UNIT-I

➢ Introduction to Internet:

The Internet is the global system of interconnected computer networks that use the Internet protocol suite to link devices worldwide. The purpose of the internet is to communicate between computers that are interconnected with each other. Internet is accessible to every user all over the world.

It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies.

The Internet carries a vast range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), electronic mail, telephony and file sharing.

Browser is a tool used to access the internet using WWW (World Wide Web) and HTTP (Hyper Text Transfer Protocol). In the browser, if the user types the domain name such as www.tn.gov.in, the browser calls a protocol name DNS (Domain Name Server). DNS is used to get the IP address of the domain names.

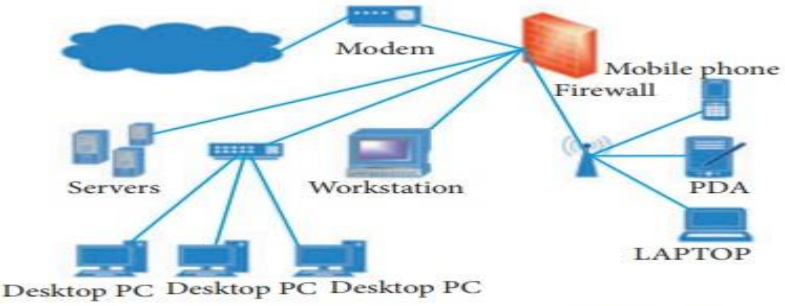


Figure 15.1 Block Diagram of Internet

Evolution of Internet:

Internet evolved in 1969 and evolved many changes in several technologies and Infrastructural levels.

• Internet was started by ARPANET (Advanced Research Project Agency Network), developed by United States. Department of Defence for communication among different government bodies, initially with four nodes.

• In 1972, the four nodes has been developed and it grown to 23 nodes located in different countries making it Internet.

• Invented TCP/IP protocols, DNS, WWW, browsers scripting languages.

• Internet is used as a medium to publish and access the information

• In 1985, The NSFNET was composed of multiple regional networks and peer networks

• In 1986, the NSFNET created a three-tiered network architecture.

• In 1988, updated the links to make it faster

• In 1990, Merit, IBM, and MCI started a new organization known as Advanced Network and Services (ANS).

• By 1991, data traffic had increased tremendously, which necessitated upgrading the NSFNET's backbone network service to T3 (45 Mbps) links.

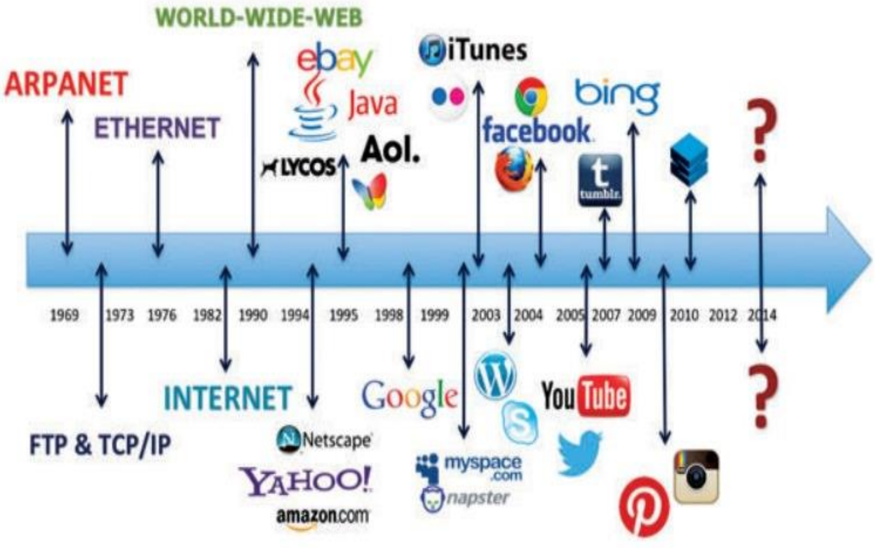


Figure 15.2 Internet History Timeline

Internet Evalution:

Internet covers almost every aspect of life. Internet allows the users to communicate with the people sitting at remote locations. There are various applications available on the web that uses Internet as a medium for communication. One can find various social networking sites such: Facebook, Twitter, Yahoo, Google+, Flickr, and Orkut. One can surf for any kind of information over the internet. Information regarding various topics such as Technology, Health and Science, Social Studies, Geographical Information, Information Technology and Products can be surfed with help of a search engine.

Apart from communication and source of information, internet also serves as a medium for entertainment. Internet also allows the users to use as many services as like E-mail, Internet Banking, Online Shopping, Online Ticket Booking, Online Bill Payment and Data Sharing. Internet provides concept of electronic commerce that allows the business deals to be conducted on electronic systems.

Hardware and Software Requirements for Internet connection:

The following are the methods of connecting a computer to the Internet using software and hardware peripherals.

Three

- Connecting a computer using Wireless Broadband
- Connecting a computer using an Ethernet Cable
- Connecting a Computer Using Dial-Up Community

Hardware Requirement:

- To connect the Internet, any one of the following is mandatory.
- Modem is used to connect Internet through Telephone connection.
- NIC- Network Interface Card(wired/ wireless) facility is the most important hardware required to connect Internet. For example, the Laptop can be connected Internet through the wired/wireless.
- Dongle is used to connect the Internet using cellular network
- Wi-Fi router or Hotspot is used to connect the Internet using wireless network
- Electronic device which supports cellular network
- Internet Connectivity such as Dial-up connection, ISDN, DSL, Cable TV, wired and wireless (Cellular) Network.

Software Requirement

- The operating system should support TCP (Transfer Control Protocol) / IP (Internet Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), HTTP (Hyper Text Transfer Protocol) and HTTPS (Hyper Text Transfer Protocol Secured) protocols.
- Browsers and other Internet clients access to the web applications such as Outlook, Gmail, Whatsapp, Facebook, Twitter and etc.

Connection Types:

The following methods are able to connect internet.

Dial-up Connection :

A dial-up connection is established when two or more data communication devices use a Public Switched Telephone Network (PSTN) to connect to an Internet Service Provider (ISP) from computers. Many remote locations depend on Internet dial-up connections because broadband and cable are rare in remote areas with low population. Internet Service Providers often provide dial-up connections, a feasible alternative for budget-conscious subscribers.

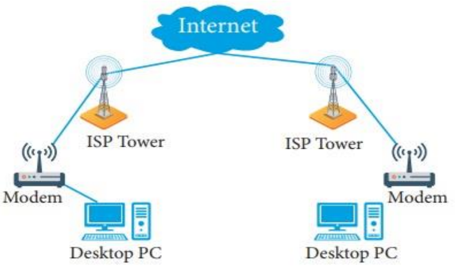


Figure 15.3 Dial-up Connection

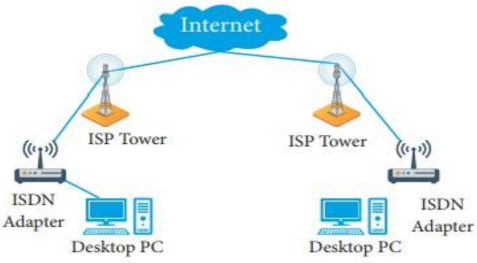


Figure 15.4a Integrated Services Digital Network

ISDN

ISDN is the acronym of Integrated Services Digital Network. It establishes the connection using the phone lines (PSTN) which carry digital signals instead of analog signals. It is a set of communication standards for simultaneous digital transmission of data, voice, video, and other services over the traditional circuits of the public switched telephone network. There are two techniques to deliver ISDN services such as Basic Rate Interface (BRI) and Primary Rate Interface (PRI).

The following diagram shows accessing internet using ISDN connection:

DSL:

Digital Subscriber Line (DSL) is a high-speed Internet service for homes and businesses that competes with cable and other forms of broadband Internet. DSL provides high-speed networking over ordinary Telephone lines using broadband modem technology. The technology behind DSL enables Internet and telephone service to work over the same phone line without requiring customers to disconnect either their Voice or Internet connections.

Cable TV Internet Connection (setup box):

The cable TV network can be used for connecting a computer or a local network to the Internet, competing directly with DSL (Digital Subscriber Line) technology.

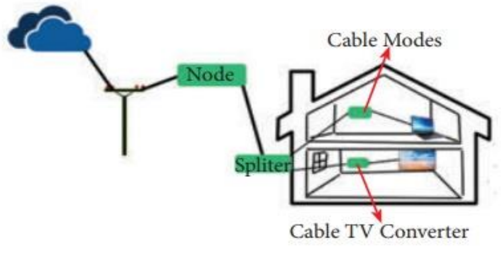


Figure 15.4b Cable TV Connection

This type of network is classified as HFC (Hybrid Fiber-Coaxial), as it uses both fiber optics and coaxial cables. The connection between the cable TV company to the distribution points (Optical nodes) is made using fiber optics, with distances up to 25 miles (40 km). Each optical node is typically serves between 500 and 2,000 clients (customers).

The following diagram shows that how internet is accessed using Cable TV connection:

Satellite Internet Connection:

Satellite Internet access is Internet access provided through satellite communication for domestic and enterprise usage. The facility of modern consumer grade satellite Internet service is typically provided to individual users through geostationary satellites. It provides fairly high data speeds, along with latest satellites using Ka-band to attain downstream data speeds up to 50 Mbps internet speed.

Wireless Internet Connection:

It is a technology for wireless local area networking with devices based on the IEEE 802.11 standards. Devices that can use Wi-Fi technology include personal computers, video-game consoles, phones and tablets, digital cameras, smart TVs, digital audio players and modern printers. Wi-Fi compatible devices can connect to the Internet via a WLAN and a wireless access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range of outdoors. Hotspot coverage can be as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points.

Services Available on the Internet:

- Data Transfer
- Internet banking
- E-commerce
- E-Learning
- E-Governance
- Browsing and Chating
- E-Mail

➢ Browser Architecture-IE

Web Browser web Browser is an application software that allows us to view and explore information on the web. User can request for any web page by just entering a URL into address bar.

Web browser can show text, audio, video, animation and more. It is the responsibility of a web browser to interpret text and commands contained in the web page.

