   Dropdown button
18. </button>
19. <div class="dropdown-menu">
20. <a class="dropdown-item" href="#">Normal</a>
21. <a class="dropdown-item active" href="#">Active</a>
22. <a class="dropdown-item disabled" href="#">Disabled</a>
23. </div>
24. </div>
25. </div>
26.
27. </body>
28. </html>
```

---

## Dropdown Position

It is used to position the dropdown items. By default dropdown is positioned at left. You can right-align the dropdown by adding the `.dropdown-menu-right` class to the element with `.dropdown-menu`.

### Example:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Bootstrap Example</title>
5. <meta name="viewport" content="width=device-width, initial-scale=1">
6. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
```

```
7. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
8. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
9. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
10. </head>
11. <body>
12.
13. <div class="container">
14. <h2>Right Aligned Dropdowns</h2>
15. <div class="dropdown">
16. <button type="button" class="btn btn-primary dropdown-toggle" data-
  toggle="dropdown">
17.   Dropdown button
18. </button>
19. <div class="dropdown-menu dropdown-menu-right">
20. <a class="dropdown-item" href="#">Link 1</a>
21. <a class="dropdown-item" href="#">Link 2</a>
22. <a class="dropdown-item" href="#">Link 3</a>
23. </div>
24. </div>
25. </div>
26.
27. </body>
28. </html>
```

---

## Bootstrap 4 Dropup

If you want the dropdown menu to expand upwards instead of downwards, change the

element with class="dropdown" to "dropup":

### Example:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Bootstrap Example</title>
5. <meta name="viewport" content="width=device-width, initial-scale=1">
6. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
```

```
7. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
8. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
9. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
10. </head>
11. <body>
12. <br>
13.
14. <div class="container">
15. <h2>Dropup Example</h2>
16. <p>The .dropup class makes the dropdown menu expand upwards instead of downwards:
  </p>
17. <div class="dropup">
18. <button type="button" class="btn btn-primary dropdown-toggle" data-
  toggle="dropdown">
19.   Dropup button
20. </button>
21. <div class="dropdown-menu">
22. <a class="dropdown-item" href="#">Link 1</a>
23. <a class="dropdown-item" href="#">Link 2</a>
24. </div>
25. </div>
26. </div>
27.
28. </body>
29. </html>
```

---

## Split Button Dropdowns

It is used to show dropdowns as split buttons. Here we use all contextual classes.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-  
beta.2/css/bootstrap.min.css">  
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  
9. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><  
/script>  
10. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-  
beta.2/js/bootstrap.min.js"></script>  
11. </head>  
12. <body>  
13.  
14. <div class="container">  
15. <h2>Dropdown Split Buttons Example</h2>  
16. <div class="btn-group">  
17. <button type="button" class="btn btn-primary">Primary</button>  
18. <button type="button" class="btn btn-primary dropdown-toggle dropdown-toggle-  
split" data-toggle="dropdown">  
19. <span class="caret"></span>  
20. </button>  
21. <div class="dropdown-menu">  
22. <a class="dropdown-item" href="#">Link 1</a>  
23. <a class="dropdown-item" href="#">Link 2</a>  
24. </div>  
25. </div>  
26.  
27. <div class="btn-group">  
28. <button type="button" class="btn btn-secondary">Secondary</button>  
29. <button type="button" class="btn btn-secondary dropdown-toggle dropdown-toggle-  
split" data-toggle="dropdown">  
30. <span class="caret"></span>  
31. </button>  
32. <div class="dropdown-menu">  
33. <a class="dropdown-item" href="#">Link 1</a>  
34. <a class="dropdown-item" href="#">Link 2</a>  
35. </div>  
36. </div>  
37.  
38. <div class="btn-group">  
39. <button type="button" class="btn btn-success">Success</button>  
40. <button type="button" class="btn btn-success dropdown-toggle dropdown-toggle-  
split" data-toggle="dropdown">
```

```
41.   <span class="caret"></span>
42. </button>
43. <div class="dropdown-menu">
44.   <a class="dropdown-item" href="#">Link 1</a>
45.   <a class="dropdown-item" href="#">Link 2</a>
46. </div>
47. </div>
48.
49. <div class="btn-group">
50.   <button type="button" class="btn btn-info">Info</button>
51.   <button type="button" class="btn btn-info dropdown-toggle dropdown-toggle-
split" data-toggle="dropdown">
52.     <span class="caret"></span>
53.   </button>
54.   <div class="dropdown-menu">
55.     <a class="dropdown-item" href="#">Link 1</a>
56.     <a class="dropdown-item" href="#">Link 2</a>
57.   </div>
58. </div>
59.
60. <div class="btn-group">
61.   <button type="button" class="btn btn-warning">Warning</button>
62.   <button type="button" class="btn btn-warning dropdown-toggle dropdown-toggle-
split" data-toggle="dropdown">
63.     <span class="caret"></span>
64.   </button>
65.   <div class="dropdown-menu">
66.     <a class="dropdown-item" href="#">Link 1</a>
67.     <a class="dropdown-item" href="#">Link 2</a>
68.   </div>
69. </div>
70.
71. <div class="btn-group">
72.   <button type="button" class="btn btn-danger">Danger</button>
73.   <button type="button" class="btn btn-danger dropdown-toggle dropdown-toggle-
split" data-toggle="dropdown">
74.     <span class="caret"></span>
75.   </button>
76.   <div class="dropdown-menu">
77.     <a class="dropdown-item" href="#">Link 1</a>
```

```
78. <a class="dropdown-item" href="#">Link 2</a>
79. </div>
80. </div>
81. </div>
82.
83. </body>
84. </html>
```

## Bootstrap Images

Bootstrap supports for images. There are three classes in Bootstrap that can be used to apply some simple style to the images.

**The following classes add style to the images:**

Classes	Uses
.img-rounded	It adds border-radius:6px to give the image rounded corners.
.img-circle	It makes the entire image round by adding border-radius:500px.
.img-thumbnail	It adds a bit of padding and a gray border.

## Bootstrap Image-rounded Example

The **class .img-rounded** is used to add rounded corners to an image ( IE8 does not support rounded corners).

**Example:**

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap image</title>
5.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
6. </head>
7. <body>
8.   <div class="container">
```

9. `<h2>Rounded Corners</h2>`
10. ``
11. `</div>`
- 12.
13. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
14. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
15. `</body>`
16. `</html>`

## Bootstrap Image-circle Example

The **class .img-circle** is used to shape the image to a circle (IE8 does not support rounded corners).

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap image</title>`
5. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
6. `</head>`
7. `<body>`
8. `<div class="container">`
9. `<h2>Circle</h2>`
10. ``
11. `</div>`
- 12.
13. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
14. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
15. `</body>`
16. `</html>`
- 17.
- 18.

## Bootstrap Thumbnail Image Example

The **class .img-thumbnail** is used to shape an image to a thumbnail.

### Example:

1. `<!DOCTYPE html>`
  2. `<html lang="en">`
  3. `<head>`
  4. `<title>Bootstrap image</title>`
  5. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
  6. `</head>`
  7. `<body>`
  8. `<div class="container">`
  9. `<h2>Thumbnail</h2>`
  10. ``
  11. `</div>`
  - 12.
  13. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
  14. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
  15. `</body>`
  16. `</html>`
- 

## Bootstrap Responsive images

The responsive images can adjust themselves automatically to fit the size of screen. You can create responsive images by adding an **.img-responsive class** to the `<img>` tag. The image will then scale nicely to the parent element.

The **.img-responsive class** applies `display: block;` and `max-width: 100%;` and `height: auto;` to the image.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
- 3.
4. `<head>`
5. `<title>Bootstrap Example</title>`
6. `<meta charset="utf-8">`
7. `<meta name="viewport" content="width=device-width, initial-scale=1">`
8. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`



```
9. </head>
10. <body>
11.
12. <div class="container">
13.   <h2>Responsive Image</h2>
14.   
15. </div>
16. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
17. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
18.
19. </body>
20. </html>
21.
22.
```

---

## Bootstrap Responsive Videos / Embeds

In Bootstrap, you can also add videos and scale them properly on any devices. The **class .embed-responsive-item** is used to create a responsive video. Class can be applied directly to `<iframe>`, `<embed>`, `<video>`, and `<object>` elements.

### Let's take an example:

In the following example, we add **.embed-responsive-item class** to an `<iframe>` tag to make the video responsive. It can scale the video nicely according to the parent element.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
8. </head>
9. <body>
10.
11. <div class="container">
```

12. `<h2>Responsive Embed</h2>`
13. `<div class="embed-responsive embed-responsive-16by9">`
14. `<iframe class="embed-responsive-`  
`item" src="https://www.youtube.com/embed/XGSy3_Czz8k"></iframe>`
15. `</div>`
16. `</div>`
17. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
18. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scri`  
`pt>`
- 19.
20. `</body>`
21. `</html>`

---

## Bootstrap4 Images

Aligning images are used to float an image to the right with the .float-right class or to the left with .float-left.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-`  
`beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><`  
`/script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-`  
`beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="container">`
15. `<h2>Aligning images Example</h2>`
16. `<p>Use the float classes to float the image to the left or to the right:</p>`
17. ``

18. ``
19. `</div>`
- 20.
21. `</body>`
22. `</html>`

## Bootstrap Navigation Bar

A navigation bar is like a navigation header that is placed at the top of the page. You can collapse or extend it according to the screen size.

You can create a standard navigation bar at the top of the page with with `<nav class="navbar navbar-default">`.

**See this example:**

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Case</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
- 8.
9. `</head>`
10. `<body>`
- 11.
12. `<nav class="navbar navbar-default">`
13. `<div class="container-fluid">`
14. `<div class="navbar-header">`
15. `<a class="navbar-brand" href="#">WebSiteName</a>`
16. `</div>`
17. `<ul class="nav navbar-nav">`
18. `<li class="active"><a href="#">Home</a></li>`
19. `<li><a href="#">Page 1</a></li>`
20. `<li><a href="#">Page 2</a></li>`
21. `<li><a href="#">Page 3</a></li>`
22. `</ul>`
23. `</div>`
24. `</nav>`
- 25.

26. `<div class="container">`
  27. `<h3>Basic Navbar Example</h3>`
  28. `<p>A navigation bar is a navigation header that is placed at the top of the page.</p>`
  29. `</div>`
  30. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
  31. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
  32. `</body>`
  33. `</html>`
- 

## Bootstrap Inverted Navigation Bar

Inverted navigation bar provides an alternative black navbar. It can be used to style the default navigation bar by changing **.navbar-default class** into **.navbar-inverse class**.

### See this example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Case</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
- 8.
9. `</head>`
10. `<body>`
- 11.
12. `<nav class="navbar navbar-inverse">`
13. `<div class="container-fluid">`
14. `<div class="navbar-header">`
15. `<a class="navbar-brand" href="#">WebSiteName</a>`
16. `</div>`
17. `<ul class="nav navbar-nav">`
18. `<li class="active"><a href="#">Home</a></li>`

```
19.   <li><a href="#">Page 1</a></li>
20.   <li><a href="#">Page 2</a></li>
21.   <li><a href="#">Page 3</a></li>
22.   </ul>
23. </div>
24. </nav>
25.
26. <div class="container">
27.   <h3>Inverted Navbar</h3>
28.   <p>An inverted navbar is black instead of gray.</p>
29. </div>
30. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
31. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
32. </body>
33. </html>
```

---

## Bootstrap Navigation Bar with Dropdown

You can also add dropdown menu with navigation bars. The following example adds a dropdown menu for the "page 1" button.

