Introduction to WWW

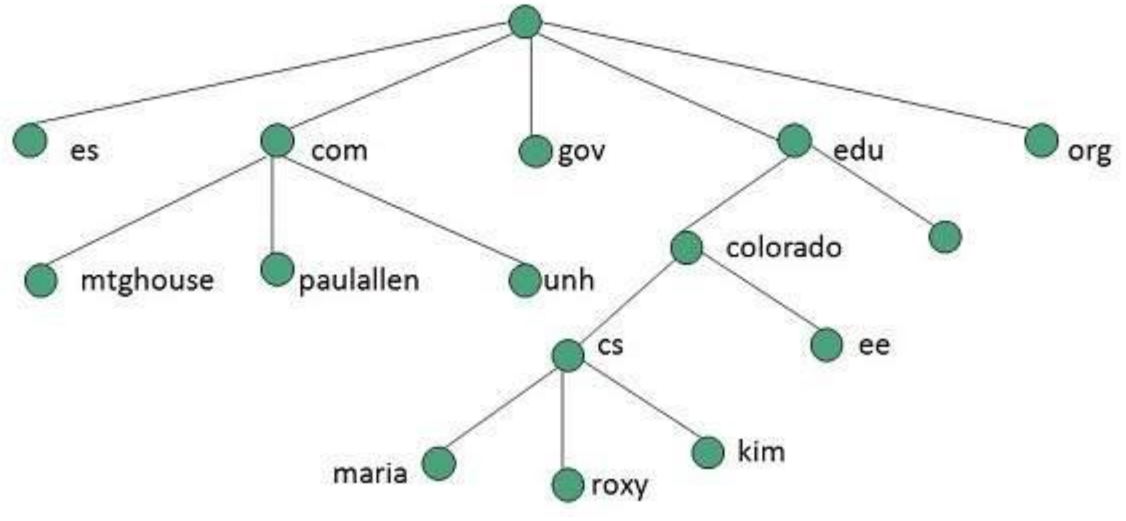
WWW stands for **World Wide Web.** A technical definition of the World Wide Web is : all the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP).

A broader definition comes from the organization that Web inventor **Tim Berners-Lee** helped found, the **World Wide Web Consortium (W3C).**

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.

In simple terms, The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.

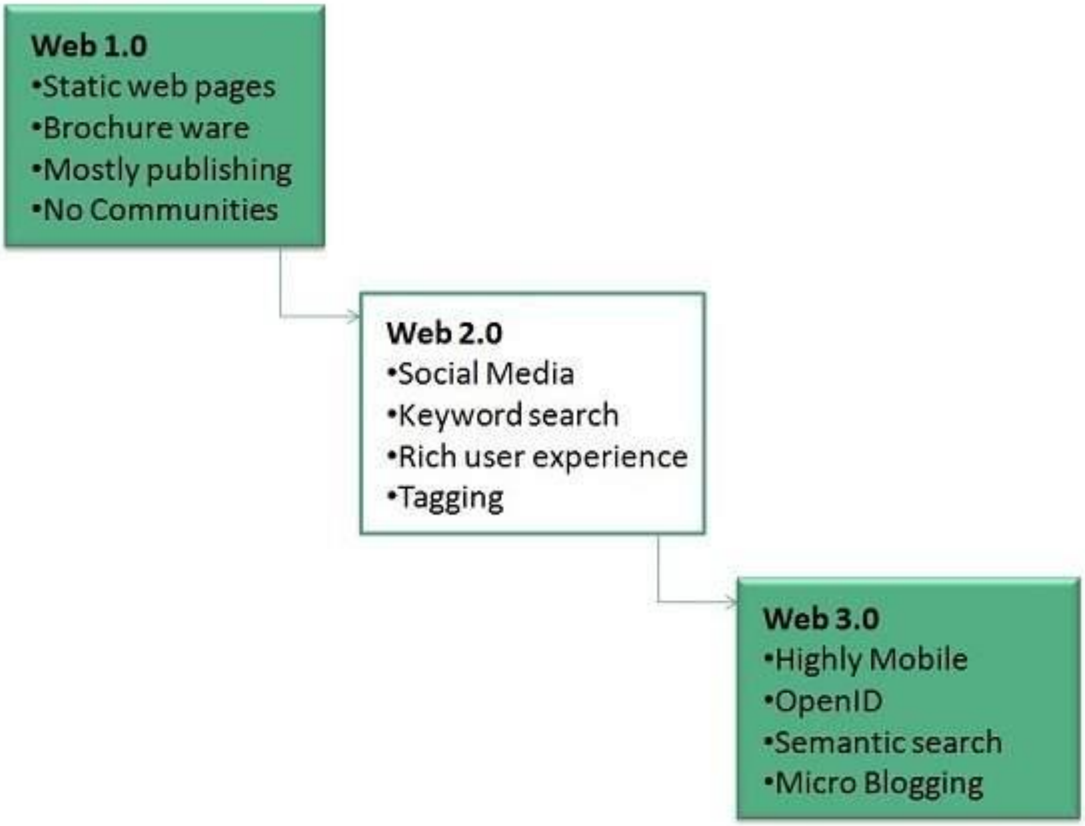
Internet and **Web** is not the same thing: Web uses internet to pass over the information.