Earlier the web browsers were text-based while now days graphical-based or voice-based web browsers are also available. Following are the most common web browser available today:

Browser	Vendor
Internet Explorer	Microsoft
Google Chrome	Google
Mozilla Firefox	Mozilla
Netscape Navigator	Netscape Communications Corp.
Opera	Opera Software
Safari	Apple
Sea Monkey	Mozilla Foundation
K-meleon	K-meleon

Architecture

There is a lot of web browser available in the market. All of them interpret and display information on the screen however their capabilities and structure varies depending upon implementation. But the most basic component that all web browsers must exhibit are listed below:

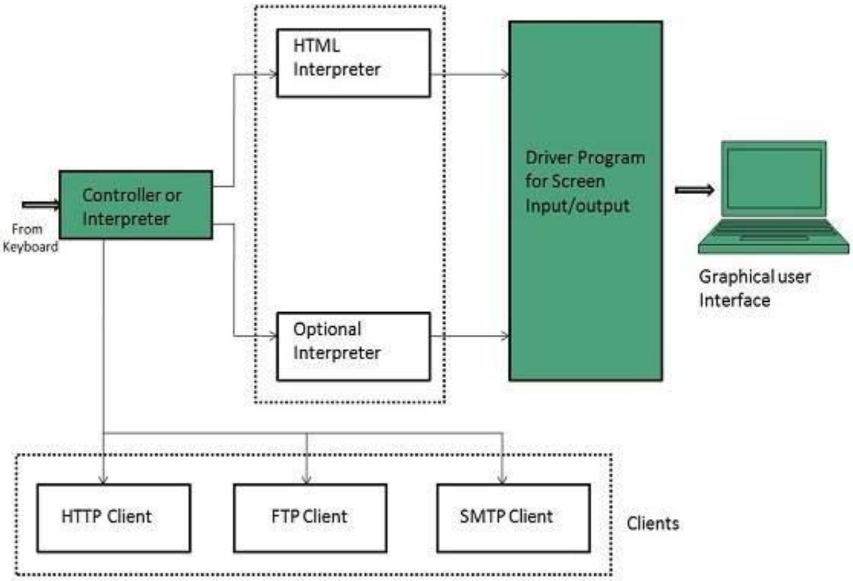
- Controller/Dispatcher
- Interpreter
- Client Programs

Controller works as a control unit in CPU. It takes input from the keyboard or mouse, interpret it and make other services to work on the basis of input it receives.

Interpreter receives the information from the controller and executes the instruction line by line. Some interpreter are mandatory while some are optional For example, HTML interpreter program is mandatory and java interpreter is optional.

Client Program describes the specific protocol that will be used to access a particular service. Following are the client programs that are commonly used:

- HTTP
- SMTP
- FTP
- NNTP
- POP



Starting Internet Explorer

Internet explorer is a web browser developed by Microsoft. It is installed by default with the windows operating system however, it can be downloaded and be upgraded.

To start internet explorer, follow the following steps:

- Go to Start button and click Internet Explorer.

The Internet Explorer window will appear as shown in the following diagram:

Accessing Web Page

Accessing web page is very simple. Just enter the URL in the address bar as shown the following diagram:

Navigation

A web page may contain hyperlinks. When we click on these links other web page is opened. These hyperlinks can be in form of text or image. When we take the mouse over an hyperlink, pointer change its shape to hand.

Key Points

- In case, you have accessed many web pages and willing to see the previous webpage then just click back button.
- You can open a new web page in the same tab, or different tab or in a new window.

Saving Webpage

You can save web page to use in future. In order to save a webpage, follow the steps given below:

- Click File > Save As. Save Webpage dialog box appears.
- Choose the location where you want to save your webpage from save in: list box. Then choose the folder where you want to save the webpage.
- Specify the file name in the File name box.
- Select the type from Save as type list box.
  - Webpage, complete
  - Web Archive
  - Webpage HTML only
  - Text File
- From the encoding list box, choose the character set which will be used with your webpage. By default, Western European is selected.
- Click save button and the webpage is saved.

Saving Web Elements

Web elements are the pictures, links etc. In order to save these elements follow the steps given below:

- Right click on the webpage element you want to save. Menu options will appear. These options may vary depending on the element you want to save.

Evolution of the Web Browsers

Tim Berner Lee was the founder of the Web Browser as a concept developed in 1990. The initial name of his browser was World Wide Web but now it has become Nexus. Erwise was the first graphical user interface web browser. Robert Cailliau was the man behind this development.

Marc Andreessen developed the browser further in 1993 by releasing one of the most popular browsers of all time – Mosaic. The WWW system became more user-friendly and accessible after this. Andresen lead company Netscape reached its boom in 1994 with 90% of the users using it.

Internet Explorer by Microsoft came out in 1995 giving major competition to Mosaic. The upper advantage of windows in internet explorer led to increase in its market share to 95% by 2002. Opera came out in 1996 but did not manage to get a stronghold in the market by only having a 2% share by 2012.

Netscape's Mozilla foundation led to the introduction of the Firefox browser in 1998 that grew to 27% of the market share in 2011. Apple Safari was late in the market by coming out in 2003 but yet it is the dominant browser for Ios users around the globe.

Chrome which accounts for the majority of the users in the world came out in 2008 but its usage only increased with every passing year. By 2014, 45% of the internet users were using google chrome as their go-to web browser.

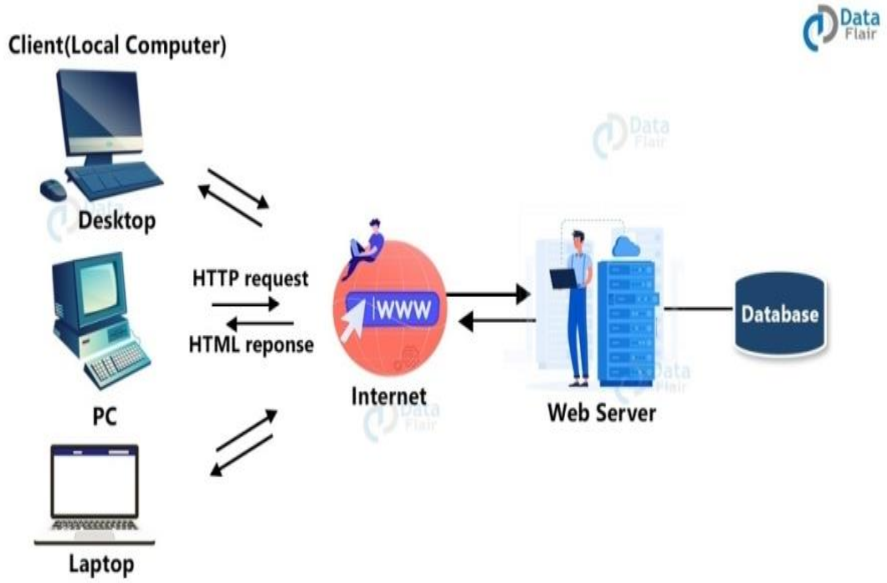
How the World Wide Web Works?

The clients server format enables the basic functioning of the web. The users request information on the web pages and the server works on transferring it to them. Software to serve these pages to the users is a web server. And the user's computer here becomes the client and the browser on this allows document retrieval.

All websites are on this server as guests and every time there is a request, the server pays the hosting price. The second users type an URL on the search, www begins to work. The three key elements here are Hypertext Markup Language (HTML), Hypertext Transfer Protocol (HTTP) and Web browsers.

How does a web browser work?

Like mentioned in the introduction, browser allows sending and receiving of information from all parts of the web. The Hypertext Transfer Protocol allows data transfer while governing all pieces of content. The user interface is what enables graphical representation of the data and allows users to explore it.



The backend of the browsers are the one handling requests and becomes the carrier for information transfer and interaction. While the clean front end allows users to scroll and browse. The backend supports it completely with handling all the functions smoothly. This is how the web browsers function.

Components of a Web Browser

1. User Interface

It is an environment allowing users to use certain features like search bar, refresh button, menu, bookmarks, etc.

2. Browser Engine

The bridge connects the interface and the engine. It monitors the rendition engine while manipulating the inputs coming from multiple user interfaces.

3. Networking

The protocol provides an URL and manages all sorts of safety, privacy and communication. In addition, the store network traffic gets saved in retrieved documents.

4. Data Storage

The cookies store information as the data store is an uniform layer that the browsers use. Storage processes like IndexedDB, WebSQL, localStorage, etc works well on browsers.

5. JavaScript Interpreter

It allows conversion of JavaScript code in a document and the executes it. Then the engine shows the translation on the screen to the users.

Architecture of Web Browser

The architecture of a web browser has certain components. They are Controller/Dispatcher, Interpreter, and Client Programs. The control unit in a CPU is a controller that takes in the input, interprets it, and then instructs the device to work in a certain way.

The interpreter receives this info from the controller to get ahead in the given task step by step. Lastly, the client program has sets of protocols to complete a particular service. Some common names are – HTTP, SMTP, FTP, etc.

Websites

As mentioned above, the location on a web browser using a URL opens up a website. They are of two types –

1. Static Websites – These are flat or stationary websites made by the client that only they can change. The users can read and watch them but cannot interact with the content in any way. They use HTML language to create as the information is fixed.

2. Dynamic Websites – These websites keep updating and the users will find new information every time. The ajax technology allows users to change information as they want to make it very dynamic.

Web Pages

The web pages are the places where users find all sorts of information. The hyperlinks to many other pages allowing users to switch and explore. And a collection of these pages make up a website. Web pages are also of two types like websites – Static and Dynamic. The brief for them is the same as above.

Under dynamic web pages, they can be either Server-side dynamic web pages or Client-side dynamic web pages. The server-side dynamic page has server-side scripting while the other one has client-side scripting using the Document Object Model. The scripting has become quite common nowadays and some of them are –

Client-Side Web Scripting

1. JavaScript – prototype-based scripting that inherits its name from java and is saved as a .js extension.
2. ActionScript – an object-oriented language used to target adobe flash player for website development.
3. Dart – source to source compiler and is the open-source language by Google.
4. VBScript – open source web language by Microsoft that has static typing class-based object programming.