**See this example:**

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Case</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
  ap.min.css">
8.
9. </head>
10. <body>
11.
12. <nav class="navbar navbar-inverse">
13.   <div class="container-fluid">
14.     <div class="navbar-header">
15.       <a class="navbar-brand" href="#">WebSiteName</a>
16.     </div>
```

```

17. <ul class="nav navbar-nav">
18.   <li class="active"><a href="#">Home</a></li>
19.   <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="#">Page 1 <span class="caret"></span></a>
20.     <ul class="dropdown-menu">
21.       <li><a href="#">Page 1-1</a></li>
22.       <li><a href="#">Page 1-2</a></li>
23.       <li><a href="#">Page 1-3</a></li>
24.     </ul>
25.   </li>
26.   <li><a href="#">Page 2</a></li>
27.   <li><a href="#">Page 3</a></li>
28. </ul>
29. </div>
30. </nav>
31.
32. <div class="container">
33.   <h3>Navbar With Dropdown</h3>
34.   <p>This example adds a dropdown menu for the "Page 1" button in the navigation bar.</
p>
35. </div>
36. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
37. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scrip
t>
38. </body>
39. </html>

```

---

## Bootstrap Right-Aligned Navigation Bar

Use the **.navbar-right class** to right-align navigation bar buttons. In the following example, we add "Sign Up" and "Log in" button to the right in the navigation bar.

**See this example:**

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Case</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">

```

```
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
   ap.min.css">
8.
9. </head>
10. <body>
11.
12. <nav class="navbar navbar-inverse">
13.   <div class="container-fluid">
14.     <div class="navbar-header">
15.       <a class="navbar-brand" href="#">WebSiteName</a>
16.     </div>
17.     <ul class="nav navbar-nav">
18.       <li class="active"><a href="#">Home</a></li>
19.       <li class="dropdown"><a class="dropdown-toggle" data-
   toggle="dropdown" href="#">Page 1 <span class="caret"></span></a>
20.         <ul class="dropdown-menu">
21.           <li><a href="#">Page 1-1</a></li>
22.           <li><a href="#">Page 1-2</a></li>
23.           <li><a href="#">Page 1-3</a></li>
24.         </ul>
25.       </li>
26.       <li><a href="#">Page 2</a></li>
27.       <li><a href="#">Page 3</a></li>
28.     </ul>
29.     <ul class="nav navbar-nav navbar-right">
30.       <li><a href="#"><span class="glyphicon glyphicon-
   user"></span> Sign Up</a></li>
31.       <li><a href="#"><span class="glyphicon glyphicon-log-
   in"></span> Login</a></li>
32.     </ul>
33.   </div>
34. </nav>
35.
36. <div class="container">
37.   <h3>Right Aligned Navbar</h3>
38.   <p>The .navbar-right class is used to right-align navigation bar buttons.</p>
39. </div>
40. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
41. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scrip
   t>
```

42. `</body>`

43. `</html>`

---

## Basic Navbar

In Bootstrap 4, you can extend or collapse a navigation bar depending on the screen size. A standard navigation bar is created with the `.navbar` class, followed by a responsive collapsing class: `.navbar-expand-xl|lg|md|sm` (stacks the navbar vertically on extra-large, large, medium or small screens).

To add links inside the navbar, use a `<ul>` element with `class="navbar-nav"`. Then add `<li>` elements with a `.nav-item` class followed by an `<a>` element with a `.nav-link` class.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="container-fluid">`
15. `<h3>Basic Navbar Example</h3>`
16. `<p>A navigation bar is a navigation header that is placed at the top of the page.</p>`
- 17.
18. `</div>`
19. `<nav class="navbar navbar-expand-sm bg-light">`
20. `<ul class="navbar-nav">`
21. `<li class="nav-item">`
22. `<a class="nav-link" href="#">Link 1</a>`
23. `</li>`



```
24. <li class="nav-item">
25.   <a class="nav-link" href="#">Link 2</a>
26. </li>
27. <li class="nav-item">
28.   <a class="nav-link" href="#">Link 3</a>
29. </li>
30. </ul>
31. </nav>
32. <br>
33. </body>
34. </html>
```

---

## Vertical Navbar

If you want to create a vertical navigation bar, remove the .navbar-expand-  
xl|lg|md|sm class.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
13.
14. <div class="container-fluid">
15.   <h3>Vertical Navbar Example</h3>
16.   <p>A navigation bar is a navigation header that is placed at the top of the page.</p>
17. </div>
18.
19. <nav class="navbar bg-light">
```

```
20. <ul class="navbar-nav">
21.   <li class="nav-item">
22.     <a class="nav-link" href="#">Link 1</a>
23.   </li>
24.   <li class="nav-item">
25.     <a class="nav-link" href="#">Link 2</a>
26.   </li>
27.   <li class="nav-item">
28.     <a class="nav-link" href="#">Link 3</a>
29.   </li>
30. </ul>
31. </nav>
32. <br>
33.
34. </body>
35. </html>
```

---

## Colored Navbar

You can use any of the .bg-color classes to change the background color of the navbar (.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark and .bg-light).

*Note: Add a white text color to all links in the navbar with the .navbar-dark class, or use the .navbar-light class to add a black text color.*

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
```

```
12. <body>
13. <div class="container">
14. <h3>Colored Navbar Example</h3>
15.
16. </div>
17. <nav class="navbar navbar-expand-sm bg-light navbar-light">
18. <ul class="navbar-nav">
19. <li class="nav-item active">
20. <a class="nav-link" href="#">Active</a>
21. </li>
22. <li class="nav-item">
23. <a class="nav-link" href="#">Link</a>
24. </li>
25. <li class="nav-item">
26. <a class="nav-link" href="#">Link</a>
27. </li>
28. <li class="nav-item">
29. <a class="nav-link disabled" href="#">Disabled</a>
30. </li>
31. </ul>
32. </nav>
33. <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
34. <ul class="navbar-nav">
35. <li class="nav-item active">
36. <a class="nav-link" href="#">Active</a>
37. </li>
38. <li class="nav-item">
39. <a class="nav-link" href="#">Link</a>
40. </li>
41. <li class="nav-item">
42. <a class="nav-link" href="#">Link</a>
43. </li>
44. <li class="nav-item">
45. <a class="nav-link disabled" href="#">Disabled</a>
46. </li>
47. </ul>
48. </nav>
49. <nav class="navbar navbar-expand-sm bg-primary navbar-dark">
50. <ul class="navbar-nav">
51. <li class="nav-item active">
```

```
52.   <a class="nav-link" href="#">Active</a>
53. </li>
54. <li class="nav-item">
55.   <a class="nav-link" href="#">Link</a>
56. </li>
57. <li class="nav-item">
58.   <a class="nav-link" href="#">Link</a>
59. </li>
60. <li class="nav-item">
61.   <a class="nav-link disabled" href="#">Disabled</a>
62. </li>
63. </ul>
64. </nav>
65. <nav class="navbar navbar-expand-sm bg-success navbar-dark">
66. <ul class="navbar-nav">
67. <li class="nav-item active">
68.   <a class="nav-link" href="#">Active</a>
69. </li>
70. <li class="nav-item">
71.   <a class="nav-link" href="#">Link</a>
72. </li>
73. <li class="nav-item">
74.   <a class="nav-link" href="#">Link</a>
75. </li>
76. <li class="nav-item">
77.   <a class="nav-link disabled" href="#">Disabled</a>
78. </li>
79. </ul>
80. </nav>
81. <nav class="navbar navbar-expand-sm bg-info navbar-dark">
82. <ul class="navbar-nav">
83. <li class="nav-item active">
84.   <a class="nav-link" href="#">Active</a>
85. </li>
86. <li class="nav-item">
87.   <a class="nav-link" href="#">Link</a>
88. </li>
89. <li class="nav-item">
90.   <a class="nav-link" href="#">Link</a>
91. </li>
```

```
92. <li class="nav-item">
93.   <a class="nav-link disabled" href="#">Disabled</a>
94. </li>
95. </ul>
96. </nav>
97. <nav class="navbar navbar-expand-sm bg-warning navbar-dark">
98.   <ul class="navbar-nav">
99.     <li class="nav-item active">
100.       <a class="nav-link" href="#">Active</a>
101.     </li>
102.     <li class="nav-item">
103.       <a class="nav-link" href="#">Link</a>
104.     </li>
105.     <li class="nav-item">
106.       <a class="nav-link" href="#">Link</a>
107.     </li>
108.     <li class="nav-item">
109.       <a class="nav-link disabled" href="#">Disabled</a>
110.     </li>
111.   </ul>
112. </nav>
113. <nav class="navbar navbar-expand-sm bg-danger navbar-dark">
114.   <ul class="navbar-nav">
115.     <li class="nav-item active">
116.       <a class="nav-link" href="#">Active</a>
117.     </li>
118.     <li class="nav-item">
119.       <a class="nav-link" href="#">Link</a>
120.     </li>
121.     <li class="nav-item">
122.       <a class="nav-link" href="#">Link</a>
123.     </li>
124.     <li class="nav-item">
125.       <a class="nav-link disabled" href="#">Disabled</a>
126.     </li>
127.   </ul>
128. </nav>
129. <nav class="navbar navbar-expand-sm bg-secondary navbar-dark">
130.   <ul class="navbar-nav">
131.     <li class="nav-item active">
```

```
132.     <a class="nav-link" href="#">Active</a>
133. </li>
134. <li class="nav-item">
135.     <a class="nav-link" href="#">Link</a>
136. </li>
137. <li class="nav-item">
138.     <a class="nav-link" href="#">Link</a>
139. </li>
140. <li class="nav-item">
141.     <a class="nav-link disabled" href="#">Disabled</a>
142. </li>
143. </ul>
144. </nav>
145. </body>
146. </html>
```

---

## Brand / Logo

The .navbar-brand class is used to highlight the brand/logo/project name of your page.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
13.
14. <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
15. <!-- Brand/logo -->
```

```

16. <a class="navbar-brand" href="#">Logo</a>
17.
18. <!-- Links -->
19. <ul class="navbar-nav">
20.   <li class="nav-item">
21.     <a class="nav-link" href="#">Link 1</a>
22.   </li>
23.   <li class="nav-item">
24.     <a class="nav-link" href="#">Link 2</a>
25.   </li>
26.   <li class="nav-item">
27.     <a class="nav-link" href="#">Link 3</a>
28.   </li>
29. </ul>
30. </nav>
31.
32. <div class="container-fluid">
33.   <h3>Brand / Logo</h3>
34. </div>
35.
36. </body>
37. </html>

```

If you use .navbar-brand class on images, Bootstrap 4 will automatically style the image to fit the navbar.

## Example:

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>

```

```
12. <body>
13. <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
14. <!-- Brand/logo -->
15. <a class="navbar-brand" href="#">
16.   
17. </a>
18. <!-- Links -->
19. <ul class="navbar-nav">
20.   <li class="nav-item">
21.     <a class="nav-link" href="#">Link 1</a>
22.   </li>
23.   <li class="nav-item">
24.     <a class="nav-link" href="#">Link 2</a>
25.   </li>
26.   <li class="nav-item">
27.     <a class="nav-link" href="#">Link 3</a>
28.   </li>
29. </ul>
30. </nav>
31. <div class="container-fluid">
32.   <h3>Brand / Logo</h3>
33. </div>
34. </body>
35. </html>
```

---

## Navbar Forms and Buttons

The <form> element with class="form-inline" is used to group inputs and buttons side-by-side.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```



9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
13. `<nav class="navbar navbar-expand-sm bg-dark navbar-dark">`
14. `<form class="form-inline" action="/action_page.php">`
15. `<input class="form-control" type="text" placeholder="Search">`
16. `<button class="btn btn-success" type="submit">Search</button>`
17. `</form>`
18. `</nav>`
19. `<br>`
20. `<div class="container">`
21. `<h3>Navbar Forms</h3>`
22. `<p>Use the .form-inline class to align form elements side by side inside the navbar.</p>`
23. `</div>`
24. `</body>`
25. `</html>`

You can also use other input classes, such as `.input-group-addon` to attach an icon or help text next to the input field. You will learn more about these classes in the Bootstrap Inputs chapter.

## Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
13. `<nav class="navbar navbar-expand-sm bg-dark navbar-dark">`

```

14. <form class="form-inline">
15.   <div class="input-group">
16.     <span class="input-group-addon">@</span>
17.     <input type="text" class="form-control" placeholder="Username">
18.   </div>
19. </form>
20. </nav>
21. <br>
22. <div class="container">
23.   <h3>Navbar Forms</h3>
24.   <p>Use the .form-inline class to align form elements side by side inside the navbar.</p>
25. </div>
26. </body>
27. </html>

```

---

## Navbar Text

The .navbar-text class is used to vertical align any elements inside the navbar that are not links. You must ensure the proper padding and text color.