Evolution

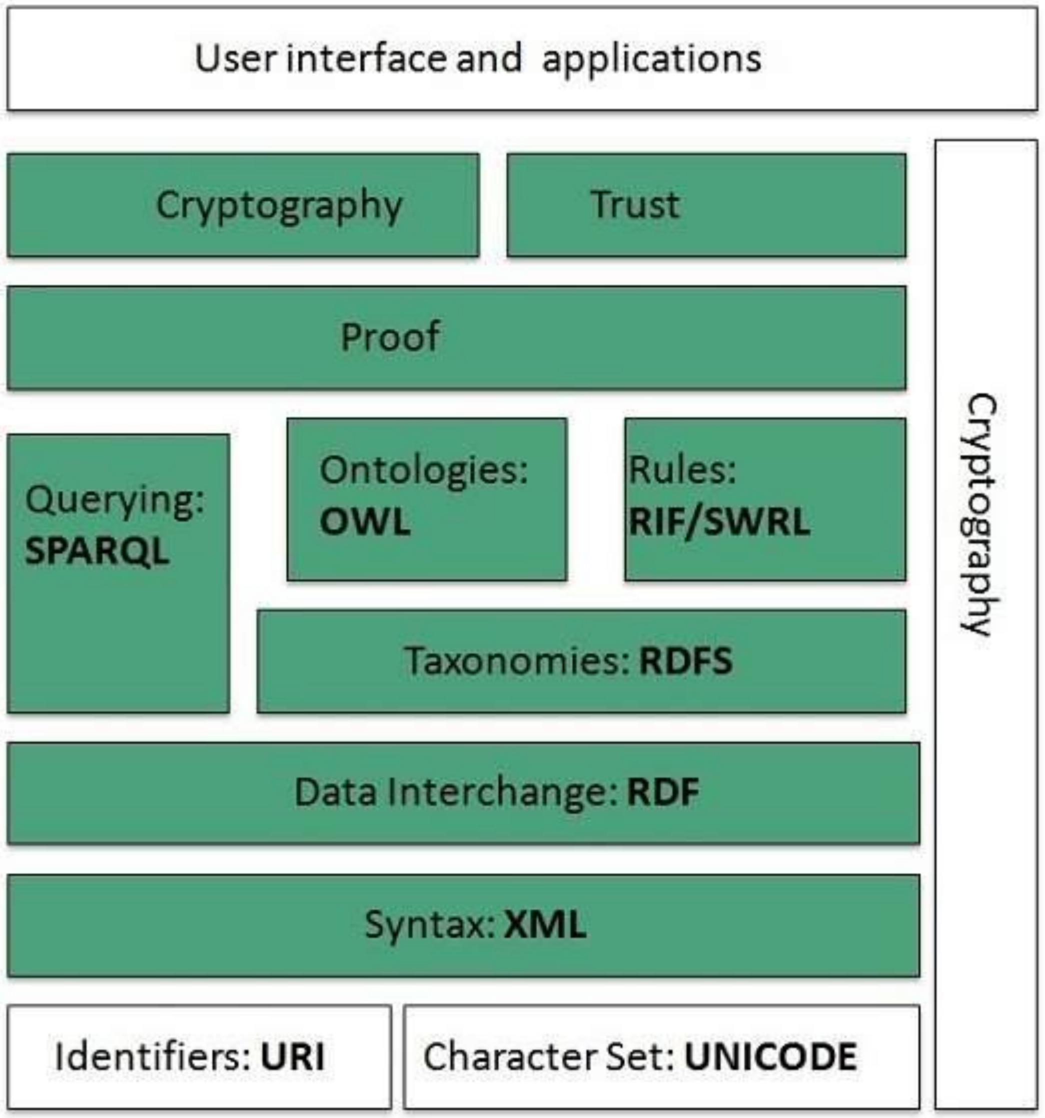
World Wide Web was created by **Timothy Berners Lee** in 1989 at **CERN** in **Geneva.** World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN.** Eventually it became **World Wide Web.**