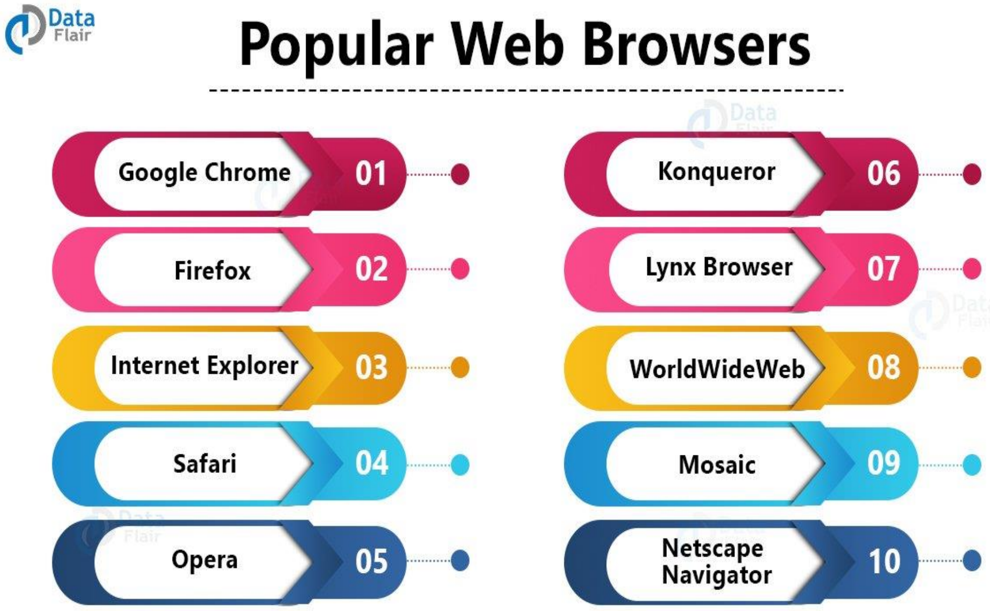
Server-side Web Scripting

1. Active Server Pages – supports Component Object Model that uses functions of DLL.
2. ActiveVFP – uses native Visual Foxpro language and database for website creation
3. ASP.net – develops websites, web applications, and web services
4. Java – Java code with byte code creates the website
5. Python – object-oriented and functional programming for website creation
6. WebDNA – uses an embedded database system

Features and Functions of Web Browser

- Giving users access to all sorts of information is the primary function of a web browser that includes retrieval, display, and navigation.
- The new browsers are very minimal with an easy user interface with support of HTML opening all file formats and protocols in one place.
- There are internet suites to support email, Usenet news, and Internet Relay Chat and they are often not the same as web browsers.
- Users can open several information resources at the same time in the same window or different windows.
- They also block the redirect to prevent users from reaching unwanted windows and websites without their consent.
- There are options for users to bookmark the web pages so it’s easier and quicker to access in the future. These bookmarks are the user’s “favorites” and most of the browsers have this as an inbuilt feature.
- The user interfaces is mostly similar in all browsers with a refresh or reload button, stop button, an address bar, a search bar, a status bar, a viewport, and other incremental find features on the web page.
- There are a few devices with photos of LG Smart TV on-screen keyboard with a magic motion remote.
- There is an option to delete web history, cookies, and cache ensuring security as well as privacy of the user.
- Before there was only HTML support but the new browsers have HTML and XHTML making the process smoother.
- There is a browser extension as well for extending the web browser’s functionality.
- The web browsers have a user interface, layout engine, scripting, interpreter, user interface backend, networking, and data.

Popular Web Browsers



1. Google Chrome
Google Chrome has the most number of users in the world. It came out very late but its effective results and a large pool of information attracted users to it making google the most preferred search engine. This works well on most of the operating systems and devices becoming easily accessible as well. 89.0.4389.105 is the latest version of this browser.

2. Firefox
One of the widely used web browsers, it supports all sorts of operating systems becoming very flexible for the users. And because it is open-source, a large community of developers uses it regularly. There are many plugins/add-ons allowing customization. It is a Mozilla foundation product and the latest version is Firefox

3. Internet Explorer
A windows operating system browser, internet explorer comes preinstalled in all Microsoft computers. The design is a way to improve the user interface so they can open multiple web pages at the same time within the operating system The latest versions are Internet Explorer 7 with 8 in beta.

4. Safari
Being an Apple product, safari works only on Ios devices like iPhone, MacBook, iPad, etc. This browser comes pre-installed on these devices as other browsers cannot function well on Apple devices. The latest version of this browser is macOS 11 Big Sur launched in 2021.

5. Opera
Opera is a commonly used mobile browser with magnification and gesture features to handle internet-related tasks. It has inbuilt malware and phasing security with strong encryption to secure user’s and client’s personal data and information. Opera works well on Windows, OS X, and Linux.

6. Konqueror
An open-source browser that supports all basic file management on local UNIX to advanced remote and local network file browsing. It is not the most preferred option by the users due to lack of user-friendliness and results

7. Lynx Browser
Again not a very famous name, it is a World Wide Web browser for Umic and VMS users. They use it to run cursor-addressable, terminals or emulators.

8. WorldWideWeb
The first web browser came out in 1990 but now its allied Nexus. This is to avoid the misconception between www as a concept and nexus as a browser. Not used very commonly due to less interactive interface and no bookmark option.

9. Mosaic
The second web browser that came out in 1993 with an improved interface and was by the National Center for Supercomputing Applications. Marc Andersson being the team leader made it very popular.
10. Netscape Navigator
This browser came out in 1994 and became the market leader for good 10 years. Netscape kept improving by launching better versions with license schemes and free usage.

Browser	Vendor
Internet Explorer	Microsoft
Google Chrome	Google
Mozilla Firefox	Mozilla
Netscape Navigator	Netscape Communications Corp.
Opera	Opera Software
Safari	Apple
Sea Monkey	Mozilla Foundation
K-meleon	K-meleon

➢ Chrome-Search Engine Introduction

Search Engine refers to a huge database of internet resources such as web pages, newsgroups, programs, images etc. It helps to locate information on World Wide Web.

User can search for any information by passing query in form of keywords or phrase. It then searches for relevant information in its database and return to the user.

Search Engine Components
Generally there are three basic components of a search engine as listed below:
1. Web Crawler
2. Database
3. Search Interfaces
Web crawler
It is also known as spider or bots. It is a software component that traverses the web to gather information.
Database
All the information on the web is stored in database. It consists of huge web resources.
Search Interfaces
This component is an interface between user and the database. It helps the user to search through the database.

Search Engine Working

Web crawler, database and the search interface are the major component of a search engine that actually makes search engine to work. Search engines make use of Boolean expression AND, OR, NOT to restrict and widen the results of a search. Following are the steps that are performed by the search engine:

- The search engine looks for the keyword in the index for predefined database instead of going directly to the web to search for the keyword.
- It then uses software to search for the information in the database. This software component is known as web crawler.
- Once web crawler finds the pages, the search engine then shows the relevant web pages as a result. These retrieved web pages generally include title of page, size of text portion, first several sentences etc.

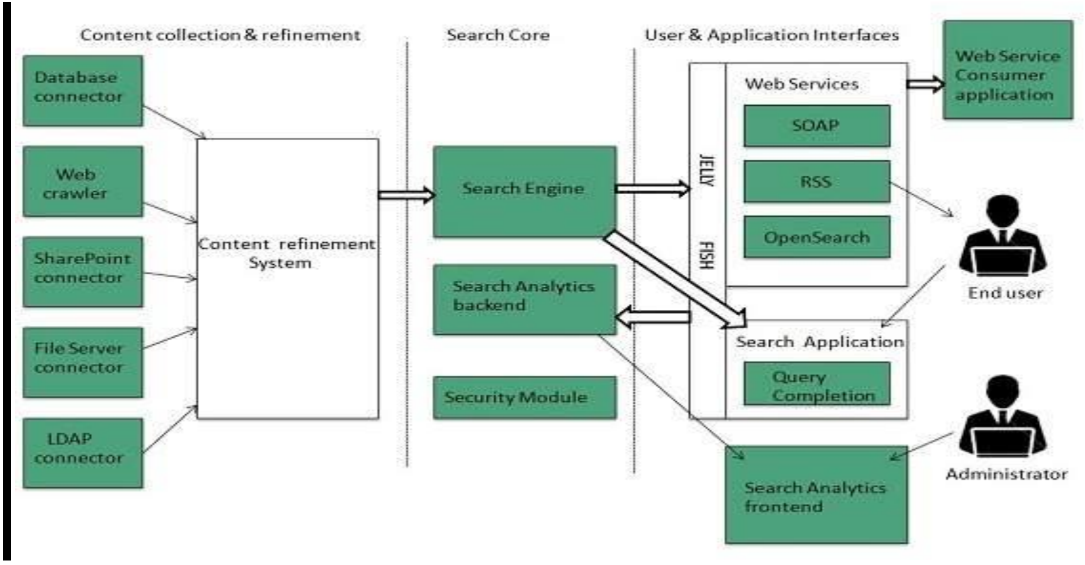
These search criteria may vary from one search engine to the other. The retrieved information is ranked according to various factors such as frequency of keywords, relevancy of information, links etc.

- User can click on any of the search results to open it.

Architecture

The search engine architecture comprises of the three basic layers listed below:

- Content collection and refinement.
- Search core
- User and application interfaces



Search Engine Processing

Indexing Process

Indexing process comprises of the following three tasks:

- Text acquisition
- Text transformation
- Index creation

Text acquisition

It identifies and stores documents for indexing.

Text Transformation

It transforms document into index terms or features.

Index Creation

It takes index terms created by text transformations and create data structures to support fast searching.

Query Process

Query process comprises of the following three tasks:

- User interaction
- Ranking
- Evaluation

User interaction

It supporst creation and refinement of user query and displays the results.

Ranking

It uses query and indexes to create ranked list of documents.

➢ Introduction to HTML

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>
```

Description of HTML Example

<! DOCTYPE>: It defines the document type or it instructs 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.

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.

➢ HTML5

HTML5 provides details of all 40+ HTML tags including audio, video, header, footer, data, datalist, article etc. This HTML tutorial is designed for beginners and professionals.

HTML5 is a next version of HTML. Here, you will get some brand new features which will make HTML much easier. These new introducing features make your website layout clearer to both website designers and users. There are some elements like <header>, <footer>, <nav> and <article> that define the layout of a website.

Why use HTML5

It is enriched with advance features which makes it easy and interactive for designer/developer and users.

It allows you to play a video and audio file.

It allows you to draw on a canvas.

It facilitate you to design better forms and build web applications that work offline.

It provides you advance features for that you would normally have to write JavaScript to do.

The most important reason to use HTML 5 is, we believe it is not going anywhere. It will be here to serve for a long time according to W3C recommendation.

HTML 5 Example

Let's see a simple example of HTML5.