### Example:

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
13. <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
14.   <!-- Links -->
15.   <ul class="navbar-nav">
16.     <li class="nav-item">

```

```
17.   <a class="nav-link" href="#">Link 1</a>
18.   </li>
19.   <li class="nav-item">
20.     <a class="nav-link" href="#">Link 2</a>
21.   </li>
22. </ul>
23. <!-- Navbar text-->
24. <span class="navbar-text">
25.   Navbar text
26. </span>
27. </nav>
28. <br>
29. <div class="container">
30.   <h3>Navbar Text</h3>
31. </div>
32. </body>
33. </html>
```

---

## Fixed Navigation Bar

You can make navigation bar to be fixed at the top or at the bottom of the page. The fixed navigation bar stays visible at a fixed position (top or bottom) without depending on page scrolling.

### Top Fixed navbar

The .fixed-top class makes the navigation bar fixed at the top.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
```

```
10. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-  
    beta.2/js/bootstrap.min.js"></script>  
11. </head>  
12. <body style="height:1500px">  
13. <nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-top">  
14.   <a class="navbar-brand" href="#">Logo</a>  
15.   <ul class="navbar-nav">  
16.     <li class="nav-item">  
17.       <a class="nav-link" href="#">Link</a>  
18.     </li>  
19.     <li class="nav-item">  
20.       <a class="nav-link" href="#">Link</a>  
21.     </li>  
22.   </ul>  
23. </nav>  
24. <div class="container-fluid" style="margin-top:80px">  
25.   <h3>Top Fixed Navbar</h3>  
26. </div>  
27. </body>  
28. </html>
```

---

## Bottom Fixed Navbar

The .fixed-bottom class is used to make the navbar stay at the bottom of the page.

### Example:

```
1. <!DOCTYPE html>  
2. <html lang="en">  
3. <head>  
4.   <title>Bootstrap Example</title>  
5.   <meta charset="utf-8">  
6.   <meta name="viewport" content="width=device-width, initial-scale=1">  
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-  
    beta.2/css/bootstrap.min.css">  
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><  
    /script>  
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-  
    beta.2/js/bootstrap.min.js"></script>  
11. </head>
```

```
12. <body style="height:1500px">
13. <nav class="navbar navbar-expand-sm bg-dark navbar-dark fixed-bottom">
14. <a class="navbar-brand" href="#">Logo</a>
15. <ul class="navbar-nav">
16. <li class="nav-item">
17. <a class="nav-link" href="javascript:void(0)">Link</a>
18. </li>
19. <li class="nav-item">
20. <a class="nav-link" href="javascript:void(0)">Link</a>
21. </li>
22. </ul>
23. </nav>
24. <div class="container-fluid"><br>
25. <h3>Bottom Fixed Navbar</h3>
26. <h1>Scroll this page to see the effect</h1>
27. </div>
28. </body>
29. </html>
```

---

## Sticky Navbar

A sticky navigation bar stays fixed at the top of the page when you scroll past it. You can see the effect by scrolling this page.

*Note: This facility doesn't work in IE11 and earlier version.*

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
```

```
12. <body style="height:1500px">
13.
14. <div class="container-fluid">
15.   <br>
16.   <h3>Sticky Navbar</h3>
17.   <p>You can see the effect by scrolling this page.</p>
18.   <p><strong>Note:</strong> This facility doesn't work in IE11 and earlier version. </p>
19. </div>
20.
21. <nav class="navbar navbar-expand-sm bg-dark navbar-dark sticky-top">
22.   <a class="navbar-brand" href="#">Logo</a>
23.   <ul class="navbar-nav">
24.     <li class="nav-item">
25.       <a class="nav-link" href="#">Link</a>
26.     </li>
27.     <li class="nav-item">
28.       <a class="nav-link" href="#">Link</a>
29.     </li>
30.   </ul>
31. </nav>
32.
33. <div class="container-fluid"><br>
34.   <p>Some example text. Some example text. Some example text. Some example text. Some
    example text.</p>
35.   <p>Some example text. Some example text. Some example text. Some example text. Some
    example text.</p>
36.   <p>Some example text. Some example text. Some example text. Some example text. Some
    example text.</p>
37.   <p>Some example text. Some example text. Some example text. Some example text. Some
    example text.</p>
38. </div>
39.
40. </body>
41. </html>
```

## Bootstrap Images

Bootstrap supports for images. There are three classes in Bootstrap that can be used to apply some simple style to the images.

The following classes add style to the images:

Classes	Uses
.img-rounded	It adds border-radius:6px to give the image rounded corners.
.img-circle	It makes the entire image round by adding border-radius:500px.
.img-thumbnail	It adds a bit of padding and a gray border.

## Bootstrap Image-rounded Example

The **class .img-rounded** is used to add rounded corners to an image ( IE8 does not support rounded corners).

### Example:

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap image</title>
5.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
6. </head>
7. <body>
8.   <div class="container">
9.     <h2>Rounded Corners</h2>
10.    
11.   </div>
- 12.
13. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
14. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
15. </body>
16. </html>

## Bootstrap Image-circle Example

The **class .img-circle** is used to shape the image to a circle (IE8 does not support rounded corners).

#### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap image</title>
5.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
6. </head>
7. <body>
8.   <div class="container">
9.     <h2>Circle</h2>
10.    
11.  </div>
12.
13. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
14. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
15. </body>
16. </html>
17.
18.
```

## Bootstrap Thumbnail Image Example

The **class .img-thumbnail** is used to shape an image to a thumbnail.

#### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap image</title>
5.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
6. </head>
7. <body>
8.   <div class="container">
9.     <h2>Thumbnail</h2>
```



```
10. 
11. </div>
12.
13. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
14. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scri
    pt>
15. </body>
16. </html>
```

---

## Bootstrap Responsive images

The responsive images can adjust themselves automatically to fit the size of screen. You can create responsive images by adding an **.img-responsive class** to the `<img>` tag. The image will then scale nicely to the parent element.

The **.img-responsive class** applies `display: block;` and `max-width: 100%;` and `height: auto;` to the image.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3.
4. <head>
5.   <title>Bootstrap Example</title>
6.   <meta charset="utf-8">
7.   <meta name="viewport" content="width=device-width, initial-scale=1">
8.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
    ap.min.css">
9. </head>
10. <body>
11.
12. <div class="container">
13.   <h2>Responsive Image</h2>
14.   
15. </div>
16. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
17. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scri
    pt>
18.
19. </body>
```

20. `</html>`

21.

22.

---

## Bootstrap Responsive Videos / Embeds

In Bootstrap, you can also add videos and scale them properly on any devices. The **class .embed-responsive-item** is used to create a responsive video. Class can be applied directly to `<iframe>`, `<embed>`, `<video>`, and `<object>` elements.

### Let's take an example:

In the following example, we add **.embed-responsive-item class** to an `<iframe>` tag to make the video responsive. It can scale the video nicely according to the parent element.

### Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
8. </head>
9. <body>
10.
11. <div class="container">
12.   <h2>Responsive Embed</h2>
13.   <div class="embed-responsive embed-responsive-16by9">
14.     <iframe class="embed-responsive-item" src="https://www.youtube.com/embed/XGSy3_Czz8k"></iframe>
15.   </div>
16. </div>
17. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
18. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
19.
20. </body>
21. </html>
```

---

# Bootstrap4 Images

Aligning images are used to float an image to the right with the .float-right class or to the left with .float-left.

## Example:

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>
10. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
- 13.
14. <div class="container">
15. <h2>Aligning images Example</h2>
16. <p>Use the float classes to float the image to the left or to the right:</p>
17. 
18. 
19. </div>
- 20.
21. </body>
22. </html>

# Bootstrap Pagination

Pagination is used to sort the web pages of your website in an organized manner. It becomes very necessary if your website has a lot of web pages.

Following is a list of classes that Bootstrap provides to handle pagination.

## Table:

Class	Description
.pagination	Add this class to get the pagination on your page.
.disabled, .active	you can customize links by using .disabled for unclickable links and .active to indicate current page.
.pagination-lg, .pagination-sm	use these classes to get different size items.

## Bootstrap Pagination Example

Add the **.pagination class** to an `<ul>` element to create a basic pagination.

### See this example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
- 8.
9. `</head>`
10. `<body>`
- 11.
12. `<div class="container">`
13. `<h2>A basic pagination example:</h2>`
14. `<p>The .pagination class provides pagination links:</p>`
15. `<ul class="pagination">`
16. `<li><a href="#">1</a></li>`
17. `<li><a href="#">2</a></li>`

```
18. <li><a href="#">3</a></li>
19. <li><a href="#">4</a></li>
20. <li><a href="#">5</a></li>
21. </ul>
22. </div>
23.
24. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
25. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
26.
27. </body>
28. </html>
```

---

## Bootstrap Active State

The active state specifies the current page on which user is active.

The **class .active** is used to specify active state.

**See this example:**

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13. <h2>Active State Pagination: Example</h2>
14. <p>Specify the current active state of the user.</p>
15. <ul class="pagination">
16. <li><a href="#">1</a></li>
17. <li class="active"><a href="#">2</a></li>
18. <li><a href="#">3</a></li>
19. <li><a href="#">4</a></li>
```

```

20. <li><a href="#">5</a></li>
21. </ul>
22. </div>
23.
24. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
25. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scri
    pt>
26.
27. </body>
28. </html>

```

---

## Bootstrap Disabled State

The disabled state specifies that you cannot click on the link.

Add **class .disabled** to disable the links you don't need more.

**See this example:**

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
    ap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13. <h2>Disabled State Pagination Example:</h2>
14. <p>Here, 4 and 5 links are disabled.</p>
15. <ul class="pagination">
16. <li><a href="#">1</a></li>
17. <li><a href="#">2</a></li>
18. <li><a href="#">3</a></li>
19. <li class="disabled"><a href="#">4</a></li>
20. <li><a href="#">5</a></li>
21. </ul>

```

```

22. </div>
23.
24. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"> </script>
25. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"> </scri
    pt>
26.
27. </body>
28. </html>

```

---

## Bootstrap Pagination Sizing

You can also size pagination blocks larger or smaller accordingly.

Add **class .pagination-lg** for larger blocks or .pagination-sm for smaller blocks.

**See this example:**

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
    ap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13.   <h2>Pagination - Sizing</h2>
14.   <p>Add class .pagination-lg for larger blocks or .pagination-sm for smaller blocks:</p>
15.   <ul class="pagination pagination-lg">
16.     <li><a href="#">1</a></li>
17.     <li><a href="#">2</a></li>
18.     <li><a href="#">3</a></li>
19.     <li><a href="#">4</a></li>
20.     <li><a href="#">5</a></li>
21.   </ul>
22.
23.   <ul class="pagination pagination-sm">

```

```
24. <li><a href="#">1</a></li>
25. <li><a href="#">2</a></li>
26. <li><a href="#">3</a></li>
27. <li><a href="#">4</a></li>
28. <li><a href="#">5</a></li>
29. </ul>
30. </div>
31.
32. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
33. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
34.
35. </body>
36. </html>
```

---

## Bootstrap Breadcrumbs

It is another form of pagination. It indicates the current page's location within a navigational hierarchy.

The **.breadcrumb class** is used to create breadcrumbs.

### See this example:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta name="viewport" content="width=device-width, initial-scale=1">
5. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
6.
7. </head>
8. <body>
9.
10. <div class="container">
11. <h2>Breadcrumbs Example:</h2>
12. <p>The .breadcrumb class indicates the current page's location within a navigational hierarchy:</p>
13. <ul class="breadcrumb">
14. <li><a href="#">Java</a></li>
15. <li><a href="#">SQL</a></li>
```



```

16. <li><a href="#">Php</a></li>
17. <li class="active">.Net</li>
18. </ul>
19. </div>
20.
21. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
22. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
23.
24. </body>
25. </html>

```

## Bootstrap Jumbotron

A Bootstrap jumbotron specifies a big box for getting extra attention to some special content or information. It is displayed as a grey box with rounded corners. It can also enlarge the font sizes of the text inside it.

You can put any valid HTML or other Bootstrap elements/ classes inside a jumbotron.

The **class .jumbotron** within the `<div>` element is used to create a jumbotron.

---

## Jumbotron Inside Container

The Inside container is used in jumbotron, if you want the jumbotron to not extend to the edge of the screen.

Put the jumbotron inside the `<div class="container">`.

### Example:

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
8. </head>
9. <body>
10.

```

11. `<div class="container">`
12. `<div class="jumbotron">`
13. `<h1>This is Jumbotron inside container!</h1>`
14. `<p>Jumbotron specifies a big box for getting extra attention to some special content or information.</p>`
15. `</div>`
16. `<p>This is some text.</p>`
17. `<p>This is another text.</p>`
18. `</div>`
- 19.
20. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
21. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
22. `</body>`
23. `</html>`

## Jumbotron Outside Container

It is used when you want the jumbotron to extend to the screen edges.