The following diagram briefly defines evolution of World Wide Web:



WWW Architecture

WWW architecture is divided into several layers as shown in the following diagram:



Identifiers and Character Set

Uniform Resource Identifier (URI) is used to uniquely identify resources on the web and **UNICODE** makes it possible to built web pages that can be read and write in human languages.

Syntax

XML (Extensible Markup Language) helps to define common syntax in semantic web.

Data Interchange

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

Taxonomies

RDF Schema (RDFS) allows more standardized description of **taxonomies**and other **ontological** constructs.

Ontologies

Web Ontology Language (OWL) offers more constructs over RDFS. It comes in following three versions:

- OWL Lite for taxonomies and simple constraints.
- OWL DL for full description logic support.
- OWL for more syntactic freedom of RDF

Rules

RIF and **SWRL** offers rules beyond the constructs that are available from **RDFs** and **OWL.** Simple Protocol and **RDF Query Language (SPARQL)** is SQL like language used for querying RDF data and OWL Ontologies.

Proof

All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

Cryptography

Cryptography means such as digital signature for verification of the origin of sources is used.

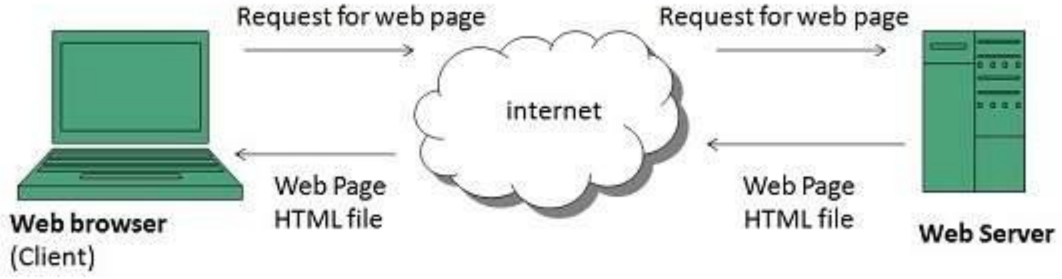
User Interface and Applications

On the top of layer **User interface and Applications** layer is built for user interaction.

WWW Operation

WWW works on client- server approach. Following steps explains how the web works:

1. User enters the URL (say, **http://www.tutorialspoint.com**) of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.



Future

There had been a rapid development in field of web. It has its impact in almost every area such as education, research, technology, commerce, marketing etc. So the future of web is almost unpredictable.

Apart from huge development in field of WWW, there are also some technical issues that W3 consortium has to cope up with.

User Interface

Work on higher quality presentation of 3-D information is under deveopment. The W3 Consortium is also looking forward to enhance the web to full fill requirements of global communities which would include all regional languages and writing systems.

Technology

Work on privacy and security is under way. This would include hiding information, accounting, access control, integrity and risk management.

Architecture

There has been huge growth in field of web which may lead to overload the internet and degrade its performance. Hence more better protocol are required to be developed.

Web Page

web page is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser.

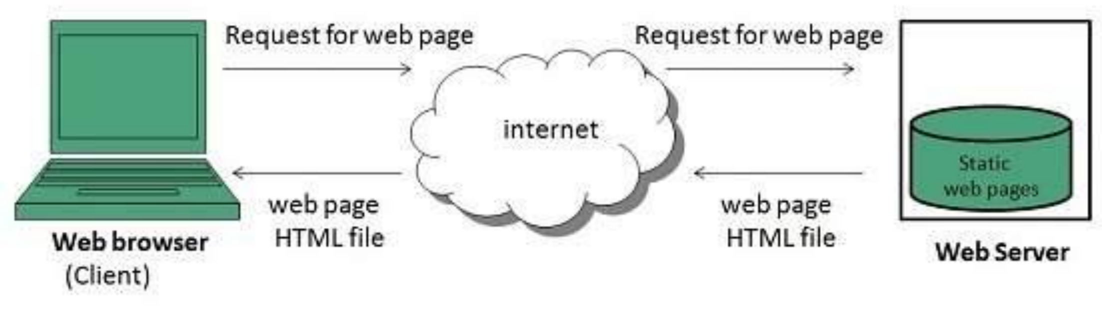
A web page can cotain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages.