```
1.  <!DOCTYPE>
2.  <html>
3.  <body>
4.  <h1>Write Your First Heading</h1>
5.  <p>Write Your First Paragraph.</p>
6.  </body>
7.  </html>
```

HTML5 User Manual

HTML Coding Convention

You should follow some convention while using HTML:

Use a Consistent CSS

A user must use consistent style while writing HTML. It makes the code simpler and more understandable for people.

Your code must be small, clean and well-formed.

Use Correct Document Type

You should declare document type at the starting of your code..2M630TML Tutorial

For example:

<!DOCTYPE html>

You can also use <!DOCTYPE html> to maintain your lower case practice.

Use Lower Case Element Names

HTML5 facilitates you to use upper case and lower case letters in element name. But it is good practice to use only lower case. Because:

- Mixing lower case and upper case letters in elements is not a good idea.
- Lowercase looks neat and clean.
- Lower case is easy to write.

- Developers mainly use lower case letters.

Example:

Bad practice:

```
<SECTION>
<p>This is javatpoint.com</p>
</SECTION>
```

Very Bad practice:

```
<Section>
<p>This is a javatpoint.com</p>
</SECTION>
```

Good practice:

```
<section>
<p>This is javatpoint.com.</p>
</section>
```

Close all HTML Elements

In HTML5, it is not required to close all HTML tags. But, it is recommended to close tags.

Bad practice:

```
<section>
<p>This is javatpoint.com
</section>
```

Good practice:

```
<section>
<p>This is javatpoint.com</p>
</section>
```

Close empty HTML Elements

In HTML5, it is not mandatory to close empty HTML tags. You can close it or let it be unclosed.

Good practice:

```
<meta charset="utf-8">
```

Don't Omit <html> and <body>

HTML5 facilitates you to omit and tag. You can exclude both tags and the program will work well enough.

Example:

```
<!DOCTYPE html>
<head>
<title>Page Title</title>
</head>
<h1>This is javatpoint.com</h1>
<p>Welcome to javatpoint.com</p>
```

Syntax:

```
<!DOCTYPE html>
<html lang="en-US">
```

Example:

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<title>Page Title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

HTML 5 Tags

There is a list of newly included tags in HTML 5. These HTML 5 tags (elements) provide a better document structure. This list shows all HTML 5 tags in alphabetical order with description.

List of HTML 5 Tags

Tag	Description
<article>	This element is used to define an independent piece of content document, that may be a blog, a magazine or a newspaper article.
<aside>	It specifies that article is slightly related to the rest of the whole page.
<audio>	It is used to play audio file in HTML.
<bdi>	The bdi stands for bi-directional isolation. It isolates a part of text that is formatted in other direction from the outside text document.
<canvas>	It is used to draw canvas.
<data>	It provides machine readable version of its data.
<datalist>	It provides auto complete feature for text field.
<details>	It specifies the additional information or controls required by user.
<dialog>	It defines a window or a dialog box.
<figcaption>	It is used to define a caption for a <figure> element.
<figure>	It defines a self-contained content like photos, diagrams etc.
<footer>	It defines a footer for a section.
<header>	It defines a header for a section.
<main>	It defines the main content of a document.
<mark>	It specifies the marked or highlighted content.
<menuitem>	It defines a command that the user can invoke from a popup menu.
<meter>	It is used to measure the scalar value within a given range.
<nav>	It is used to define the navigation link in the document.
<progress>	It specifies the progress of the task.
<rp>	It defines what to show in browser that don't support ruby annotation.
<rt>	It defines an explanation/pronunciation of characters.
<ruby>	It defines ruby annotation along with <rp> and <rt>.
<section>	It defines a section in the document.
<summary>	It specifies a visible heading for <detailed> element.
<svg>	It is used to display shapes.
<time>	It is used to define a date/time.
<video>	It is used to play video file in HTML.
<wbr>	It defines a possible line break.

➢ HTML Forms-Controls

An **HTML form** is *a* section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc. .

Why use HTML Form

HTML forms are required if you want to collect some data from of the site visitor.

Tag	Description
<form>	It defines an HTML form to enter inputs by the used side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<fieldset>	It groups the related element in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.

For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

HTML Form Syntax

```
<form action="server url" method="get|post">
  //input controls e.g. textfield, textarea, radiobutton, button
</form>
```

HTML Form Tags

Let's see the list of HTML 5 form tags.

HTML 5 Form Tags

Let's see the list of HTML 5 form tags.

Tag	Description
<datalist>	It specifies a list of pre-defined options for input control.
<keygen>	It defines a key-pair generator field for forms.
<output>	It defines the result of a calculation.

HTML <form> element

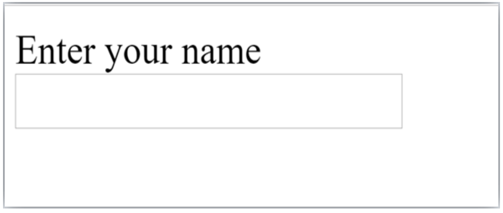
The HTML <form> element provide a document section to take input from user. It provides various interactive controls for submitting information to web server such as text field, text area, password field, etc.

Syntax:

```
<form>
//Form elements  </form>
```

HTML <input> element

The HTML <input> element is fundamental form element. It is used to create form fields, to take input from user. We can apply different input filed to gather different information form user. Following is the example to show the simple text input.



Enter your name

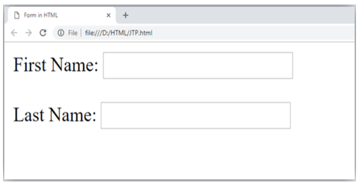
Example:

```
<body>
  <form>
     Enter your name  <br>
    <input type="text" name="username">    </form>
</body>
```

Output:

HTML TextField Control

The type="text" attribute of input tag creates textfield control also known as single line textfield control. The name attribute is optional, but it is required for the server side component such as JSP, ASP, PHP etc.



First Name:

Last Name:

```
        <form>
    First Name: <input type="text" name="firstname"/> <br/>
  Last Name:  <input type="text" name="lastname"/> <br/>   </form>
```

Output:

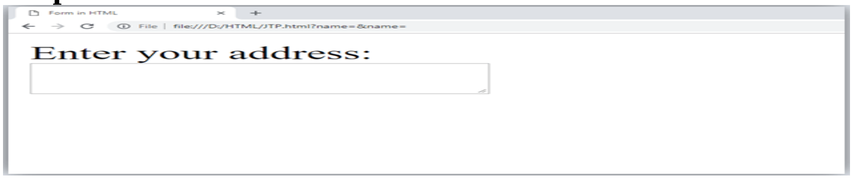
HTML <textarea> tag in form

The <textarea> tag in HTML is used to insert multiple-line text in a form. The size of <textarea> can be specify either using "rows" or "cols" attribute or by CSS.

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Form in HTML</title>
</head>
<body>
  <form>
        Enter your address:<br>
      <textarea rows="2" cols="20"></textarea>
  </form>
</body>
</html>
```

Output:



Enter your address:

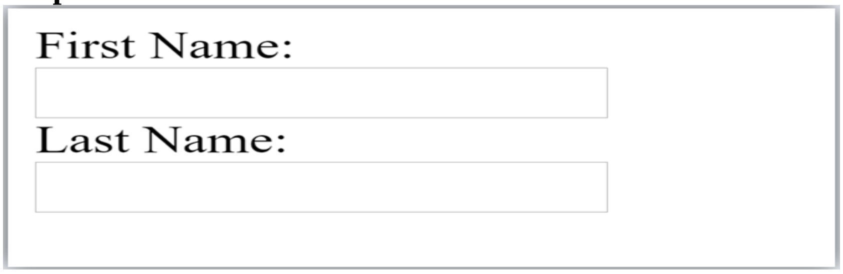
Label Tag in Form

It is considered better to have label in form. As it makes the code parser/browser/user friendly.

If you click on the label tag, it will focus on the text control. To do so, you need to have for attribute in label tag that must be same as id attribute of input tag.

```
<form>
    <label for="firstname">First Name: </label> <br/>
              <input type="text" id="firstname" name="firstname"/> <br/>
   <label for="lastname">Last Name: </label>
              <input type="text" id="lastname" name="lastname"/> <br/>
 </form>
```

Output:



First Name:

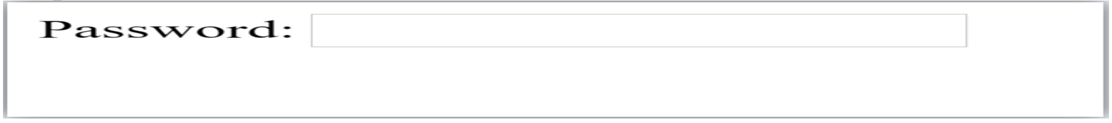
Last Name:

HTML Password Field Control

The password is not visible to the user in password field control.

```
<form>
    <label for="password">Password: </label>
              <input type="password" id="password" name="password"/> <br/>
</form>
```

Output:



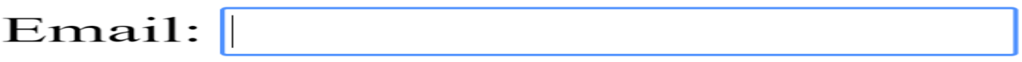
Password:

HTML 5 Email Field Control

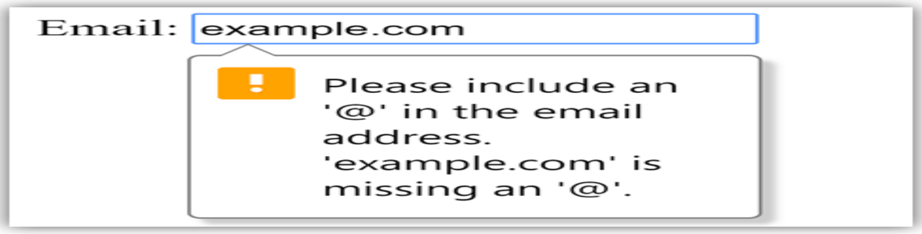
The email field in new in HTML 5. It validates the text for correct email address. You must use @ and . in this field.

```
<form>
    <label for="email">Email: </label>
              <input type="email" id="email" name="email"/> <br/>
</form>
```

It will display in browser like below:



Email:



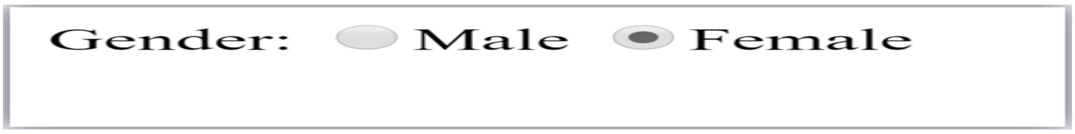
Email: example.com

Please include an '@' in the email address. 'example.com' is missing an '@'.