Put the jumbotron outside the `<div class="container">`.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
8. `</head>`
9. `<body>`
- 10.
11. `<div class="jumbotron">`
12. `<h1>This is Jumbotron outside container!</h1>`
13. `<p>Jumbotron specifies a big box for getting extra attention to some special content or information.</p>`
14. `</div>`
- 15.
16. `<div class="container">`

17. `<p>This is some text.</p>`
  18. `<p>This is another text.</p>`
  19. `</div>`
  - 20.
  21. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
  22. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
  23. `</body>`
  24. `</html>`
- 

## Full-width Jumbotron

To get a jumbotron without rounded borders, you have to add the `.jumbotron-fluid` class and a `.container` or `.container-fluid` inside it.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="jumbotron jumbotron-fluid">`
15. `<div class="container">`
16. `<h1>Full-width Jumbotron</h1>`
17. `<p>Jumbotron specifies a big box for getting extra attention to some special content or information.</p>`
18. `</div>`
- 19.
20. `<div class="container">`

21. `<p>This is some text.</p>`
22. `<p>This is another text.</p>`
23. `</div>`
- 24.
25. `</body>`
26. `</html>`

## Bootstrap Page Header

A page header is like a section divider. It adds a horizontal line under the heading.

The **.page-header class** is used to create a page header.

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
8. `</head>`
9. `<body>`
- 10.
11. `<div class="container">`
12. `<div class="page-header">`
13. `<h1>Example Page Header</h1>`
14. `</div>`
15. `<p>This is some text.</p>`
16. `<p>This is another text.</p>`
17. `</div>`
- 18.
19. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`
20. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`
21. `</body>`
22. `</html>`

## Bootstrap Alerts

Bootstrap Alerts are used to provide an easy way to create predefined alert messages. Alert adds a style to your messages to make it more appealing to the users.

There are four classes that are used within <div> element for alerts.

- .alert-success
- .alert-info
- .alert-warning
- .alert-danger

## Bootstrap Alert Example

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
9. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
10. </head>
11. <body>
- 12.
13. <div class="container">
14. <h2>Alerts</h2>
15. <div class="alert alert-success">
16. <strong>Success!</strong> This alert box indicates a successful or positive action.
17. </div>
18. <div class="alert alert-info">
19. <strong>Info!</strong> This alert box indicates a neutral informative change or action.
20. </div>
21. <div class="alert alert-warning">
22. <strong>Warning!</strong> This alert box indicates a warning that might need attention.
23. </div>
24. <div class="alert alert-danger">

25. `<strong>`Danger!`</strong>` This alert box indicates a dangerous or potentially negative action.
26. `</div>`
27. `</div>`
- 28.
29. `</body>`
30. `</html>`

## Bootstrap4 Alert

Bootstrap 4 adds 4 new alerts in Bootstrap Alert defined in Bootstrap 3 tutorial.

These are:

- **Primary:** This alert box indicates an important action.
- **Secondary:** This alert box indicates a less important action.
- **Dark:** Dark grey alert box.
- **Light:** Light grey alert box.

Alerts are created with the `.alert` class, followed by one of the contextual classes.

### List of all contextual classes:

- `.alert-success`
- `.alert-info`
- `.alert-warning`
- `.alert-danger`
- `.alert-primary`
- `.alert-secondary`
- `.alert-light`
- `.alert-dark`

## Example

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>`Bootstrap Example`</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`

7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`

8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`

9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`

10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`

11. `</head>`

12. `<body>`

13.

14. `<div class="container">`

15. `<h2>Alerts</h2>`

16. `<div class="alert alert-success">`

17. `<strong>Success!</strong>` Used to indicate successful or positive action.

18. `</div>`

19. `<div class="alert alert-info">`

20. `<strong>Info!</strong>` Used to indicate a neutral informative change or action.

21. `</div>`

22. `<div class="alert alert-warning">`

23. `<strong>Warning!</strong>` Used to indicate a warning that might need attention.

24. `</div>`

25. `<div class="alert alert-danger">`

26. `<strong>Danger!</strong>` Used to indicate a dangerous or potentially negative action.

27. `</div>`

28. `<div class="alert alert-primary">`

29. `<strong>Primary!</strong>` Used to indicate an important action.

30. `</div>`

31. `<div class="alert alert-secondary">`

32. `<strong>Secondary!</strong>` Used to indicate a slightly less important action.

33. `</div>`

34. `<div class="alert alert-dark">`

35. `<strong>Dark!</strong>` Dark grey alert.

36. `</div>`

37. `<div class="alert alert-light">`

38. `<strong>Light!</strong>` Light grey alert.

39. `</div>`

40. `</div>`

41.

42. `</body>`

43. `</html>`

## Alert Links

You have to add the alert-link class to any links inside the alert box to create "matching colored links":

### Example:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="container">`
15. `<h2>Bootstrap Alert Links</h2>`
16. `<div class="alert alert-success">`
17. `<strong>Success!</strong> You should <a href="#" class="alert-link">read this message</a>.`
18. `</div>`
19. `<div class="alert alert-info">`
20. `<strong>Info!</strong> You should <a href="#" class="alert-link">read this message</a>.`
21. `</div>`
22. `<div class="alert alert-warning">`
23. `<strong>Warning!</strong> You should <a href="#" class="alert-link">read this message</a>.`
24. `</div>`
25. `<div class="alert alert-danger">`



```

26. <strong>Danger!</strong> You should <a href="#" class="alert-
    link">read this message</a>.
27. </div>
28. <div class="alert alert-primary">
29. <strong>Primary!</strong> You should <a href="#" class="alert-
    link">read this message</a>.
30. </div>
31. <div class="alert alert-secondary">
32. <strong>Secondary!</strong> You should <a href="#" class="alert-
    link">read this message</a>.
33. </div>
34. <div class="alert alert-dark">
35. <strong>Dark!</strong> You should <a href="#" class="alert-
    link">read this message</a>.
36. </div>
37. <div class="alert alert-light">
38. <strong>Light!</strong> You should <a href="#" class="alert-
    link">read this message</a>.
39. </div>
40. </div>
41.
42. </body>
43. </html>

```

## Closing Alerts

If you want to close the alert message, you have to add an `.alert-dismissible` class to the alert container. Then add `class="close"` and `data-dismiss="alert"` to a link or a button element (when you click on the close sign, the alert box will be closed.)

### Example:

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
    beta.2/css/bootstrap.min.css">
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"> </script>

```

```
9. <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
    beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
13.
14. <div class="container">
15.   <h2>Alerts</h2>
16.   <div class="alert alert-success alert-dismissible">
17.     <button type="button" class="close" data-dismiss="alert"> × </button>
18.     <strong>Success!</strong> This alert box could indicate a successful or positive action.
19.   </div>
20.   <div class="alert alert-info alert-dismissible">
21.     <button type="button" class="close" data-dismiss="alert"> × </button>
22.     <strong>Info!</strong> This alert box could indicate a neutral informative change or act
    ion.
23.   </div>
24.   <div class="alert alert-warning alert-dismissible">
25.     <button type="button" class="close" data-dismiss="alert"> × </button>
26.     <strong>Warning!</strong> This alert box could indicate a warning that might need att
    ention.
27.   </div>
28.   <div class="alert alert-danger alert-dismissible">
29.     <button type="button" class="close" data-dismiss="alert"> × </button>
30.     <strong>Danger!</strong> This alert box could indicate a dangerous or potentially nega
    tive action.
31.   </div>
32.   <div class="alert alert-primary alert-dismissible">
33.     <button type="button" class="close" data-dismiss="alert"> × </button>
34.     <strong>Primary!</strong> Indicates an important action.
35.   </div>
36.   <div class="alert alert-secondary alert-dismissible">
37.     <button type="button" class="close" data-dismiss="alert"> × </button>
38.     <strong>Secondary!</strong> Indicates a slightly less important action.
39.   </div>
40.   <div class="alert alert-dark alert-dismissible">
41.     <button type="button" class="close" data-dismiss="alert"> × </button>
42.     <strong>Dark!</strong> Dark grey alert.
43.   </div>
```

```

44. <div class="alert alert-light alert-dismissible">
45.   <button type="button" class="close" data-dismiss="alert"> × </button>
46.   <strong>Light!</strong> Light grey alert.
47. </div>
48. </div>
49.
50. </body>
51. </html>

```

## Animated Alerts

You can use `.fade` and `.show` classes to add a fading effect when closing the alert message. Example:

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"> </script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"> <
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"> </script>
11. </head>
12. <body>
13.
14. <div class="container">
15.   <h2>Animated Alerts Example</h2>
16.   <div class="alert alert-success alert-dismissible fade show">
17.     <button type="button" class="close" data-dismiss="alert"> × </button>
18.     <strong>Success!</strong> This alert box could indicate a successful or positive action.
19.   </div>
20.   <div class="alert alert-info alert-dismissible fade show">
21.     <button type="button" class="close" data-dismiss="alert"> × </button>
22.     <strong>Info!</strong> This alert box could indicate a neutral informative change or act
  ion.
23.   </div>
24.   <div class="alert alert-warning alert-dismissible fade show">

```

```

25. <button type="button" class="close" data-dismiss="alert"> × </button>
26. <strong>Warning!</strong> This alert box could indicate a warning that might need att
    ention.
27. </div>
28. <div class="alert alert-danger alert-dismissible fade show">
29. <button type="button" class="close" data-dismiss="alert"> × </button>
30. <strong>Danger!</strong> This alert box could indicate a dangerous or potentially nega
    tive action.
31. </div>
32. <div class="alert alert-primary alert-dismissible fade show">
33. <button type="button" class="close" data-dismiss="alert"> × </button>
34. <strong>Primary!</strong> Indicates an important action.
35. </div>
36. <div class="alert alert-secondary alert-dismissible fade show">
37. <button type="button" class="close" data-dismiss="alert"> × </button>
38. <strong>Secondary!</strong> Indicates a slightly less important action.
39. </div>
40. <div class="alert alert-dark alert-dismissible fade show">
41. <button type="button" class="close" data-dismiss="alert"> × </button>
42. <strong>Dark!</strong> Dark grey alert.
43. </div>
44. <div class="alert alert-light alert-dismissible fade show">
45. <button type="button" class="close" data-dismiss="alert"> × </button>
46. <strong>Light!</strong> Light grey alert.
47. </div>
48. </div>
49.
50. </body>
51. </html>

```

## Bootstrap Forms

In Bootstrap, there are three types of form layouts:

- Vertical form (this is default)
- Horizontal form
- Inline form

---

## Bootstrap Form Rules

There are three standard rules for these 3 form layouts:

- Always use `<form role="form">` (helps improve accessibility for people using screen readers)
  - Wrap labels and form controls in `<div class="form-group">` (needed for optimum spacing)
  - Add class `.form-control` to all textual `<input>`, `<textarea>`, and `<select>` elements
- 

## 1) Bootstrap Vertical Form (Default)

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap example</title>`
5. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>`
6. `</head>`
7. `<body>`
- 8.
9. `<div class="container">`
10. `<h1>Vertical Form Example</h1>`
- 11.
12. `<form style="width:300px">`
13. `<div class="form-group">`
14. `<label for="exampleInputEmail1">Email address</label>`
15. `<input type="email" class="form-control" id="exampleInputEmail1" placeholder="Email">`
16. `</div>`
17. `<div class="form-group">`
18. `<label for="exampleInputPassword1">Password</label>`
19. `<input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">`
20. `</div>`
- 21.
22. `<button type="submit" class="btn btn-default">Login</button>`
23. `</form>`
- 24.
25. `</div>`
- 26.
27. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>`

28. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>`
29. `</body>`
30. `</html>`

## 2) Bootstrap Inline Form

In Bootstrap Inline forms, all elements are inline, left-aligned, and the labels are alongside.

This example is only applied to forms within viewports that are at least 768px wide!

### Example:

- 1.
2. `<!DOCTYPE html>`
3. `<html lang="en">`
4. `<head>`
5. `<title>Bootstrap Example</title>`
6. `<meta charset="utf-8">`
7. `<meta name="viewport" content="width=device-width, initial-scale=1">`
8. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`
- 9.
10. `</head>`
11. `<body>`
- 12.
13. `<div class="container">`
14. `<h2>Inline form Example</h2>`
15. `<form class="form-inline" role="form">`
16. `<form style="width:300px">`
17. `<div class="form-group">`
18. `<label for="exampleInputEmail1">Email address</label>`
19. `<input type="email" class="form-control" id="exampleInputEmail1" placeholder="Email">`
20. `</div>`
21. `<div class="form-group">`
22. `<label for="exampleInputPassword1">Password</label>`

```

23. <input type="password" class="form-
    control" id="exampleInputPassword1" placeholder="Password">
24. </div>
25.
26. <button type="submit" class="btn btn-default">Login</button>
27. </form>
28.
29. </div>
30. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
31. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scri
    pt>
32.
33. </body>
34. </html>

```

### 3) Bootstrap Horizontal Form

You have to add some additional rules if you want to create a horizontal form.