Collection of linked web pages on a web server is known as **website.**There is unique **Uniform Resource Locator (URL)** is associated with each web page.

Static Web page

Static web pages are also known as flat or stationary web page. They are loaded on the client's browser as exactly they are stored on the web server. Such web pages contain only static information. User can only read the information but can't do any modification or interact with the information.

Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.



Dynamic Web page

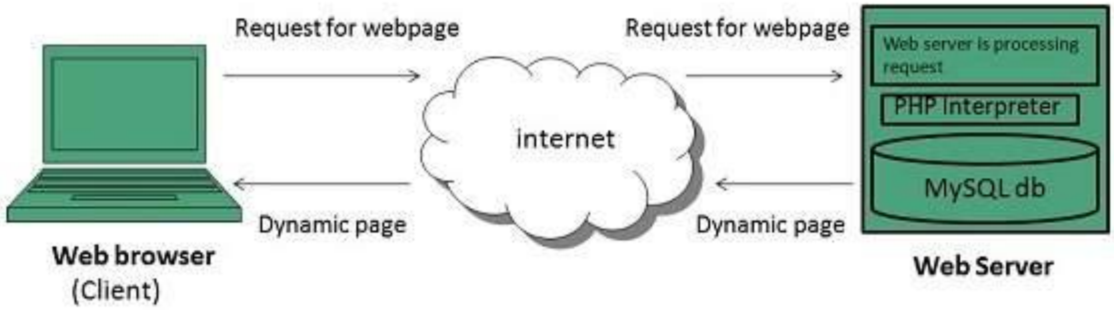
Dynamic web page shows different information at different point of time. It is possible to change a portaion of a web page without loading the entire web page. It has been made possible using **Ajax** technology.

SERVER-SIDE DYNAMIC WEB PAGE

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also include setting up of more client-side processing.

CLIENT-SIDE DYNAMIC WEB PAGE

It is processed using client side scripting such as JavaScript. And then passed in to **Document Object Model (DOM).**



Scripting Laguages

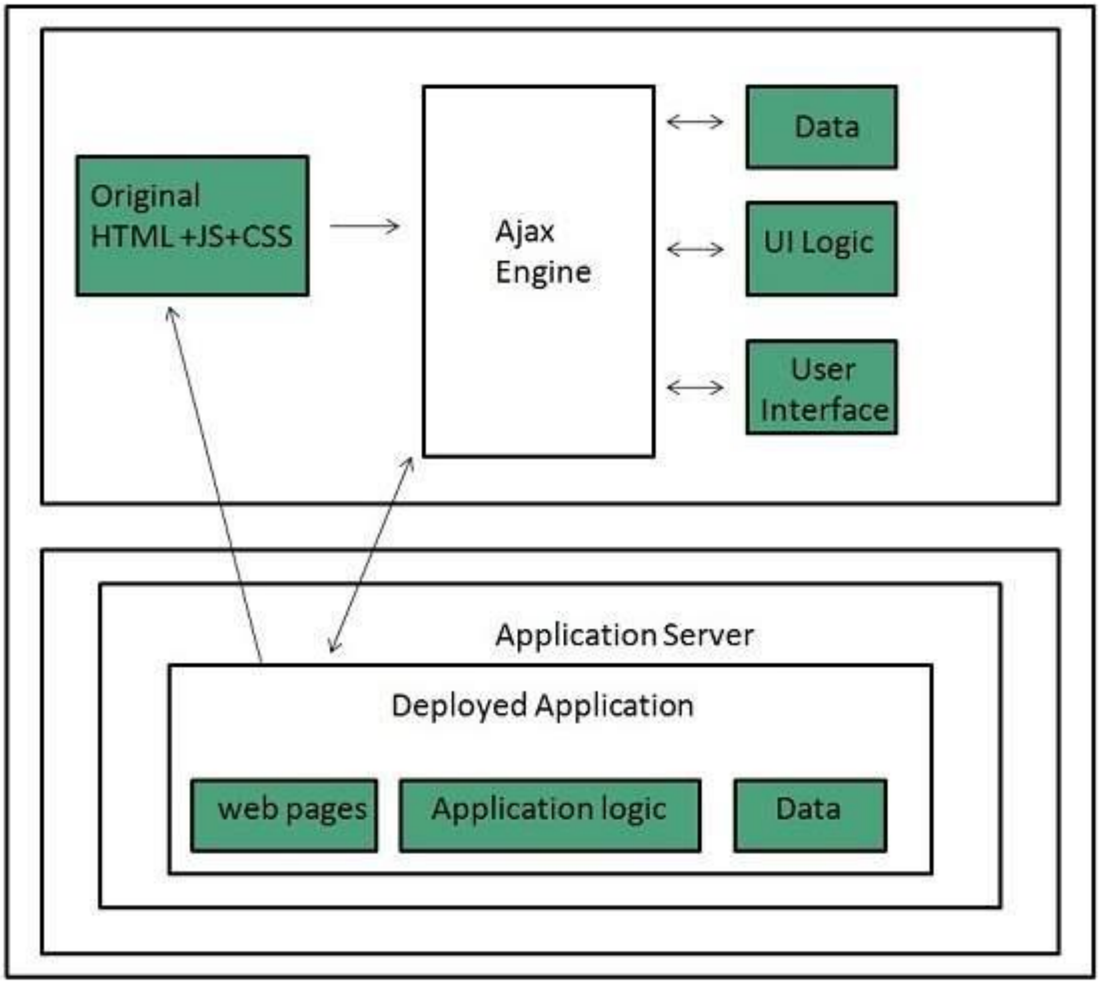
Scripting languages are like programming languages that allow us to write programs in form of script. These scripts are interpreted not compiled and executed line by line.