Radio Button Control

The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.
If you use one name for all the radio buttons, only one radio button can be selected at a time.
Using radio buttons for multiple options, you can only choose a single option at a time.

```
<form>
        <label for="gender">Gender: </label>
         <input type="radio" id="gender" name="gender" value="male"/>Male
         <input type="radio" id="gender" name="gender" value="female"/>Female <br/>
</form>
```



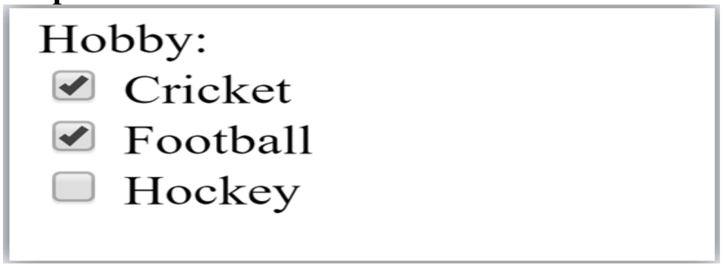
Gender: ○ Male ● Female

Checkbox Control

The checkbox control is used to check multiple options from given checkboxes.

```
<form>
Hobby:<br>
         <input type="checkbox" id="cricket" name="cricket" value="cricket"/>
           <label for="cricket">Cricket</label> <br>
         <input type="checkbox" id="football" name="football" value="football"/>
           <label for="football">Football</label> <br>
         <input type="checkbox" id="hockey" name="hockey" value="hockey"/>
           <label for="hockey">Hockey</label>
</form>
```

Output:



Hobby:
☑ Cricket
☑ Football
□ Hockey

Submit button control

HTML **<input type="submit">** are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

Syntax:

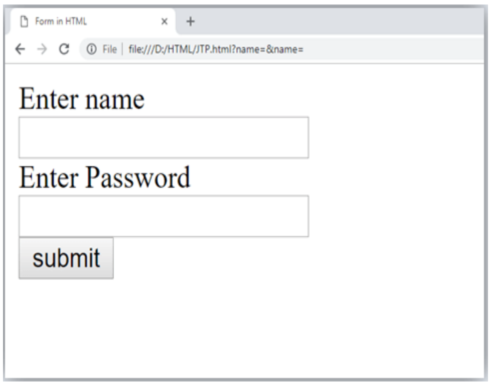
```
<input type="submit" value="submit">
```

The type = submit , specifying that it is a submit button
The value attribute can be anything which we write on button on web page.
The name attribute can be omit here.

Example:

```
  <form>
   <label for="name">Enter name</label><br>
   <input type="text" id="name" name="name"><br>
   <label for="pass">Enter Password</label><br>
   <input type="Password" id="pass" name="pass"><br
>
   <input type="submit" value="submit">
</form>
```

Output:



Enter name

Enter Password

submit

HTML <fieldset> element:

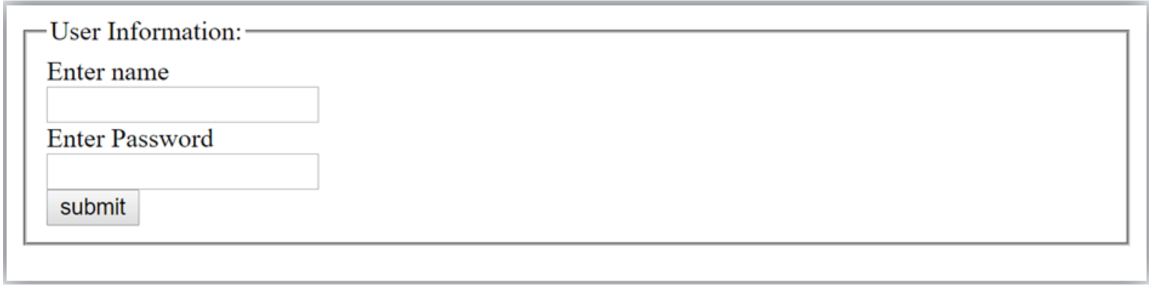
The <fieldset> element in HTML is used to group the related information of a form. This element is used with <legend> element which provide caption for the grouped elements.

Example:

```
<form>
```

```
    <fieldset>
     <legend>User Information:</legend>
    <label for="name">Enter name</label><br>
<input type="text" id="name" name="name"><br>
<label for="pass">Enter Password</label><br>
<input type="Password" id="pass" name="pass"><br>
<input type="submit" value="submit">
</fieldset>
lt;/form>
```

Output:

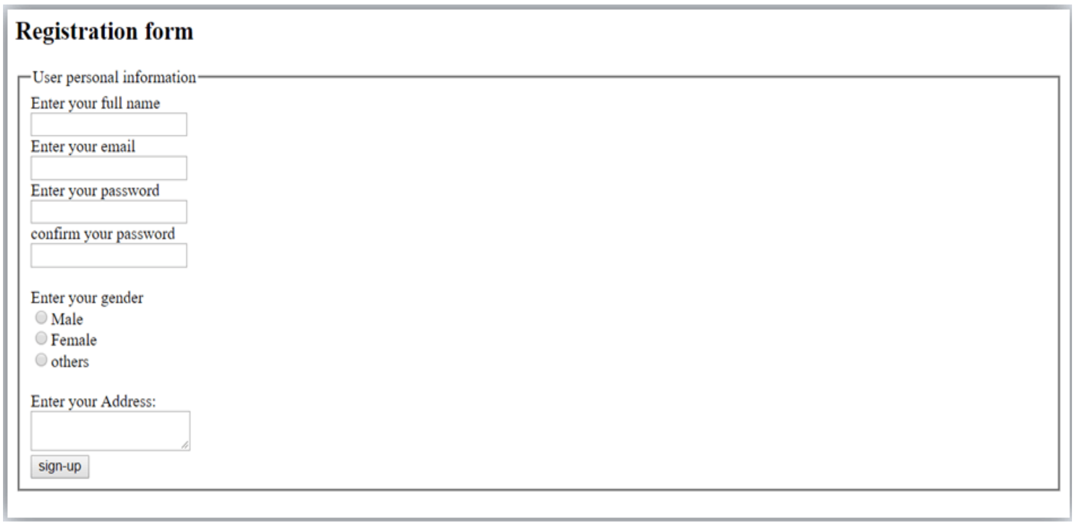


HTML Form Example

Following is the example for a simple form of registration.

```
<!DOCTYPE html>
 <html>
 <head>
  <title>Form in HTML</title>
</head>
 <body>
    <h2>Registration form</h2>
   <form>
    <fieldset>
       <legend>User personal information</legend>
       <label>Enter your full name</label><br>
       <input type="text" name="name"><br>
        <label>Enter your email</label><br>
        <input type="email" name="email"><br>
        <label>Enter your password</label><br>
        <input type="password" name="pass"><br>
        <label>confirm your password</label><br>
        <input type="password" name="pass"><br>
        <br><label>Enter your gender</label><br>
        <input type="radio" id="gender" name="gender" value="male"/>Male  <br>
        <input type="radio" id="gender" name="gender" value="female"/>Female <br/>
        <input type="radio" id="gender" name="gender" value="others"/>others <br/>
         <br>Enter your Address:<br>
        <textarea></textarea><br>
        <input type="submit" value="sign-up">
    </fieldset>
  </form>
 </body>
</html>
```

Output :



Registration form

User personal information

Enter your full name

Enter your email

Enter your password

confirm your password

Enter your gender

○ Male

○ Female

○ others

Enter your Address:

sign-up

HTML Form Example

Let's see a simple example of creating HTML form.

```
<form action="#">
<table>
<tr>
    <td class="tdLabel"><label for="register_name" class="label">Enter name:</label></td>

    <td><input type="text" name="name" value="" id="register_name" style="width:160px"/></td>
</tr>
<tr>
    <td class="tdLabel"><label for="register_password" class="label">Enter password:</label></td>
    <td><input type="password" name="password" id="register_password" style="width:160px"/></td>
</tr>
<tr>
    <td class="tdLabel"><label for="register_email" class="label">Enter Email:</label></td>
    <td
><input type="email" name="email" value="" id="register_email" style="width:160px"/></td>
</tr>
<tr>
    <td class="tdLabel"><label for="register_gender" class="label">Enter Gender:</label></td>
    <td>
<input type="radio" name="gender" id="register_gendermale" value="male"/>
<label for="register_gendermale">male</label>
<input type="radio" name="gender" id="register_genderfemale" value="female"/>
<label for="register_genderfemale">female</label>
    </td>
</tr>
<tr>
    <td class="tdLabel"><label for="register_country" class="label">Select Country:</label></td>
    <td><select name="country" id="register_country" style="width:160px">
    <option value="india">india</option>
    <option value="pakistan">pakistan</option>
    <option value="africa">africa</option>
    <option value="china">china</option>
    <option value="other">other</option>
</select>
</td>
</tr>
<tr>
    <td colspan="2"><div align="right"><input type="submit" id="register_0" value="register"/>
</div></td>
</tr>
</table>
</form>
```

➢ HTML Audio Tag

HTML audio tag is used to define sounds such as music and other audio clips. Currently there are three supported file format for HTML 5 audio tag.

1. mp3
2. wav
3. ogg

HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items.

This table defines that which web browser supports which audio file format.

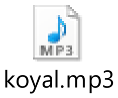
Browser	mp3	wav	ogg
Internet Explorer	yes	no	no
Google Chrome	yes	yes	yes
Mozilla Firefox	yes*	yes	yes
Opera	no	yes	yes
Apple Safari	yes	yes	no

HTML Audio Tag Example

Let's see the code to play mp3 file using HTML audio tag.

```
<audio controls>
  <source src="koyal.mp3" type="audio/mpeg">
Your browser does not support the html audio tag.
</audio>
        <audio controls>
          <source src="koyal.mp3" type="audio/mpeg">
        Your browser does not support the html audio tag.
        </audio>
```

Output:



koyal.mp3

Let's see the example to play ogg file using HTML audio tag.

```
<audio controls>
  <source src="koyal.ogg" type="audio/ogg">
Your browser does not support the html audio tag.
</audio>
```

Supporting Browsers

Element	Chrome	IE	Firefox	Opera	Safari
<audio>	Yes	Yes	Yes	Yes	Yes

Attributes of HTML Audio Tag

There is given a list of HTML audio tag.