#### Additional rules for a horizontal form:

- Add class .form-horizontal to the <form> element
- Add class .control-label to all <label> elements

#### Example:

```

1.
2. <!DOCTYPE html>
3. <html lang="en">
4. <head>
5.   <title>Bootstrap Example</title>
6.   <meta charset="utf-8">
7.   <meta name="viewport" content="width=device-width, initial-scale=1">
8.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
    ap.min.css">
9.
10. </head>
11. <body>
12.
13. <div class="container">
14.   <h2>Horizontal form Example</h2>

```

```

15. <form class="form-horizontal" role="form">
16.   <form style="width:300px">
17.     <div class="form-group">
18.       <label class="control-label col-sm-2" for="email">Email:</label>
19.       <div class="col-sm-10">
20.         <input type="email" class="form-control" id="email" placeholder="Enter email">
21.       </div>
22.     </div>
23.     <div class="form-group">
24.       <label class="control-label col-sm-2" for="pwd">Password:</label>
25.       <div class="col-sm-10">
26.         <input type="password" class="form-
control" id="pwd" placeholder="Enter password">
27.       </div>
28.     </div>
29.
30.   <div class="form-group">
31.     <div class="col-sm-offset-2 col-sm-10">
32.       <button type="submit" class="btn btn-default">Submit</button>
33.     </div>
34.   </div>
35. </form>
36. </div>
37. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
38. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scrip
t>
39.
40. </body>
41. </html>

```

## Bootstrap 4 Forms

In Bootstrap4, form controls automatically receive some global styling with Bootstrap.

All textual `<input>`, `<textarea>`, and `<select>` elements with class `.form-control` have a width of 100%.

### Stacked (full-width) form:

Bootstrap 4 provides full width stacked forms.



## Example:

Let's take an example to create a stacked form with two input fields, one checkbox, and a submit button.

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="container">`
15. `<h2>Stacked form</h2>`
16. `<form>`
17. `<div class="form-group">`
18. `<label for="email">Email:</label>`
19. `<input type="email" class="form-control" id="email" placeholder="Enter email">`
20. `</div>`
21. `<div class="form-group">`
22. `<label for="pwd">Password:</label>`
23. `<input type="password" class="form-control" id="pwd" placeholder="Enter password">`
24. `</div>`
25. `<div class="form-check">`
26. `<label class="form-check-label">`
27. `<input class="form-check-input" type="checkbox"> Remember me`
28. `</label>`
29. `</div>`
30. `<button type="submit" class="btn btn-primary">Submit</button>`
31. `</form>`
32. `</div>`

33.

34. `</body>`

35. `</html>`

## Bootstrap Progress Bar

The progress bar shows how far a user is in a process. In Bootstrap, there are several types of progress bars.

The **class .progress** within a `<div>` element is used to create a default progress bar in bootstrap.

## Bootstrap Progress Bar Example

1. `<!DOCTYPE html>`

2. `<html lang="en">`

3. `<head>`

4. `<title>Bootstrap Progress bar</title>`

5. `<meta charset="utf-8">`

6. `<meta name="viewport" content="width=device-width, initial-scale=1">`

7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">`

8.

9. `</head>`

10. `<body>`

11.

12. `<div class="container">`

13. `<h2>Default Progress Bar</h2>`

14. `<div class="progress">`

15. `<div class="progress-bar" role="progressbar" aria-valuenow="70" aria-valuemin="0" aria-valuemax="100" style="width:70%">`

16. `<span class="sr-only">70% Complete</span>`

17. `</div>`

18. `</div>`

19. `</div>`

20. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>`

21. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>`

22. `</body>`

23. `</html>`

# Bootstrap 4 Basic Progress Bar

Progress bar is used to show a user how far long he/she is in a process. To create a default progress bar, add a .progress class to a container element and the progress-bar class to its child element. Use the CSS width property to set the width of the progress bar.

## Example

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`
7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="container">`
15. `<h2>Progress Bar With Label</h2>`
16. `<div class="progress">`
17. `<div class="progress-bar" style="width:70%">70%</div>`
18. `</div>`
19. `</div>`
- 20.
21. `</body>`
22. `</html>`

## Bootstrap Progress bar with Label

The progress bar with label specifies the percentage of progress of a specific process.

You have to remove the .sr-only class from the progress bar to show a visible percentage.

### See this example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13.   <h2>Progress Bar With Label</h2>
14.   <div class="progress">
15.     <div class="progress-bar" role="progressbar"
16.       aria-valuenow="76" aria-valuemin="0" aria-valuemax="100" style="width:76%">
17.       76%
18.     </div>
19.   </div>
20. </div>
21.
22. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
23. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
24.
25. </body>
26. </html>
```

## Bootstrap Colored Progress bar

You can use contextual classes to create colored progress bar.

The contextual classes that are used to create colored progress bar:

- .progress-bar-success

- .progress-bar-info
- .progress-bar-warning
- .progress-bar-danger

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <title>Bootstrap Example</title>
5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
  ap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13. <h2>Colored Progress Bars</h2>
14. <p>The contextual classes colors the progress bars:</p>
15. <div class="progress">
16.   <div class="progress-bar progress-bar-success"
17.     role="progressbar" aria-valuenow="40" aria-valuemin="0" aria-
  valuemax="100" style="width:40%">
18.     40% Complete (success)
19.   </div>
20. </div>
21. <div class="progress">
22.   <div class="progress-bar progress-bar-info"
23.     role="progressbar" aria-valuenow="50" aria-valuemin="0" aria-
  valuemax="100" style="width:50%">
24.     50% Complete (info)
25.   </div>
26. </div>
27. <div class="progress">
28.   <div class="progress-bar progress-bar-warning"
29.     role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-
  valuemax="100" style="width:60%">
30.     60% Complete (warning)
31.   </div>
32. </div>

```

```

33. <div class="progress">
34.   <div class="progress-bar progress-bar-danger"
35.     role="progressbar" aria-valuenow="70" aria-valuemin="0" aria-
       valuemax="100" style="width:70%">
36.     70% Complete (danger)
37.   </div>
38. </div>
39. </div>
40.
41. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
42. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scrip
    t>
43.
44. </body>
45. </html>

```

## Bootstrap Stripped Progress bar

You can create stripped progress bar by using **class .progress-bar-striped** .

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
    ap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13.   <h2>Striped Progress Bars</h2>
14.   <p>The .progress-bar-striped class adds stripes to the progress bars:</p>
15.   <div class="progress">
16.     <div class="progress-bar progress-bar-success progress-bar-striped"
17.       role="progressbar" aria-valuenow="40" aria-valuemin="0" aria-
         valuemax="100" style="width:40%">
18.       40% Complete (success)

```

```

19. </div>
20. </div>
21. <div class="progress">
22.   <div class="progress-bar progress-bar-info progress-bar-striped"
23.     role="progressbar" aria-valuenow="50" aria-valuemin="0" aria-
        valuemax="100" style="width:50%">
24.     50% Complete (info)
25.   </div>
26. </div>
27. <div class="progress">
28.   <div class="progress-bar progress-bar-warning progress-bar-striped"
29.     role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-
        valuemax="100" style="width:60%">
30.     60% Complete (warning)
31.   </div>
32. </div>
33. <div class="progress">
34.   <div class="progress-bar progress-bar-danger progress-bar-striped"
35.     role="progressbar" aria-valuenow="70" aria-valuemin="0" aria-
        valuemax="100" style="width:70%">
36.     70% Complete (danger)
37.   </div>
38. </div>
39. </div>
40.
41. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
42. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scrip
    t>
43.
44. </body>
45. </html>

```

---

## Bootstrap Animated progress bar

You have to use **class .active** to create animated progress bar.

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>

```

```

5. <meta charset="utf-8">
6. <meta name="viewport" content="width=device-width, initial-scale=1">
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstr
  ap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13. <h2>Animated Progress Bar</h2>
14. <p>The .active class animates the progress bar:</p>
15. <div class="progress">
16.   <div class="progress-bar progress-bar-striped active"
17.     role="progressbar" aria-valuenow="70" aria-valuemin="0" aria-
      valuemax="100" style="width:70%">
18.     70%
19.   </div>
20. </div>
21. </div>
22.
23. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
24. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></scrip
    t>
25.
26. </body>
27. </html>

```

## Bootstrap Stacked Progress bar (Multi-colored progress bar)

You can create stacked progress bar by placing multiple bars into the same <div class="progress">

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">

```



```
7. <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
8.
9. </head>
10. <body>
11.
12. <div class="container">
13. <h2>Stacked Progress Bars</h2>
14. <p>Create a stacked progress bar by placing multiple bars into the same div with class .progress:</p>
15. <div class="progress">
16. <div class="progress-bar progress-bar-success"
17. role="progressbar" style="width:40%">
18.   Free Space
19. </div>
20. <div class="progress-bar progress-bar-warning"
21. role="progressbar" style="width:10%">
22.   Warning
23. </div>
24. <div class="progress-bar progress-bar-danger"
25. role="progressbar" style="width:20%">
26.   Danger
27. </div>
28. </div>
29. </div>
30.
31. <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
32. <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
33.
34. </body>
35. </html>
```

*Note: Two types of new colored progress bar are added in Bootstrap 4:*

- .progress-bar-white
- .progress-bar-secondary
- .progress-bar-light
- .progress-bar-dark

## Example:

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <title>Bootstrap Example</title>
5.   <meta charset="utf-8">
6.   <meta name="viewport" content="width=device-width, initial-scale=1">
7.   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/css/bootstrap.min.css">
8.   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9.   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"><
  /script>
10.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
  beta.2/js/bootstrap.min.js"></script>
11. </head>
12. <body>
13.
14. <div class="container">
15.   <h2>Colored Progress Bars Example</h2>
16.   <!-- Blue -->
17.   <div class="progress">
18.     <div class="progress-bar" style="width:10%"></div>
19.   </div><br>
20.
21.   <!-- Green -->
22.   <div class="progress">
23.     <div class="progress-bar bg-success" style="width:20%"></div>
24.   </div><br>
25.
26.   <!-- Turquoise -->
27.   <div class="progress">
28.     <div class="progress-bar bg-info" style="width:30%"></div>
29.   </div><br>
30.
31.   <!-- Orange -->
32.   <div class="progress">
33.     <div class="progress-bar bg-warning" style="width:40%"></div>
34.   </div><br>
35.
36.   <!-- Red -->
```

```
37. <div class="progress">
38.   <div class="progress-bar bg-danger" style="width:50%"> </div>
39. </div> <br>
40.
41. <!-- White -->
42. <div class="progress border">
43.   <div class="progress-bar bg-white" style="width:60%"> </div>
44. </div> <br>
45.
46. <!-- Grey -->
47. <div class="progress">
48.   <div class="progress-bar bg-secondary" style="width:70%"> </div>
49. </div> <br>
50.
51. <!-- Light Grey -->
52. <div class="progress border">
53.   <div class="progress-bar bg-light" style="width:80%"> </div>
54. </div> <br>
55.
56. <!-- Dark Grey -->
57. <div class="progress">
58.   <div class="progress-bar bg-dark" style="width:90%"> </div>
59. </div>
60. </div>
61.
62. </body>
63. </html>
```

## Bootstrap Grid

Wikipedia says:

"In graphic design, a grid is a structure (usually two-dimensional) made up of a series of intersecting straight (vertical, horizontal) lines used to structure the content. It is widely used to design layout and content structure in print design. In web design, it is a very effective method to create a consistent layout rapidly and effectively using HTML and CSS."

---

# Bootstrap Grid System

The Bootstrap Grid System allows up to 12 columns across the page. You can use all 12 columns individually or you can group the columns together to create wider columns.

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Bootstrap Grid System is responsive and the columns are re-arranged automatically according to the screen size.

---

## Grid Classes:

There are four classes in Bootstrap Grid System:

- xs (for phones)
- sm (for tablets)
- md (for desktops)
- lg (for larger desktops)

You can combine the above classes to create more dynamic and flexible layouts.

---

## Basic Structure of a Bootstrap Grid:

1. `<div class="row">`
2. `<div class="col-*-*"> </div>`
3. `</div>`
4. `<div class="row">`
5. `<div class="col-*-*"> </div>`
6. `<div class="col-*-*"> </div>`
7. `<div class="col-*-*"> </div>`
8. `</div>`
9. `<div class="row">`

10. ...