Scripting language is used to create dynamic web pages.

Client-side Scripting

Client-side scripting refers to the programs that are executed on client-side. Client-side scripts contains the instruction for the browser to be executed in response to certain user's action.

Client-side scripting programs can be embedded into HTML files or also can be kept as separate files.



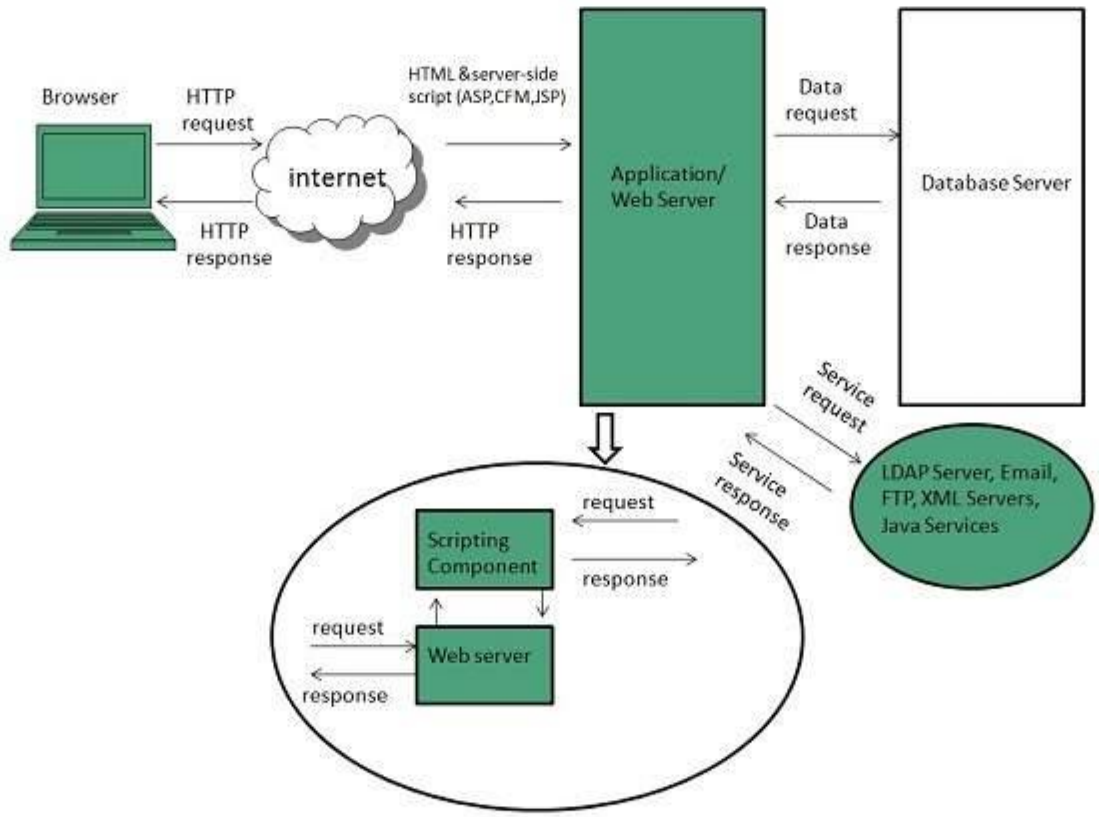
Following table describes commonly used Client-Side scripting languages:

S.N.	Scripting Language Description
1.	**JavaScript** It is a prototype based scripting language. It inherits its naming conventions from java. All java script files are stored in file having **.js**extension.
2.	**ActionScript**It is an object oriented programming language used for the development of websites and software targeting Adobe flash player.

3.	**Dart** It is an open source web programming language developed by Google. It relies on source-to-source compiler to JavaScript.
4.	**VBScript** It is an open source web programming language developed by Microsoft. It is superset of JavaScript and adds optional static typing class-based object oriented programming.

Server-side Scripting

Sever-side scripting acts as an interface for the client and also limit the user access the resources on web server. It can also collects the user's characteristics in order to customize response.



Following table describes commonly used Server-Side scripting languages:

S.N.	Scripting Language Description

1.	**ASP** Active Server Pages (ASP)is server-side script engine to create dynamic web pages. It supports **Component Object Model (COM)** which enables ASP web sites to access functionality of libraries such as DLL.
2.	**ActiveVFP** It is similar to PHP and also used for creating dynamic web pages. It uses native **Visual Foxpro** language and database.
3.	**ASP.net** It is used to develop dynamic websites, web applications, and web services.
4.	**Java** Java Server Pages are used for creating dynamic web applications. The Java code is compiled into byte code and run by **Java Virtual Machine (JVM).**
5.	**Python** It supports multiple programming paradigms such as object-oriented, and functional programming. It can also be used as non-scripting language using third party tools such as **Py2exe** or **Pyinstaller.**
6.	**WebDNA** It is also a server-side scripting language with an embedded database system.

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 a 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 are 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 browser must exhibit are listed below:

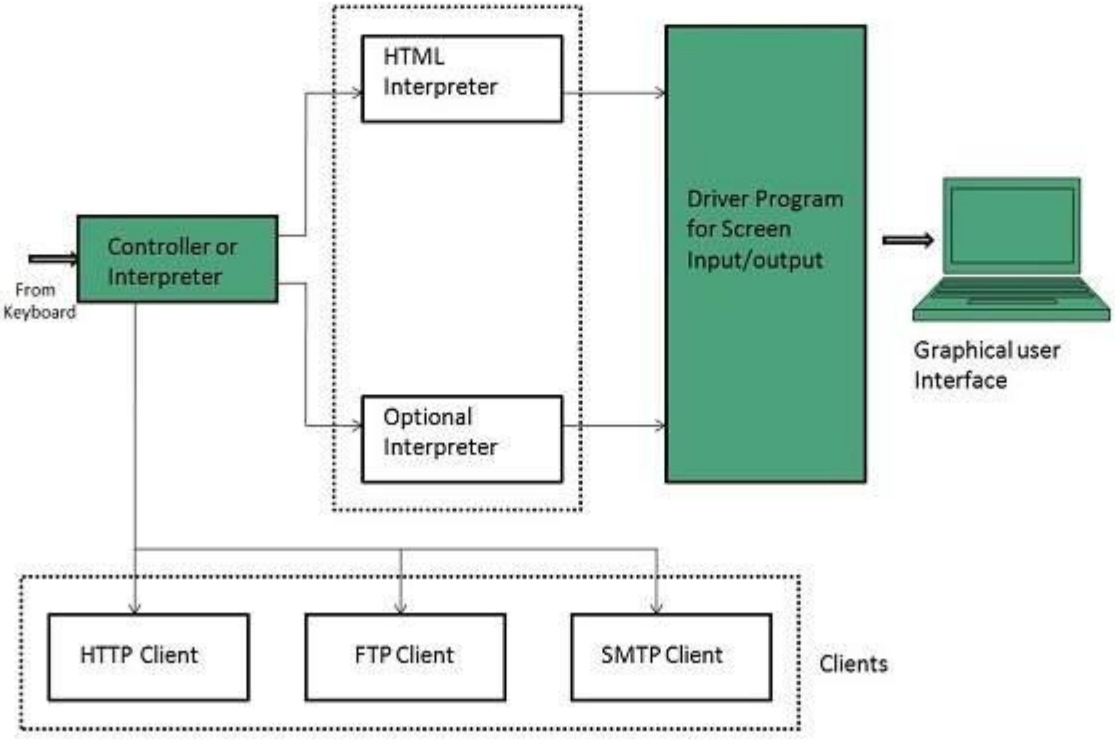
- Controller/Dispatcher
- Interpreter
- Client Programs