Attribute	Description
controls	It defines the audio controls which is displayed with play/pause buttons.
autoplay	It specifies that the audio will start playing as soon as it is ready.
loop	It specifies that the audio file will start over again, every time when completed.
muted	It is used to mute the audio output.
preload	It specifies the author view to upload audio file when the page loads.
src	It specifies the source URL of the audio file.

HTML Audio Tag Attribute Example

Here we are going to use controls, autoplay, loop and src attributes of HTML audio tag.

```
1. <audio controls autoplay loop>
2.   <source src="koyal.mp3" type="audio/mpeg"></audio>
```

MIME Types for HTML Audio format

The available MIME type HTML audio tag is given below.

Audio Format	MIME Type
mp3	audio/mpeg
ogg	audio/ogg
wav	audio/wav

➢ HTML Video Tag

HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

Currently, there are three video formats supported for HTML video tag:

1. mp4
2. webM
3. ogg

Let's see the table that defines which web browser supports video file format.

Browser	mp4	webM	ogg
Internet Explorer	yes	no	no
Google Chrome	yes	yes	yes
Mozilla Firefox	yes	yes	yes
Opera	no	yes	yes
Apple Safari	yes	no	no

HTML Video Tag Example

Let's see the code to play mp4 file using HTML video tag.

```
1. <video controls>
2.   <source src="movie.mp4" type="video/mp4">
3.   Your browser does not support the html video tag.
4. </video>
```

Let's see the example to play ogg file using HTML video tag.

```
1. <video controls>
2.    <source src="movie.ogg" type="video/ogg">
3.   Your browser does not support the html video tag.
4. </video>
```

Supporting Browsers

Element	Chrome	IE	Firefox	Opera	Safari
<video>	Yes	Yes	Yes	Yes	Yes

Attributes of HTML Video Tag

Let's see the list of HTML 5 video tag attributes.

Attribute	Description
controls	It defines the video controls which is displayed with play/pause buttons.
height	It is used to set the height of the video player.
width	It is used to set the width of the video player.
poster	It specifies the image which is displayed on the screen when the video is not played.
autoplay	It specifies that the video will start playing as soon as it is ready.
loop	It specifies that the video file will start over again, every time when completed.
muted	It is used to mute the video output.
preload	It specifies the author view to upload video file when the page loads.
src	It specifies the source URL of the video file.

HTML Video Tag Attribute Example

Let's see the example of video tag in HTML where are using height, width, autoplay, controls and loop attributes.

```
<video width="320" height="240" controls autoplay loop>
  <source src="movie.mp4" type="video/mp4">
  Your browser does not support the html video tag.
</video>
```

MIME Types for HTML Video format

The available MIME type HTML video tag is given below.

Video Format	MIME Type
mp4	video/mp4
ogg	video/ogg
webM	video/webM

```
<!DOCTYPE>
<html>
<head>
<style>
h1{
color:white;
background-color:red;
padding:5px;
}
p{
color:blue;
}
</style>
</head>
<body>
<h1>Write Your First CSS Example</h1>
<p>This is Paragraph.</p>
</body>
</html>
```

➢ CSS STYLING

Introduction:

- **Content and Style**
- **HTML Structure**
- **CSS Attachment to Pages**
- **CSS Main Facilities**
- **Browser Support**
- **CSS 1 Specification**

Content and Style:

Two aspects of any document are **content** and **style.**

The **content** gives the **information** to be presented

And the **style** defines how the information is presented. Most publishers have a House Style that is a consistent way o presenting information.

HTML's main role is to **define content**. What is needed is the ability for publishers to have control of style. In the context of the Web, the publisher can be the organisation that owns the Web site but it can also be the person viewing the information. It is only by having a clear separation between content and style that this can be achieved. So in this Primer we shall use HTML in most of the examples as a means of defining the **content** and CSS to define the **style**.

CSS is a **World Wide Web Consortium (W3C**) Recommendation. That means that all the W3C Members(which includes all the main browser manufacturers) have agreed to support it in their products. That does not necessarily mean immediately as each has its own production schedule for enhancements to its products but long term all should support all of the features. CSS first became a W3CRecommendation in 1996 (called CSS 1) and there was a significant update in May 1998 .**HTML** allows a Web page to be laid out and there is some guidance to browsers as to how to represent each element. Thus, **h1** should be larger than **h2** and **em** should be emphasised by some method but
that is as far as it goes.

The aim of Cascading Style Sheets (CSS) is to give the page developer much more control on how apage should be displayed by all browsers.

A **style sheet** is a set of **rules** that controls the formatting of HTML elements on one or more Web pages. Thus, the appearance of a Web page can be changed by changing the style sheet associated with it. There is no need to make detailed changes within the Web page to change how it looks. <u>advantages of using style sheets</u> :

1. **accessibility**, different styling can be provided for different users dependent on their requirements.
2. Separating style and content is **good design** and will normally produce a better and more consistent web site. As one style sheet can be used for a whole web site, it normally means that the overall **size** of the web site is smaller and the downloads required for each page can be decreased by up to 40%.

An Allwoing browser to make overall decisions on styling often means that the rendering time by the browser is also shorter.

3. A Web page may have its own style sheet that refines the information in the common style sheet. Readers may define their own style sheet indicating their preferences. Thus style sheets **cascade** and decisions need to be made as to which style sheet is in control when there is a conflict.

A <u>style sheetrule</u>consists of **two parts**. A **selector** that defines which HTML elements are controlled by the rule and a **declaration** that says what the required effect is.

Thus a simple rule is:

```
h1 { color: blue }
```

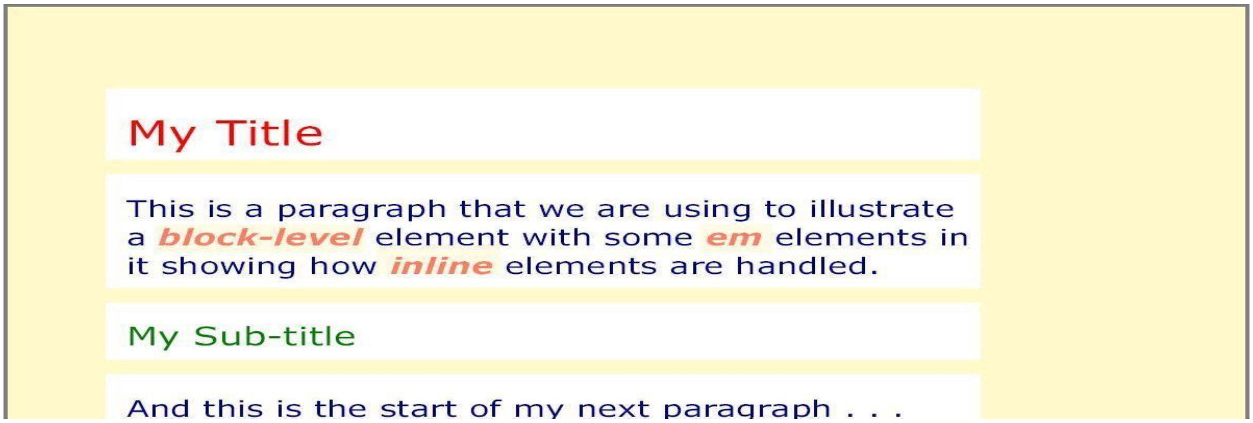
This says that all **h1** elements in a page should be displayed in blue. The selector is **h1**. **color** is the **property** that is to be changed, **blue** is the **value** that the property is changed to and **color: blue** is thedeclaration.

HTML Structure

HTML elements in the body of the document fall into two main classes:

1. Block-level
2. Inline (sometimes called character-level)

Figure 1.1: Block level and Inline Elements



My Title

This is a paragraph that we are using to illustrate a *block-level* element with some *em* elements in it showing how *inline* elements are handled.

My Sub-title

And this is the start of my next paragraph . . .

Figure 1.1: Block level and Inline Elements

Lists and tables are special types of elements that have their own unique styling.

When a **block-level element** is inserted into a document, it terminates the previous element and effectively starts a new line. Some examples are **<p>**and **<h1>**. Inline elements do not terminate the previous element and form part of the previous element; **em** is an example.

The **body** of an HTML page contains text marked-up using block-level elements (**h1, p, ul, div** etc). Each block-level element has an area which contains the content. For a **p** element, this is the area that contains the whole paragraph while for **h1** it is often an area consisting of a single line of text. CSS provides facilities for controlling where that area is placed relative to other areas.

The **inline** elements often inherit the style of the block-level element that they are part of but can be given special styling appropriate to the inline element.

For example:

```
<h1>My Title</h1>
<p>This is a paragraph that I am using to illustratea <em>block-level</em> element
with some <em>em</em> elements in it showing how <em>inline</em> elements are handled.</p>
<h2>My Sub-title</h2>
<p>And this is the start of my next paragraph . . .</p>
```

Figure 1.1 shows the areas covered by the block level elements and the sub-areas covered by the inline elements. The inline elements have been given their own background colour, colour, font-weight (bold) and font style (italic) while inheriting the text size and font from the enclosing paragraph.

CSS Attachment to Pages:

There are **four ways** of linking a style sheet to an HTML page:

1. By defining a link from the HTML page to the style sheet (normally stored in a **xxx.css** file). This allows you to have a single style sheet that changes the appearance of many Web pages.
2. By specifying an HTML **style** element in the **head** of the page. This allows you to define the appearance of a single page.
3. By adding inline styles to specific elements in the HTML file by the **style** attribute. This allows you to change a single element or set of elements. This should only be used when absolutely necessary as it negates some of the advantages of having style sheets. It effectively over-rides

the overall style for the page. Also, it is likely that it will be removed from CSS 3.

4. By importing a style sheet stored externally into the current style. The **style** element for the page can consist of a set of imports plus some rules specific to the page.