11. `</div>`

---

## Follow the below instructions while creating a Bootstrap Grid:

- Create a row (`<div class="row">`).
  - Add the number of columns, you want in the grid (tags with appropriate `.col-*-*` classes).
  - Note that numbers in `.col-*-*` should always add up to 12 for each row.
- 

## Bootstrap Grid Example

### For equal columns:

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Job</title>`
5. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>`
6. `</head>`
7. `<body>`
- 8.
9. `<div class="container">`
10. `<h1>Grid Example</h1>`
- 11.
12. `<div class="row">`
13. `<div class="col-md-3" style="background-color:lavender;">Rahul</div>`
14. `<div class="col-md-3" style="background-color:lavenderblush;">Vijay</div>`
15. `<div class="col-md-3" style="background-color:lavender;">Kartik</div>`
16. `<div class="col-md-3" style="background-color:lavenderblush;">Ajeet</div>`
17. `</div>`
- 18.
19. `</div>`
20. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"> </script>`
21. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"> </script>`
22. `</body>`
23. `</html>`

### For unequal columns:

```
1. <!DOCTYPE html>
2. <html lang="en">
3.   <head>
4.     <title>Job</title>
5.     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boots
trap.min.css"/>
6.   </head>
7.   <body>
8.
9.     <div class="container">
10.    <h1>Grid Example</h1>
11.
12.    <div class="row">
13.      <div class="col-md-1" style="background-color:lavender;">Rahul</div>
14.      <div class="col-md-2" style="background-color:lavenderblush;">Vijay</div>
15.      <div class="col-md-4" style="background-color:lavender;">Kartik</div>
16.      <div class="col-md-5" style="background-color:lavenderblush;">Ajeet</div>
17.    </div>
18.
19.  </div>
20.
21.  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
22.  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scri
pt>
23. </body>
24. </html>
```

## Bootstrap 4 Grid Classes

There are 5 classes in Bootstrap 4 grid system.

- **.col-** (extra small devices - screen width less than 576px)
- **.col-sm-** (small devices - screen width equal to or greater than 576px)
- **.col-md-** (medium devices - screen width equal to or greater than 768px)
- **.col-lg-** (large devices - screen width equal to or greater than 992px)
- **.col-xl-** (xlarge devices - screen width equal to or greater than 1200px)

You can also combine the above classes to create more dynamic and flexible layouts.

# Structure of Bootstrap 4 Grid

See the basic structure of Bootstrap 4 grid:

1. `<!-- Control the column width, and how they should appear on different devices -->`
2. `<div class="row">`
3. `<div class="col-*-*"></div>`
4. `<div class="col-*-*"></div>`
5. `</div>`
6. `<div class="row">`
7. `<div class="col-*-*"></div>`
8. `<div class="col-*-*"></div>`
9. `<div class="col-*-*"></div>`
10. `</div>`
- 11.
12. `<!-- Or let Bootstrap automatically handle the layout -->`
13. `<div class="row">`
14. `<div class="col"></div>`
15. `<div class="col"></div>`
16. `<div class="col"></div>`
17. `</div>`

First create a row (`<div class="row">`) then add the desired number of columns (tags with appropriate `.col-*-*` classes).

Here: In `.col-*-*` , the first star (\*) represents the responsiveness: sm, md, lg or xl, while the second star represents a number, which should add up to 12 for each row.

---

## Equal Columns Example

Let's take an example to see how to create an equal width column on all devices and screen widths:

### Example

1. `<!DOCTYPE html>`
2. `<html lang="en">`
3. `<head>`
4. `<title>Bootstrap Example</title>`
5. `<meta charset="utf-8">`
6. `<meta name="viewport" content="width=device-width, initial-scale=1">`

7. `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.min.css">`
8. `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>`
9. `<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.6/umd/popper.min.js"></script>`
10. `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta.2/js/bootstrap.min.js"></script>`
11. `</head>`
12. `<body>`
- 13.
14. `<div class="container-fluid">`
15. `<h2>Three equal-width columns</h2>`
16. `<div class="row">`
17. `<div class="col" style="background-color:lavender;">.col</div>`
18. `<div class="col" style="background-color:orange;">.col</div>`
19. `<div class="col" style="background-color:lavender;">.col</div>`
- 20.
21. `</div>`
22. `</div>`
- 23.
24. `</body>`
25. `</html>`

## What is xml

- **Xml** (eXtensible Markup Language) is a mark up language.
- XML is designed to store and transport data.
- Xml was released in late 90's. it was created to provide an easy to use and store self describing data.
- XML became a W3C Recommendation on February 10, 1998.
- XML is not a replacement for HTML.
- XML is designed to be self-descriptive.
- XML is designed to carry data, not to display data.
- XML tags are not predefined. You must define your own tags.
- XML is platform independent and language independent.

*Note: Self-describing data is the data that describes both its content and structure.*

## What is mark-up language



A **mark up language** is a modern system for highlight or underline a document.

Students often underline or highlight a passage to revise easily, same in the sense of modern mark up language highlighting or underlining is replaced by tags.

---

## Prerequisite

Before you start to learn xml, you should know basic of HTML & JavaScript.

---

## Why xml

**Platform Independent and Language Independent:** The main benefit of xml is that you can use it to take data from a program like Microsoft SQL, convert it into XML then share that XML with other programs and platforms. You can communicate between two platforms which are generally very difficult.

The main thing which makes XML truly powerful is its international acceptance. Many corporation use XML interfaces for databases, programming, office application mobile phones and more. It is due to its platform independent feature.

## HTML vs XML

There are many differences between HTML (Hyper Text Markup Language) and XML (eXtensible Markup Language). The important differences are given below:

No.	HTML	XML
1)	HTML is used <b>to display data</b> and focuses on how data looks.	XML is a software and hardware independent tool used <b>to transport and store data</b> . It focuses on what data is.
2)	HTML is a <b>markup language</b> itself.	XML provides a <b>framework to define markup languages</b> .
3)	HTML is <b>not case sensitive</b> .	XML is <b>case sensitive</b> .
4)	HTML is a presentation language.	XML is neither a presentation language nor a programming language.

5)	HTML <b>has its own predefined tags.</b>	You <b>can define tags according to your need.</b>
6)	In HTML, it is <b>not necessary to use a closing tag.</b>	XML <b>makes it mandatory to use a closing tag.</b>
7)	HTML is <b>static</b> because it is used to display data.	XML is <b>dynamic</b> because it is used to transport data.
8)	HTML <b>does not preserve whitespaces.</b>	XML <b>preserve whitespaces.</b>

## XML Example

XML documents create a hierarchical structure looks like a tree so it is known as XML Tree that starts at "the root" and branches to "the leaves".

### Example of XML Document

XML documents uses a self-describing and simple syntax:

1. `<?xml version="1.0" encoding="ISO-8859-1"?>`
2. `<note>`
3. `<to>Tove</to>`
4. `<from>Jani</from>`
5. `<heading>Reminder</heading>`
6. `<body>Don't forget me this weekend!</body>`
7. `</note>`

The first line is the XML declaration. It defines the XML version (1.0) and the encoding used (ISO-8859-1 = Latin-1/West European character set).

The next line describes the root element of the document (like saying: "this document is a note"):

1. `<note>`

The next 4 lines describe 4 child elements of the root (to, from, heading, and body).

1. `<to>Tove</to>`
2. `<from>Jani</from>`

3. `<heading>`Reminder`</heading>`
4. `<body>`Don't forget me this weekend!`</body>`

And finally the last line defines the end of the root element.

1. `</note>`

XML documents must contain a **root element**. This element is "the parent" of all other elements.

The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree.

All elements can have sub elements (child elements).

1. `<root>`
2. `<child>`
3. `<subchild>`.....`</subchild>`
4. `</child>`
5. `</root>`

The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters).

All elements can have text content and attributes (just like in HTML).

## Another Example of XML: Books

*File: books.xml*

1. `<bookstore>`
2. `<book category="COOKING">`
3. `<title lang="en">`Everyday Italian`</title>`
4. `<author>`Giada De Laurentiis`</author>`
5. `<year>`2005`</year>`
6. `<price>`30.00`</price>`
7. `</book>`
8. `<book category="CHILDREN">`
9. `<title lang="en">`Harry Potter`</title>`
10. `<author>`J K. Rowling`</author>`
11. `<year>`2005`</year>`
12. `<price>`29.99`</price>`
13. `</book>`

14. `<book category="WEB">`
15. `<title lang="en">Learning XML</title>`
16. `<author>Erik T. Ray</author>`
17. `<year>2003</year>`
18. `<price>39.95</price>`
19. `</book>`
20. `</bookstore>`

The root element in the example is `<bookstore>`. All elements in the document are contained within `<bookstore>`.

The `<book>` element has 4 children: `<title>`, `<author>`, `<year>` and `<price>`.

## Another Example of XML: Emails

*File: emails.xml*

1. `<?xml version="1.0" encoding="UTF-8"?>`
2. `<emails>`
3. `<email>`
4. `<to>Vimal</to>`
5. `<from>Sonoo</from>`
6. `<heading>Hello</heading>`
7. `<body>Hello brother, how are you!</body>`
8. `</email>`
9. `<email>`
10. `<to>Peter</to>`
11. `<from>Jack</from>`
12. `<heading>Birth day wish</heading>`
13. `<body>Happy birth day Tom!</body>`
14. `</email>`
15. `<email>`
16. `<to>James</to>`
17. `<from>Jaclin</from>`
18. `<heading>Morning walk</heading>`
19. `<body>Please start morning walk to stay fit!</body>`
20. `</email>`
21. `<email>`
22. `<to>Kartik</to>`
23. `<from>Kumar</from>`
24. `<heading>Health Tips</heading>`

25. `<body>Smoking is injurious to health!</body>`
26. `</email>`
27. `</emails>`

## XML Attributes

XML elements can have attributes. By the use of attributes we can add the information about the element.

XML attributes enhance the properties of the elements.

*Note: XML attributes must always be quoted. We can use single or double quote.*

Let us take an example of a book publisher. Here, book is the element and publisher is the attribute.

1. `<book publisher="Tata McGraw Hill"></book>`

Or

1. `<book publisher='Tata McGraw Hill'></book>`

**Metadata should be stored as attribute and data should be stored as element.**

1. `<book>`
2. `<book category="computer">`
3. `<author> A & B </author>`
4. `</book>`

Data can be stored in attributes or in child elements. But there are some limitations in using attributes, over child elements.

---

## Why should we avoid XML attributes

- Attributes cannot contain multiple values but child elements can have multiple values.
- Attributes cannot contain tree structure but child element can.
- Attributes are not easily expandable. If you want to change in attribute's values in future, it may be complicated.
- Attributes cannot describe structure but child elements can.
- Attributes are more difficult to be manipulated by program code.

- Attributes values are not easy to test against a DTD, which is used to define the legal elements of an XML document.

---

## Difference between attribute and sub-element

In the context of documents, attributes are part of markup, while sub elements are part of the basic document contents.

In the context of data representation, the difference is unclear and may be confusing.

Same information can be represented in two ways:

### 1st way:

1. `<book publisher="Tata McGraw Hill"> </book>`

### 2nd way:

1. `<book>`
2. `<publisher> Tata McGraw Hill </publisher>`
3. `</book>`

In the first example publisher is used as an attribute and in the second example publisher is an element.

Both examples provide the same information but it is good practice to avoid attribute in XML and use elements instead of attributes.

## XML Namespaces

XML **Namespace** is used *to avoid element name conflict* in XML document.

## XML Namespace Declaration

An XML namespace is declared using the reserved XML attribute. This attribute name must be started with "xmlns".

Let's see the XML namespace syntax:

1. `<element xmlns:name = "URL">`

Here, namespace starts with keyword "**xmlns**". The word **name** is a namespace prefix. The **URL** is a namespace identifier.

Let's see the example of XML file.

1. `<?xml version="1.0" encoding="UTF-8"?>`
2. `<cont:contact xmlns:cont="http://sssit.org/contact-us">`
3. `<cont:name>Vimal Jaiswal</cont:name>`
4. `<cont:company>SSSIT.org</cont:company>`
5. `<cont:phone>(0120) 425-6464</cont:phone>`
6. `</cont:contact>`

**Namespace Prefix:** cont

**Namespace Identifier:** http://sssit.org/contact-us

It specifies that the element name and attribute names with cont prefix belongs to http://sssit.org/contact-us name space.

In XML, elements name are defined by the developer so there is a chance to conflict in name of the elements. To avoid these types of confliction we use XML Namespaces. We can say that XML Namespaces provide a method to avoid element name conflict.

Generally these conflict occurs when we try to mix XML documents from different XML application.