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 execute 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 tat are commonly used:

- HTTP
- SMTP
- FTP
- NNTP
- POP

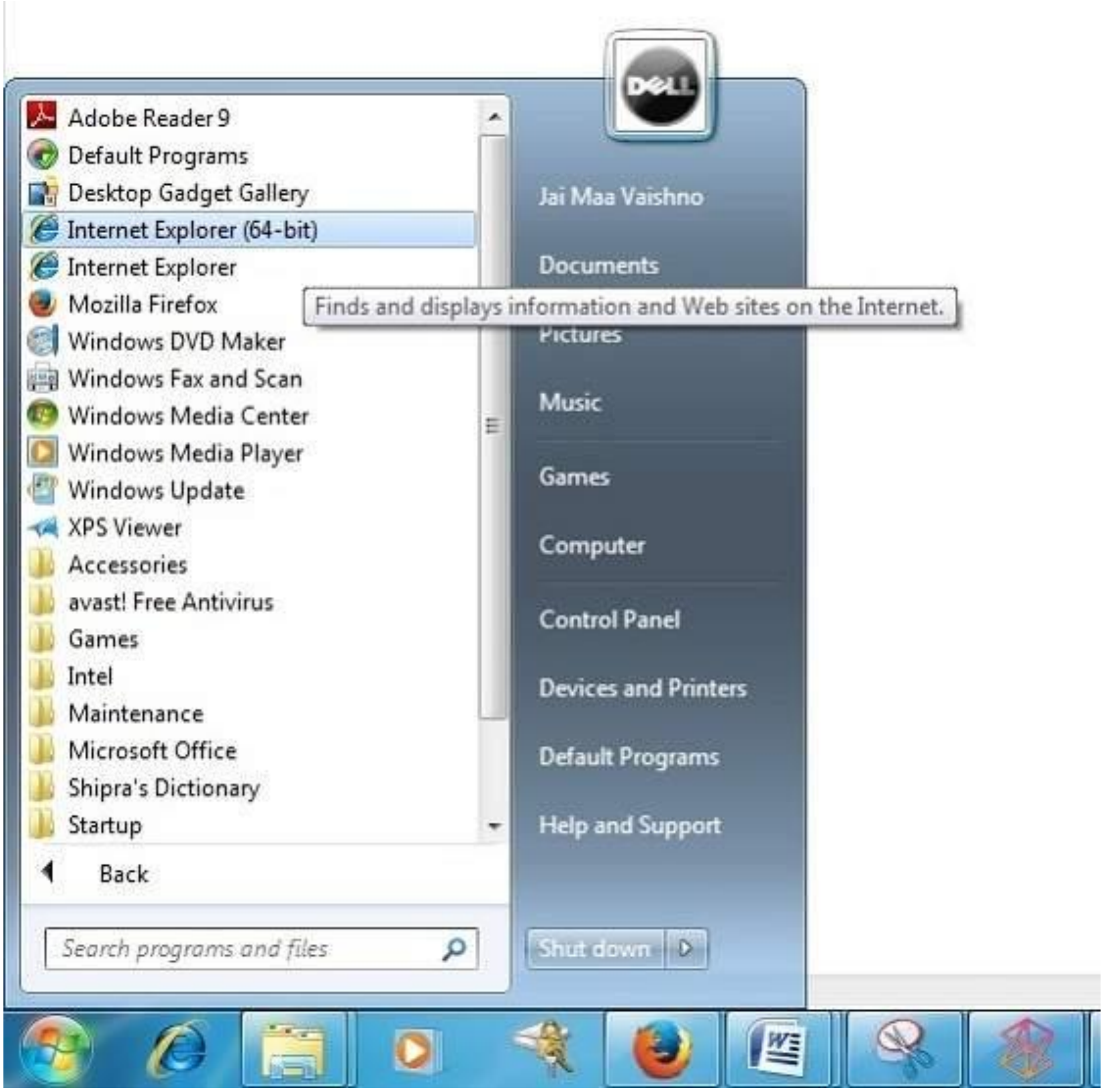