Initially, while you are experimenting, the simplest method is to use the HTML **style** element to adda style sheet to an HTML page. The **style** element, consisting of a set of rules, is placed in the document **head**. When you have decided on your house style, it is likely that you will move to having the style sheet external to an individual page and either linked in or imported.

Main Facilities

The facilities provided by style sheets are much as you would expect:

1. Better control of fonts including colour
2. Better control of block-level layout (indents, margins, alignment etc)
3. Better control of inline layout, particularly with regard to diagrams and related text

Browser Support

To achieve these effects, the Web browser has to know what to do with style sheets. Really old browsers will not have this capability but the recent offerings from Microsoft, Opera and Netscape have good support now and are committed to full support in the future. HTML Editors are beginning to have Style Sheet additions and separate Style Sheet Editors are appearing. Visit the W3C CSS Web Page [1] for all the latest information.

Given the way CSS has been implemented, it is possible to design your web pages so that they still present the HTML information even when CSS support is not there. So there is little excuse for not adding style to your web pages now even if you expect your pages are to be viewed by really old browsers.

CSS Specification

This Primer will not tell you everything about CSS. If you get to the position where you really need to know precisely what happens in some subtle situation, you need to consult the formal specification that can be found on the World-Wide Web Consortium's web site [2] [3].

To make it as easy as possible to relate the specification to the Primer, the order of presentation here is similar to the order in the Specification.

2. CSS Rules:

Introduction

Syntax for CSS Properties

Introduction

Let us assume initially that we wish to define a style sheet by adding a **style** element to a page. Thepage will look something like:

```
<html>
<style type="text/css">
h1 {font: 12pt/14pt "Palatino"; font-weight: bold; color: red} h2
{font: 10pt/12pt "Palatino"; font-weight: bold; color: blue}p
{font: 10pt/12pt "Times"; color: black}
</style>
<body>
...:
</body>
</html>
```

The **set of rules** are contained between **<style>**and **</style>**. Ignore the **type** attribute, for now. The reason for this will be come clearer later. For now let us concentrate on the rules. As stated in the introduction, each rule consists **of two parts**:

a **selector** that defines what HTML elements are controlled by the rule and a **declaration** that says what the required effect is. The selectors above define three rules that apply to all **h1** elements, all **h2** elements and all **p** elements respectively. We shall show how we can be both more selective (control a subset of **h1** elements) or control a set of elements (a block of different HTML elements) later. Initially let us look at the declarations. Each declaration consists of the name of a **propert**y and the value to be assigned to it. So the simplest style rule is:

```
selector { property: value }
```

To set several properties in a single rule requires the individual declarations to be separated bysemicolons:

```
selector { property1: value1; property2: value2; property3: value3 }
```

Syntax for CSS Properties:

The general format for each sub-section is for

the title to give a descriptive name followed immediately by the syntax definition (in BNF style) for theproperty declaration.

For example:

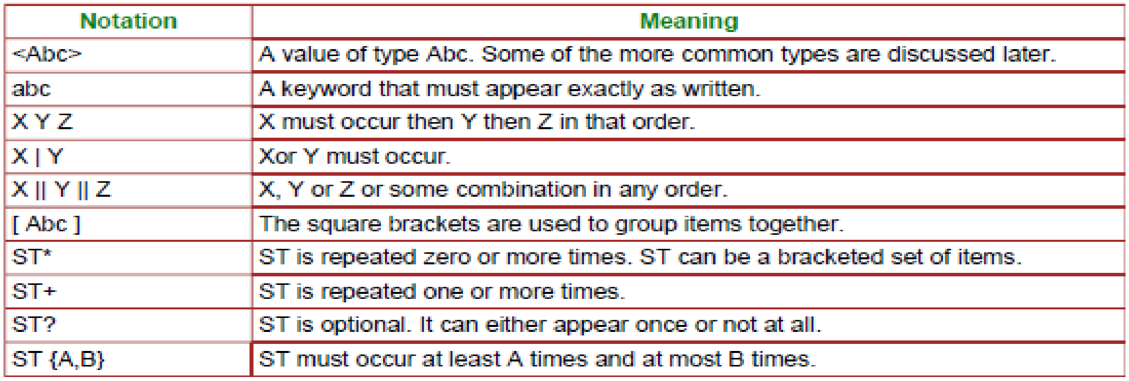
```
font-weight: <value>
```

This says that the name of the property is **font-weight** and **<value>**is assigned to it. In the CSS specification, the possible values of **<value>**are defined. In this Primer, sometimes only a sample of the possible values are given to give an idea of what is possible rather than exhaustively list them all.

In BNF, the notation could be written:

<font-weight> ::= font-weight: <value>

This style is used if, for example, the definition of <value> needs to be given separately. Some of the syntax definitions can be quite complicated. The notation used is asfollows:**Notation Meaning:**



Notation	Meaning
<Abc>	A value of type Abc. Some of the more common types are discussed later.
abc	A keyword that must appear exactly as written.
X Y Z	X must occur then Y then Z in that order.
X | Y	Xor Y must occur.
X || Y || Z	X, Y or Z or some combination in any order.
[Abc]	The square brackets are used to group items together.
ST*	ST is repeated zero or more times. ST can be a bracketed set of items.
ST+	ST is repeated one or more times.
ST?	ST is optional. It can either appear once or not at all.
ST {A,B}	ST must occur at least A times and at most B times.

3. Length, Percentage, Color and URLs:

□ 3.1 Introduction

□ 3.2 Length Values

□ 3.3 Percentage values

□ 3.4 Color Values

□ 3.5 URLs

□ 3.6 CSS Specification

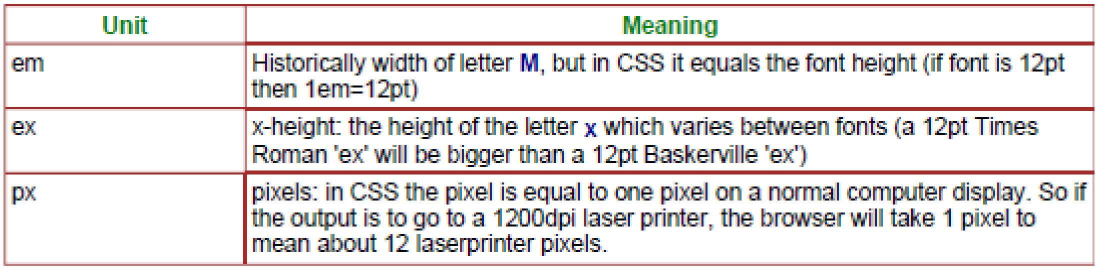
Introduction

Four values (<length>, <percentage> , <color>, <url>) are used by many of the rules. They areexplained in this secti

Length Values

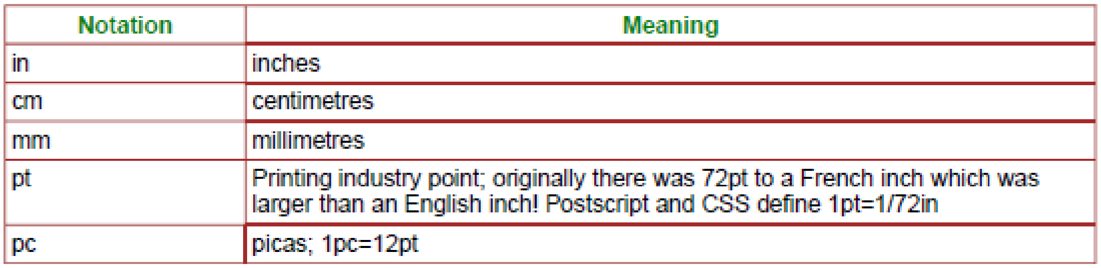
<length> ::= [+ | -]? <number><unit>

A length value is formed by an optional + or -, followed by a number, followed by a two-letter abbreviation that indicates the unit (this can be left out if the value is 0). There are no spaces in the middle of a length value. Both relative and absolute length units are supported in CSS. Relative units give a length relative to another length property. The **relative length units** are:



Unit	Meaning
em	Historically width of letter **M**, but in CSS it equals the font height (if font is 12pt then 1em=12pt)
ex	x-height: the height of the letter **x** which varies between fonts (a 12pt Times Roman 'ex' will be bigger than a 12pt Baskerville 'ex')
px	pixels: in CSS the pixel is equal to one pixel on a normal computer display. So if the output is to go to a 1200dpi laser printer, the browser will take 1 pixel to mean about 12 laserprinter pixels.

The absolute length units are:



Notation	Meaning
in	inches
cm	centimetres
mm	millimetres
pt	Printing industry point; originally there was 72pt to a French inch which was larger than an English inch! Postscript and CSS define 1pt=1/72in
pc	picas; 1pc=12pt

Percentage Values:

<percentage> ::= [+ | -]? <number> %

A percentage value is formed by an optional + or -, followed by a number, followed by %. There are no spaces in a percentage value.

Percentage values are relative to other values, as defined for each property. Often the percentage value is relative to th element's font size but it might be the width of the page.

Color Values:

<color> ::= <keyword> | # <hex><hex><hex> | # <hex><hex><hex><hex><hex>

<hex> |

A color value is a keyword or a numerical RGB specification. The original set of keyword colours was aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white and yellow. These have been enhanced as a result of the requirements from other specifications. The set of keywords with the corresponding RGB values given in Appendix C are likely to be supported by a browser.

RGB colours can be defined in one of four ways:

1. #rrggbb(e.g., #00ff00)

2. #rgb (e.g., #0f0)

3. rgb(x,x,x) where x is an integer between 0 and 255

4. rgb(y%,y%,y%) where y is a number between 0.0 and 100.0

The following examples all specify the same colour:

pre { color: red }h1 { color: #foo }

p { color: #ff0000 }

h2 { color: rgb(255,0,0) }

h3 { color: rgb(100%, 0%, 0%) }

Values outside the range are clipped. So, for example, integer values have a maximum of 255 andeverything above that will be changed to 255. Similarly 120% is interpreted as 100%.

URLs:

```
<url> ::= url( <whitespace>? [ ' | " ]? <characters in url> [ ' | " ]? <whitespace> )
```

A URL value is of the form:

```
url(xyz)
```

where xyz is the URL. The URL may be quoted with either single (') or double (") quotes and may have whitespace before the initial quote or after the final quote.