Let's take an example with two tables:

**Table1:**

1. `<table>`
2. `<tr>`
3. `<td>Aries</td>`
4. `<td>Bingo</td>`
5. `</tr>`
6. `</table>`

**Table2:** This table carries information about a computer table.

1. `<table>`
2. `<name>Computer table</name>`
3. `<width>80</width>`
4. `<length>120</length>`
5. `</table>`

If you add these both XML fragments together, there would be a name conflict because both have <table> element. Although they have different name and meaning.

---

# How to get rid of name conflict?

## 1) By Using a Prefix

You can easily avoid the XML namespace by using a name prefix.

1. `<h:table>`
2. `<h:tr>`
3. `<h:td>Aries</h:td>`
4. `<h:td>Bingo</h:td>`
5. `</h:tr>`
6. `</h:table>`
7. `<f:table>`
8. `<f:name>Computer table</f:name>`
9. `<f:width>80</f:width>`
10. `<f:length>120</f:length>`
11. `</f:table>`

*Note: In this example, you will get no conflict because both the tables have specific names.*

## 2) By Using xmlns Attribute

You can use xmlns attribute to define namespace with the following syntax:

1. `<element xmlns:name = "URL">`

Let's see the example:

1. `<root>`
2. `<h:table xmlns:h="http://www.abc.com/TR/html4/">`
3. `<h:tr>`
4. `<h:td>Aries</h:td>`
5. `<h:td>Bingo</h:td>`
6. `</h:tr>`
7. `</h:table>`
- 8.
9. `<f:table xmlns:f="http://www.xyz.com/furniture">`
10. `<f:name>Computer table</f:name>`
11. `<f:width>80</f:width>`
12. `<f:length>120</f:length>`
13. `</f:table>`
14. `</root>`



In the above example, the `<table>` element defines a namespace and when a namespace is defined for an element, the child elements with the same prefixes are associated with the same namespace.

1. `<root xmlns:h="http://www.abc.com/TR/html4/"`
2. `xmlns:f="http://www.xyz.com/furniture">`
3. `<h:table>`
4. `<h:tr>`
5. `<h:td>Aries</h:td>`
6. `<h:td>Bingo</h:td>`
7. `</h:tr>`
8. `</h:table>`
9. `<f:table>`
10. `<f:name>Computer table</f:name>`
11. `<f:width>80</f:width>`
12. `<f:length>120</f:length>`
13. `</f:table>`
14. `</root>`

*Note: The Namespace URI used in the above example is not necessary at all. It is not used by parser to look up information. It is only used to provide a unique name to the Namespace identifier.*

## Uniform Resource Identifier (URI)

Uniform Resource Identifier is used to identify the internet resource. It is a string of characters.

The most common URI is URL (Uniform Resource Locator) which identifies an internet domain address.

There is also an URI name URN (Universal Resource Name) but it is not so common. We have used only URL's in all our examples.

---

## The Default Namespace

The default namespace is used in the XML document to save you from using prefixes in all the child elements.

The only difference between default namespace and a simple namespace is that: There is no need to use a prefix in default namespace.

You can also use multiple namespaces within the same document just define a namespace against a child node.

Example of Default Namespace:

1. `<tutorials xmlns="http://www.javatpoint.com/java-tutorial">`
2. `<tutorial>`
3. `<title>Java-tutorial</title>`
4. `<author>Sonoo Jaiswal</author>`
5. `</tutorial>`
6. ...
7. `</tutorials>`

You can see that prefix is not used in this example, so it is a default namespace.

*Note: If you define a namespace without a prefix, all descendant elements are considered to belong to that namespace.*

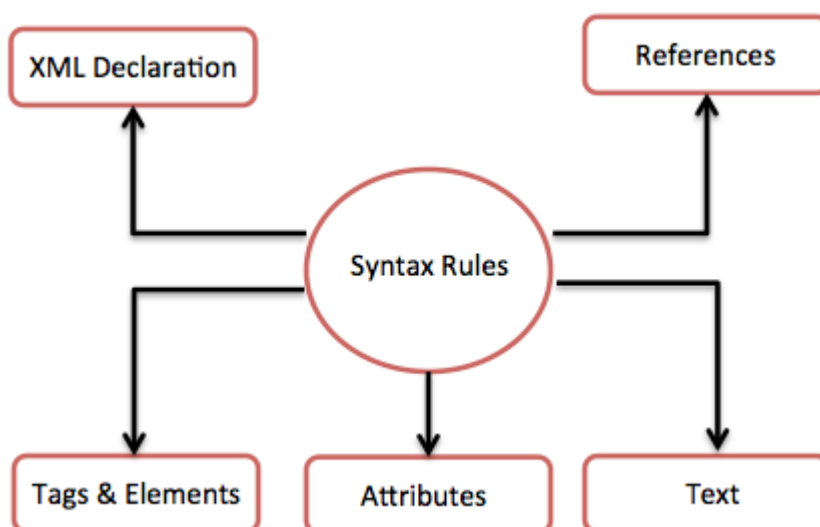
## Syntax Rule

```
<?xml version = "1.0"?>
<contact-info>
  <name>Tanmay Patil</name>
  <company>TutorialsPoint</company>
  <phone>(011) 123-4567</phone>
</contact-info>
```

You can notice there are two kinds of information in the above example –

- Markup, like `<contact-info>`
- The text, or the character data, *Tutorials Point* and *(040) 123-4567*.

The following diagram depicts the syntax rules to write different types of markup and text in an XML document.



Let us see each component of the above diagram in detail.

## XML Declaration

The XML document can optionally have an XML declaration. It is written as follows –

```
<?xml version = "1.0" encoding = "UTF-8"?>
```

Where *version* is the XML version and *encoding* specifies the character encoding used in the document.

## Syntax Rules for XML Declaration

- The XML declaration is case sensitive and must begin with "<?xml>" where "xml" is written in lower-case.
- If document contains XML declaration, then it strictly needs to be the first statement of the XML document.
- The XML declaration strictly needs be the first statement in the XML document.
- An HTTP protocol can override the value of *encoding* that you put in the XML declaration.

## Tags and Elements

An XML file is structured by several XML-elements, also called XML-nodes or XML-tags. The names of XML-elements are enclosed in triangular brackets < > as shown below –

```
<element>
```

## Syntax Rules for Tags and Elements

**Element Syntax** – Each XML-element needs to be closed either with start or with end elements as shown below –

```
<element>....</element>
```

or in simple-cases, just this way –

```
<element/>
```

**Nesting of Elements** – An XML-element can contain multiple XML-elements as its children, but the children elements must not overlap. i.e., an end tag of an element must have the same name as that of the most recent unmatched start tag.

The Following example shows incorrect nested tags –

```
<?xml version = "1.0"?>
<contact-info>
<company>TutorialsPoint
</contact-info>
</company>
```

The Following example shows correct nested tags –

```
<?xml version = "1.0"?>
<contact-info>
    <company>TutorialsPoint</company>
</contact-info>
```

**Root Element** – An XML document can have only one root element. For example, following is not a correct XML document, because both the **x** and **y** elements occur at the top level without a root element –

```
<x>...</x>
<y>...</y>
```

The Following example shows a correctly formed XML document –

```
<root>
  <x>...</x>
  <y>...</y>
</root>
```

**Case Sensitivity** – The names of XML-elements are case-sensitive. That means the name of the start and the end elements need to be exactly in the same case.

For example, **<contact-info>** is different from **<Contact-Info>**

## XML Attributes

An **attribute** specifies a single property for the element, using a name/value pair. An XML-element can have one or more attributes. For example –

```
<a href = "http://www.tutorialspoint.com/">Tutorialspoint!</a>
```

Here **href** is the attribute name and **http://www.tutorialspoint.com/** is attribute value.

### Syntax Rules for XML Attributes

- Attribute names in XML (unlike HTML) are case sensitive. That is, *href* and *href* are considered two different XML attributes.
- Same attribute cannot have two values in a syntax. The following example shows incorrect syntax because the attribute *b* is specified twice

–

```
<a b = "x" c = "y" b = "z">....</a>
```

- Attribute names are defined without quotation marks, whereas attribute values must always appear in quotation marks. Following example demonstrates incorrect xml syntax

–

```
<a b = x>....</a>
```

In the above syntax, the attribute value is not defined in quotation marks.

## XML References

References usually allow you to add or include additional text or markup in an XML document. References always begin with the symbol "&" which is a reserved character and end with the symbol ";". XML has two types of references –

- **Entity References** – An entity reference contains a name between the start and the end delimiters. For example **&amp;**; where *amp* is *name*. The *name* refers to a predefined string of text and/or markup.
- **Character References** – These contain references, such as **&#65;**, contains a hash mark (“#”) followed by a number. The number always refers to the Unicode code of a character. In this case, 65 refers to alphabet "A".

## XML Text

The names of XML-elements and XML-attributes are case-sensitive, which means the name of start and end elements need to be written in the same case. To avoid character encoding problems, all XML files should be saved as Unicode UTF-8 or UTF-16 files.

Whitespace characters like blanks, tabs and line-breaks between XML-elements and between the XML-attributes will be ignored.

Some characters are reserved by the XML syntax itself. Hence, they cannot be used directly. To use them, some replacement-entities are used, which are listed below –

Not Allowed Character	Replacement Entity	Character Description
<	&lt;	less than
>	&gt;	greater than
&	&amp;	ampersand
'	&apos;	apostrophe
"	&quot;	quotation mark

## XML DTD

### What is DTD

DTD stands for **Document Type Definition**. It defines the legal building blocks of an XML document. It is used to define document structure with a list of legal elements and attributes.

# Purpose of DTD

Its main purpose is to define the structure of an XML document. It contains a list of legal elements and define the structure with the help of them.

---

## Checking Validation

Before proceeding with XML DTD, you must check the validation. An XML document is called "well-formed" if it contains the correct syntax.

A well-formed and valid XML document is one which have been validated against DTD.

Visit <http://www.xmlvalidation.com> to validate the XML file.

---

## Valid and well-formed XML document with DTD

Let's take an example of well-formed and valid XML document. It follows all the rules of DTD.

*employee.xml*

1. `<?xml version="1.0"?>`
2. `<!DOCTYPE employee SYSTEM "employee.dtd">`
3. `<employee>`
4. `<firstname>vimal</firstname>`
5. `<lastname>jaiswal</lastname>`
6. `<email>vimal@javatpoint.com</email>`
7. `</employee>`

In the above example, the DOCTYPE declaration refers to an external DTD file. The content of the file is shown in below paragraph.

*employee.dtd*

1. `<!ELEMENT employee (firstname,lastname,email)>`
2. `<!ELEMENT firstname (#PCDATA)>`
3. `<!ELEMENT lastname (#PCDATA)>`
4. `<!ELEMENT email (#PCDATA)>`

# Description of DTD

**<!DOCTYPE employee** : It defines that the root element of the document is employee.

**<!ELEMENT employee**: It defines that the employee element contains 3 elements "firstname, lastname and email".

**<!ELEMENT firstname**: It defines that the firstname element is #PCDATA typed. (parse-able data type).

**<!ELEMENT lastname**: It defines that the lastname element is #PCDATA typed. (parse-able data type).

**<!ELEMENT email**: It defines that the email element is #PCDATA typed. (parse-able data type).

## XML DTD with entity declaration

A doctype declaration can also define special strings that can be used in the XML file.

An entity has three parts:

1. An ampersand (&)
2. An entity name
3. A semicolon (;)

Syntax to declare entity:

1. `<!ENTITY entity-name "entity-value">`

Let's see a code to define the ENTITY in doctype declaration.

author.xml

1. `<?xml version="1.0" standalone="yes" ?>`
2. `<!DOCTYPE author [`
3. `<!ELEMENT author (#PCDATA)>`
4. `<!ENTITY sj "Sonoo Jaiswal">`
5. `]>`
6. `<author>&sj;</author>`

In the above example, sj is an entity that is used inside the author element. In such case, it will print the value of sj entity that is "Sonoo Jaiswal".

*Note: A single DTD can be used in many XML files.*

# XML Schema

## What is XML schema

XML schema is a language which is used for expressing constraint about XML documents. There are so many schema languages which are used now a days for example Relax- NG and XSD (XML schema definition).

An XML schema is used to define the structure of an XML document. It is like DTD but provides more control on XML structure.

---

## Checking Validation

An XML document is called "well-formed" if it contains the correct syntax. A well-formed and valid XML document is one which have been validated against Schema.

Visit <http://www.xmlvalidation.com> to validate the XML file against schema or DTD.

---

## XML Schema Example

Let's create a schema file.

*employee.xsd*

1. `<?xml version="1.0"?>`
2. `<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"`
3. `targetNamespace="http://www.javatpoint.com"`
4. `xmlns="http://www.javatpoint.com"`
5. `elementFormDefault="qualified">`
- 6.
7. `<xs:element name="employee">`
8. `<xs:complexType>`
9. `<xs:sequence>`
10. `<xs:element name="firstname" type="xs:string"/>`
11. `<xs:element name="lastname" type="xs:string"/>`
12. `<xs:element name="email" type="xs:string"/>`
13. `</xs:sequence>`
14. `</xs:complexType>`
15. `</xs:element>`



16.

17. `</xs:schema>`

Let's see the xml file using XML schema or XSD file.

*employee.xml*

1. `<?xml version="1.0"?>`
2. `<employee`
3. `xmlns="http://www.javatpoint.com"`
4. `xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`
5. `xsi:schemaLocation="http://www.javatpoint.com employee.xsd">`
- 6.
7. `<firstname>vimal</firstname>`
8. `<lastname>jaiswal</lastname>`
9. `<email>vimal@javatpoint.com</email>`
10. `</employee>`

## Description of XML Schema

`<xs:element name="employee">` : It defines the element name employee.

`<xs:complexType>` : It defines that the element 'employee' is complex type.

`<xs:sequence>` : It defines that the complex type is a sequence of elements.

`<xs:element name="firstname" type="xs:string"/>` : It defines that the element 'firstname' is of string/text type.

`<xs:element name="lastname" type="xs:string"/>` : It defines that the element 'lastname' is of string/text type.

`<xs:element name="email" type="xs:string"/>` : It defines that the element 'email' is of string/text type.

## XML Schema Data types

There are two types of data types in XML schema.

1. simpleType
2. complexType

### simpleType

The simpleType allows you to have text-based elements. It contains less attributes, child elements, and cannot be left empty.

## complexType

The complexType allows you to hold multiple attributes and elements. It can contain additional sub elements and can be left empty.

# JSON Tutorial



## What is JSON

- JSON stands for JavaScript Object Notation.
- JSON is lightweight data-interchange format.
- JSON is easy to read and write than XML.
- JSON is language independent.
- JSON supports array, object, string, number and values.

## JSON Example

In this tutorial, you will get a lot of JSON examples to understand the topic well. The JSON file must be save with .json extension. Let's see a simple JSON example.

*File: first.json*

```
1. {"employees":[  
2.   {"name":"Sonoo", "email":"sonoojaiswal1987@gmail.com"},  
3.   {"name":"Rahul", "email":"rahul32@gmail.com"},  
4.   {"name":"John", "email":"john32bob@gmail.com"}  
5. ]}
```

# What is JSON

JSON is an acronym for JavaScript Object Notation, is an open standard format, which is lightweight and text-based, designed explicitly for human-readable data interchange. It is a language-independent data format. It supports almost every kind of language, framework, and library.

In the early 2000s, JSON was initially specified by Douglas Crockford. In 2013, JSON was standardized as ECMA-404, and RFC 8259 was published in 2017.

JSON is an open standard for exchanging data on the web. It supports data structures like objects and arrays. So, it is easy to write and read data from JSON.

In JSON, data is represented in key-value pairs, and curly braces hold objects, where a colon is followed after each name. The comma is used to separate key-value pairs. Square brackets are used to hold arrays, where each value is comma-separated.

## What is JSON

- JSON stands for JavaScript Object Notation.
- JSON is an open standard data-interchange format.
- JSON is lightweight and self-describing.
- JSON originated from JavaScript.
- JSON is easy to read and write.
- JSON is language independent.
- JSON supports data structures such as arrays and objects.

## Features of JSON

- Simplicity
- Openness
- Self-Describing
- Internationalization
- Extensibility
- Interoperability

## Why do we use JSON?

Since JSON is an easy-to-use, lightweight language data interchange format in comparison to other available options, it can be used for API integration. Following are the advantages of JSON:

- **Less Verbose:** In contrast to XML, JSON follows a compact style to improve its users' readability. While working with a complex system, JSON tends to make substantial enhancements.
- **Faster:** The JSON parsing process is faster than that of the XML because the DOM manipulation library in XML requires extra memory for handling large XML files. However, JSON requires less data that ultimately results in reducing the cost and increasing the parsing speed.
- **Readable:** The JSON structure is easily readable and straightforward. Regardless of the programming language that you are using, you can easily map the domain objects.
- **Structured Data:** In JSON, a map data structure is used, whereas XML follows a tree structure. The key-value pairs limit the task but facilitate the predictive and easily understandable model.

## JSON Data Types

Following are the most commonly used JSON data types.

Data Type	Description	Example
String	A string is always written in double-quotes. It may consist of numbers, alphanumeric and special characters.	"student", "name", "1234", "Ver_1"
Number	Number represents the numeric characters.	121, 899
Boolean	It can be either True or False.	true
Null	It is an empty value.	

## JSON Objects

In JSON, objects refer to dictionaries, which are enclosed in curly brackets, i.e., { }. These objects are written in key/value pairs, where the key has to be a string and values have to be a valid JSON data type such as string, number, object, Boolean or null. Here the key and values are separated by a colon, and a comma separates each key/value pair.

For example:

1. {"name" : "Jack", "employeeid" : 001, "present" : false}

## JSON Arrays

In JSON, arrays can be understood as a list of objects, which are mainly enclosed in square brackets [ ]. An array value can be a string, number, object, array, boolean or null.

For example:

1. [{
2. "PizzaName" : "Country Feast",
3. "Base" : "Cheese burst",
4. "Toppings" : ["Jalepenos", "Black Olives", "Extra cheese", "Sausages", "Cherry tomatoes"],
5. "Spicy" : "yes",
6. "Veg" : "yes"
7. },
- 8.
9. {
10. "PizzaName" : "Veggie Paradise",
11. "Base" : "Thin crust",
12. "Toppings" : ["Jalepenos", "Black Olives", "Grilled Mushrooms", "Onions", "Cherry tomatoes"],
13. "Spicy" : "yes",
14. "Veg" : "yes"
15. }
16. ]

In the above example, the object "Pizza" is an array. It contains five objects, i.e., PizzaName, Base, Toppings, Spicy, and Veg.

Output:

```
[
  {
    "PizzaName": "Country Feast",
    "Base": "Cheese burst",
    "Toppings": [
      "Jalepenos",
      "Black Olives",
      "Extra cheese",
      "Sausages",
      "Cherry tomatoes"
    ],
    "Spicy": "yes",
    "Veg": "yes"
  },
  {
    "PizzaName": "Veggie Paradise",
    "Base": "Thin crust",
    "Toppings": [
      "Jalepenos",
      "Black Olives",
      "Grilled Mushrooms",
      "Onions",
      "Cherry tomatoes"
    ],
    "Spicy": "yes",
    "Veg": "yes"
  }
]
```

## JSON Vs XML

JSON stands for JavaScript Object Notation, whereas XML stands for Extensive Markup Language. Nowadays, JSON and XML are widely used as data interchange formats, and both have been acquired by applications as a technique to store structured data.

### Difference between JSON and XML

JSON is easy to learn.	XML is quite more complex to learn than JSON.
It is simple to read and write.	It is more complex to read and write than JSON.
It is data-oriented.	It is document-oriented.
JSON is less secure in comparison to XML.	XML is highly secured.

It doesn't provide display capabilities.	It provides the display capability because it is a markup language.
It supports the array.	It doesn't support the array
<p>Example :</p> <pre>[ { "name" : "Peter", "employed id" : "E231", "present" : true, "numberofdayspresent" : 29 }, { "name" : "Jhon", "employed id" : "E331", "present" : true, "numberofdayspresent" : 27 } ]</pre>	<p>Example :</p> <pre>&lt;name&gt; &lt;name&gt;Peter&lt;/name&gt; &lt;/name&gt;</pre>

## JSON vs XML

Before knowing about the differences between JSON and XML, we should be aware of the definition of json and xml.

### What is json?

JSON stands for **JavaScript object notation**. JSON has been derived from javascript, where javascript is a programming language. It was originally created to hold the structured data that could be used in javascript. JSON became so popular that it is used for data for all kinds of applications. It is the most popular way of sending the data for Web APIs.

#### Basic data types supported by json are:

- **Strings:** Characters that are enclosed in single or double quotation marks.
- **Number:** A number could be integer or decimal, positive or negative.
- **Booleans:** The Boolean value could be either true or false without any quotation marks.
- **Null:** Here, null means nothing without any quotation marks.

In addition to basic data types, json has arrays and objects.

## Arrays

Arrays are the lists that are represented by the square brackets, and the values have commas in between them. They can contain mix data types, i.e., a single array can have strings, Boolean, numbers.

**For example:**

**Example 1:** [1, 2, 7.8, 5, 9, 10];

**Example 2:** ["red", "yellow", "green"];

**Example 3:** [8, "hello", null, true];

In the above, example 1 is an array of numbers, example 2 is an array of strings, and example 3 is an array of mix data types.

## Objects

Objects are JSON dictionaries that are enclosed in curly brackets. In objects, keys and values are separated by a colon ':', pairs are separated by comma. Keys and values can be of any type, but the most common type for the keys is a string.

For example: {"red" : 1, "yellow" : 2, "green" : 3};

## Nesting

Nesting involves keeping the arrays and objects inside of each other. We can put the arrays inside objects, objects inside arrays, arrays inside arrays, etc. We can say that json file is a big object with lots of objects and arrays inside.

**Example:**

```
1. {  
2.   "song" :  
3.     {  
4.       "title" : "Hey Dude";  
5.       "artist": "The Beatles";  
6.       "musicians": ["John Lennon", "Paul McCratney", "Ringo Starr"];  
7.     }  
8. }
```

In the above code, the song starts with a curly bracket. Therefore, a song is an object. It contains three key-value pairs wherein title, artist and musicians are the keys.'



# What is XML?

XML stands for an extensible markup language. It is like HTML, where HTML stands for Hypertext Markup language. HTML is used for creating websites, whereas XML can be used for any kind of structured data.

XML has two ways of handling data, i.e., Tags and Attributes. The tags work as HTML. The start tags start with the `<_>` and end with the `</_>`. The start and end tags must match. The names must only be letters, numbers, and underscore, and the tag name must start with a letter only.

## For example:

```
<title> Hello World </title>
```

## Nested Tags

When we put the tag inside of another tag that creates the nested data.

## For example:

1. `<color>`
2. `<red> 1 </red>`
3. `<yellow> 2 </yellow>`
4. `<green> 3 </green>`
5. `</color>`

As we can observe in the above code that inside the color tag, we have three more tags, i.e., red, yellow, and green.

## Similarities between the json and XML.

- **Self-describing:** Both json and xml are self-describing as both xml data and json data are human-readable text.
- **Hierarchical:** Both json and xml support hierarchical structure. Here hierarchical means that the values within values.
- **Data interchange format:** JSON and XML can be used as data interchange formats by many different programming languages.
- **Parse:** Both the formats can be easily parsed.
- **Retrieve:** Both formats can be retrieved by using HTTP requests. The methods used for retrieving the data are GET, PUT, POST.

## Differences between the json and XML.



**The following are the differences between the json and xml:**

JSON	XML
JSON stands for javascript object notation.	XML stands for an extensible markup language.
The extension of json file is .json.	The extension of xml file is .xml.
The internet media type is application/json.	The internet media type is application/xml or text/xml.
The type of format in JSON is data interchange.	The type of format in XML is a markup language.
It is extended from javascript.	It is extended from SGML.
It is open source means that we do not have to pay anything to use JSON.	It is also open source.
The object created in JSON has some type.	XML data does not have any type.
The data types supported by JSON	XML data is in a string format.

are strings, numbers, Booleans, null, array.	
It does not have any capacity to display the data.	XML is a markup language, so it has the capacity to display the content.
JSON has no tags.	XML data is represented in tags, i.e., start tag and end tag.
	XML file is larger. If we want to represent the data in XML then it would create a larger file as compared to JSON.
JSON is quicker to read and write.	XML file takes time to read and write because the learning curve is higher.
JSON can use arrays to represent the data.	XML does not contain the concept of arrays.
It can be parsed by a standard javascript function. It has to be parsed before use.	XML data which is used to interchange the data, must be parsed with respective to their programming language to use that.
It can be easily parsed and little bit code is required to parse the data.	It is difficult to parse.
File size is smaller as compared to XML.	File size is larger.
JSON is data-oriented.	XML is document-oriented.
It is less secure than XML.	It is more secure than JSON.