Parentheses, commas, spaces, single quotes, or double quotes in the URL must be escaped with a backslash. Partial URLs are interpreted relative to the style sheet source, not to the HTML source. Some examples are:

```
body { background: url("circle.gif") }
```

4. Font Properties:

- □ 4.1 Font Family (Sets typeface)
- □ 4.2 Font Style (Italicises text)
- □ 4.3 Font Variant (Mostly upper case)
- □ 4.4 Font Weight (Sets thickness of type)
- □ 4.5 Font Size (Sets size of text)
- □ 4.6 Font (Shorthand)

Font Family (set typeface)

font-family: [<family-name> | <generic-family>] [, [<family-name> | <generic-family>]]*

<family-name> ::= serif | sans-serif | cursive | fantasy | monospace

The font to use is set by the font-family property. Instead of defining a single font, a sequence of fonts should be listed. The browser will use the first if it is available but try the second and so on. The value can either be a specific font name or a generic font family.

Ex:p { font-family: "New Century Schoolbook", Times, serif }

Any font name containing spaces (multiple words) must be quoted, with either single or double quotes. That is why the first font name is quoted but Times is not.

The first two requests are for specific fonts **New Century Schoolbook** and **Times**. As both are serif fonts, the generic font family listed as a backup is the **serif** font family. In this case, Times will be used if New Century Schoolbook is not available, and a serif font if Times is not available.

Some example fonts are:

serif

Times New Roman, Bodini, Garamond

sans-serif

Trebuchet, Arial, Verdana, Futura, Gill Sans, Helvetica

cursive

Poetica, Zapf-Chancery, Roundhand, Script

fantasy

Critter, Cottonwood

monospace

Courier, Courier New, Prestige, Everson Mono

See also the **font** property which lets the user define a set of font properties in a single command..

Font Style (italicise text):

```
font-style: normal | italic | oblique
```

The font-style property has one of three values: **normal**, **italic** or **oblique** (slanted). A sample stylesheet might be:

```
h3 { font-style: italic } p
{ font-style: normal }
```

Italic and oblique are similar. ***Italic*** is normally a separate font while ***oblique*** often bends the normalfont.

Font Variant (size of upper case):

```
font-variant: normal | small-caps
```

Normally, **upper case characters** are much larger than the lower-case characters. **Small-caps are** often used when all the letters of the word are in capitals. In this case the upper case characters are onlyslightly larger than the lowercase ones. An example is:

```
p { font-variant: small-caps }
```

For example, the text **Elephant** with font-variant set to **small-caps** would appear as ELEPHANT.

Font Weight (set thickness of type):

```
font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
```

The font-weight property sets the weight of the font to values like **normal** and **bold**. It also has values **bolder** and **lighter** which are relative to any inherited font -weight. Block-level elements are displayed with default values inherited from the **body** or **div** element that surrounds it. It is also possible to define absolute font-weights. Values range from 100 to 900 with **normal** being 400 and bold being **700** approximately. If that much control is not possible, some of the weights may be grouped together (for example, 100, 200 and 300 may all look the same) and if the specified weight is not available, an alternative value will be chosen. Some examples are:

```
h1 { font-weight: 800 }
h2 { font-weight: bold }
h3 { font-weight: 500 }
```

Font Size (set size of text):

```
font-size: <absolute-size> | <relative-size> | <length> | <percentage>

<absolute-size> ::= xx-small | x-small | small | medium | large | x-large | xx-large
```

The **font-size property sets** the size of the displayed characters. The absolute-size values are **small**, **medium** and **large** with additional possibilities like **x-small** (extra small) and **xx-small**. Relative sizes are **smaller** and **larger**. They are relative to any inherited value for the property. The length value defines the precise height in units like **12pt** or **1in**. Percentage values are relative to the inherited value.

Some examples are:

```
h1 { font-size: large }
h2 { font-size: 12pt }
h3 { font-size: 5cm }
p { font-size: 10pt } li
{ font-size: 150% }
```

The guidance is that the medium font should be around 10pt so, in this case **li** and **em** might be similarin size.

Font (Shorthand):

```
font: [ <font-style> || <font-variant> ||
```

The **font property may** be used as a shorthand for the various font properties, as well as the line-height. For example:

```
p { font: italic bold 12pt/14pt Times, serif }
```

specifies paragraphs with a bold and italic Times or serif font with a size of 12 points and a line height of14 points. Note: The order of attributes is significant: the font weight and style must be specified before the others.

5. Color and Background Properties:

□ 5.1 Color

□ 5.2 Background Color

□ 5.3 Background Image

□ 5.4 Background Repeat

□ 5.5 Background Attachment

□ 5.6 Background Position

□ 5.7 Background (Sets background images or color)

Color

```
color: <color>
```

The color property sets the foreground colour of a text element. For example:

```
h1 {color: blue }

h2 {color: rgb(255,0,0) }
```

Background Color:

```
background-color: <color> | transparent
```

The **background-color property** sets the background colour of an element. For example:

```
body { background-color: white }
h1 { background-color: #000080 }
```

The value **transparent** indicates that whatever is behind the element can be seen. For example if the background to the **body** element was specified as being red, a block-level element with **backgroundcolor** set to **transparent** would appear with a red background.

Background Image:

```
background-image: <url> | none
```

The **background-image** property sets the background image of an element. For example:

```
body { background-image: url("/images/monkey.gif") }

p { background-image: url("http://www.cclrc.com/pretty.png") }
```

When a background image is defined, a compatible background colour should also be defined for thoseusers who do not load the image or to cover the case when it is unavailable. If parts of the image are transparent, the background colour will fill these parts.
For example:

```
<ul style="background-image:url('wall.png')">

<li>An item</li>

<li>Another</li>
```

Figure 5.1 shows what might be the result depending on the font-size and other properties specified for the list.

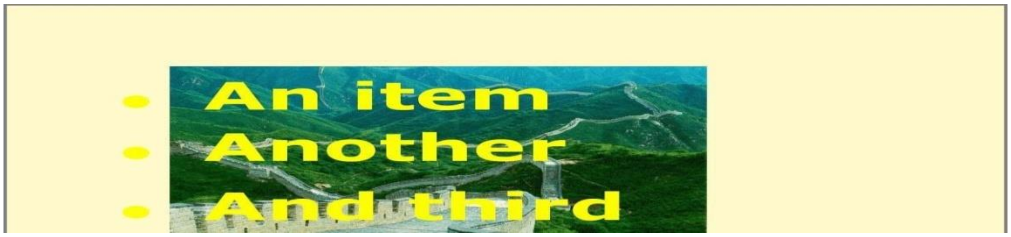


Figure 5.1: List with background-imageBackground Repeat:

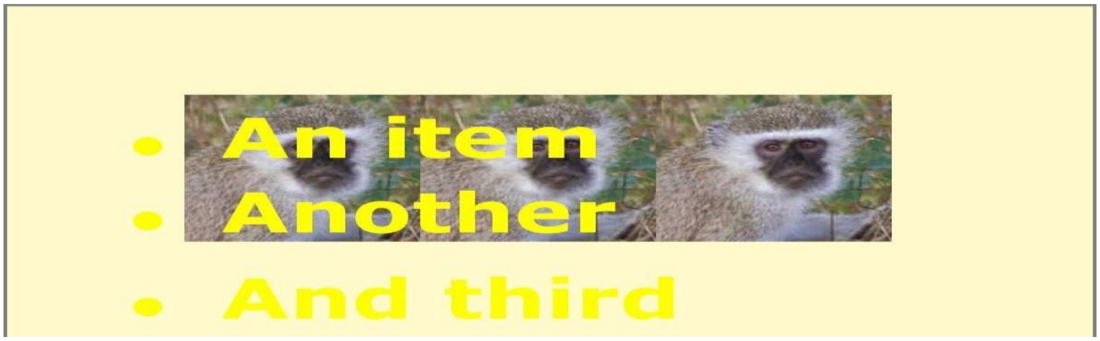
```
background-repeat: repeat | repeat-x | repeat-y | no-repeat
```

If the background image does not fill the whole element area, this description specifies how it is repeated. The **repeat-x** value will repeat the image horizontally while the **repeat-y** value will repeat the image vertically. Setting **background-image** to **repeat** will repeat the image in both x and y directions. For example:

```
body { background-image: url(/images/monkey.gif) }
body { background-repeat: repeat-x }
```

In the above example, the monkey will only be tiled horizontallya as shown in Figure 5.2.

Figure 5.2: Monkey tiled horizontally



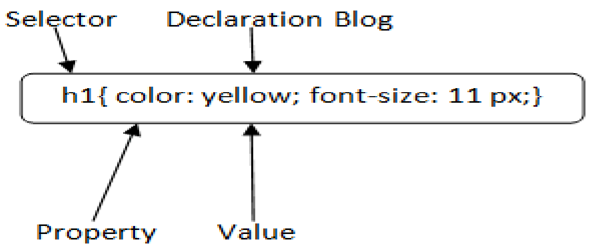
Background

Attachment: background-

attachment: scroll | fixed

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.

```
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.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
    text-align: center;
    color: blue;
}
</style>
</head>
```

```
<body>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</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".

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p id="para1">Hello Javatpoint.com</p>
<p>This paragraph will not be affected.</p>
</body>
</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.

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

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: blue;
}
</style>
```

```
</head>
<body>
<h1 class="center">This heading is blue and center-aligned.</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</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.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is not affected</h1>
<p class="center">This paragraph is blue and center-aligned.</p>
</body>
</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.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
   color: green;
   font-size: 20px;
}
</style>
</head>
<body>
<h2>This is heading</h2>
<p>This style will be applied on every paragraph.</p>
```

```
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</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.

```
h1 {
    text-align: center;
    color: blue;
}
h2 {
    text-align: center;
    color: blue;
}
p {
    text-align: center;
    color: blue;
}
```

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

```
h1,h2,p {
    text-align: center;
    color: blue;
}
```

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

```
<!DOCTYPE html>
<html>
<head>
<style>
h1, h2, p {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1>Hello Javatpoint.com</h1>
<h2>Hello Javatpoint.com (In smaller font)</h2>
<p>This is a paragraph.</p>
```

```
</body>
</html>
```

Output:

```
Hello Javatpoint.com
Hello Javatpoint.com (In smaller font)
This is a paragraph.
```

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:

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

For more visit here: Inline CSS

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:

```
<style>
p{color: blue}
</style>
```

For more visit here: Internal CSS

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:

```
p{color:blue}
```

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

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