Starting Internet Explorer

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

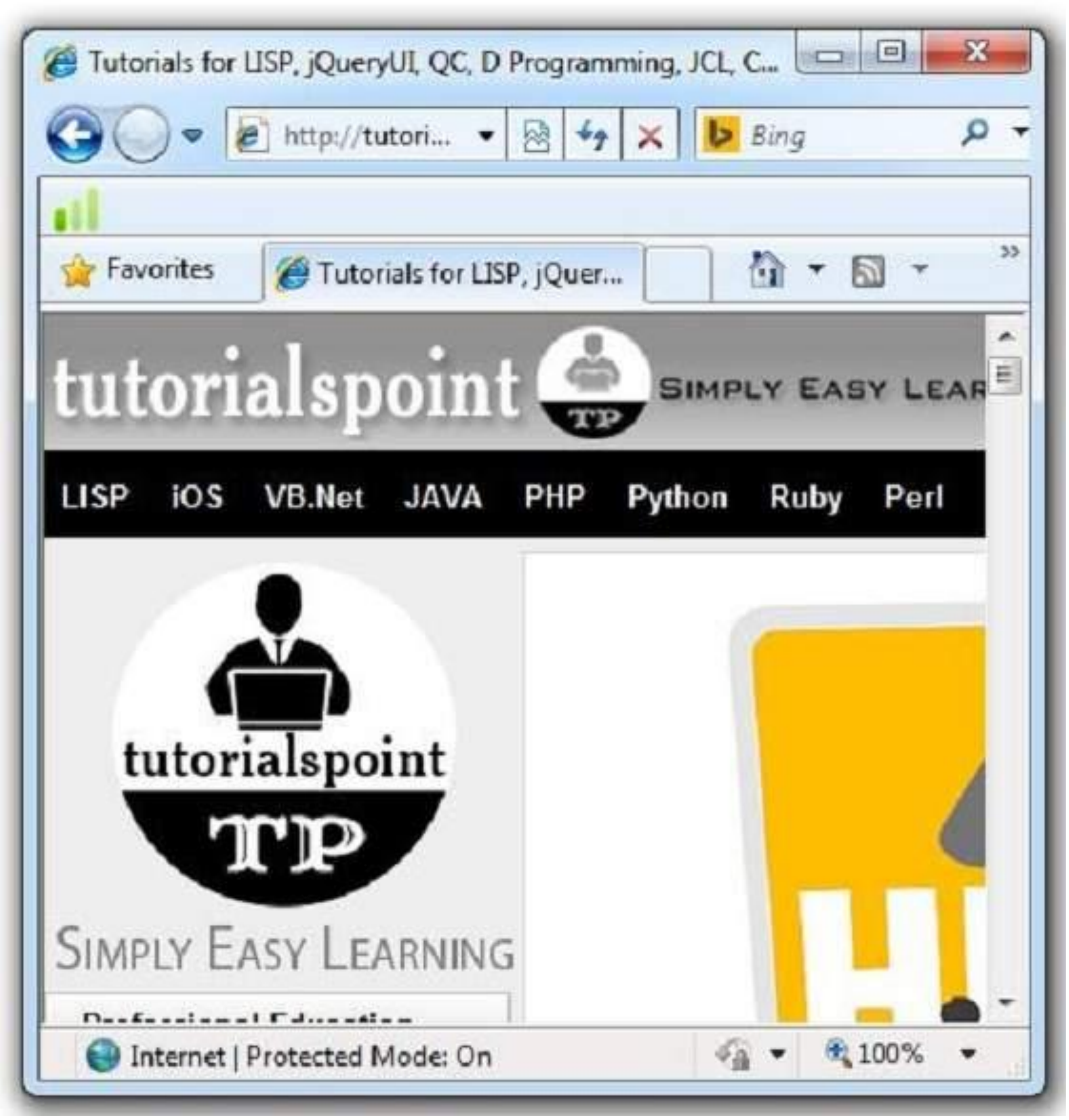
To start internet explorer, follow the following steps:

- Go to **Start** button and click **Internet Explorer.**



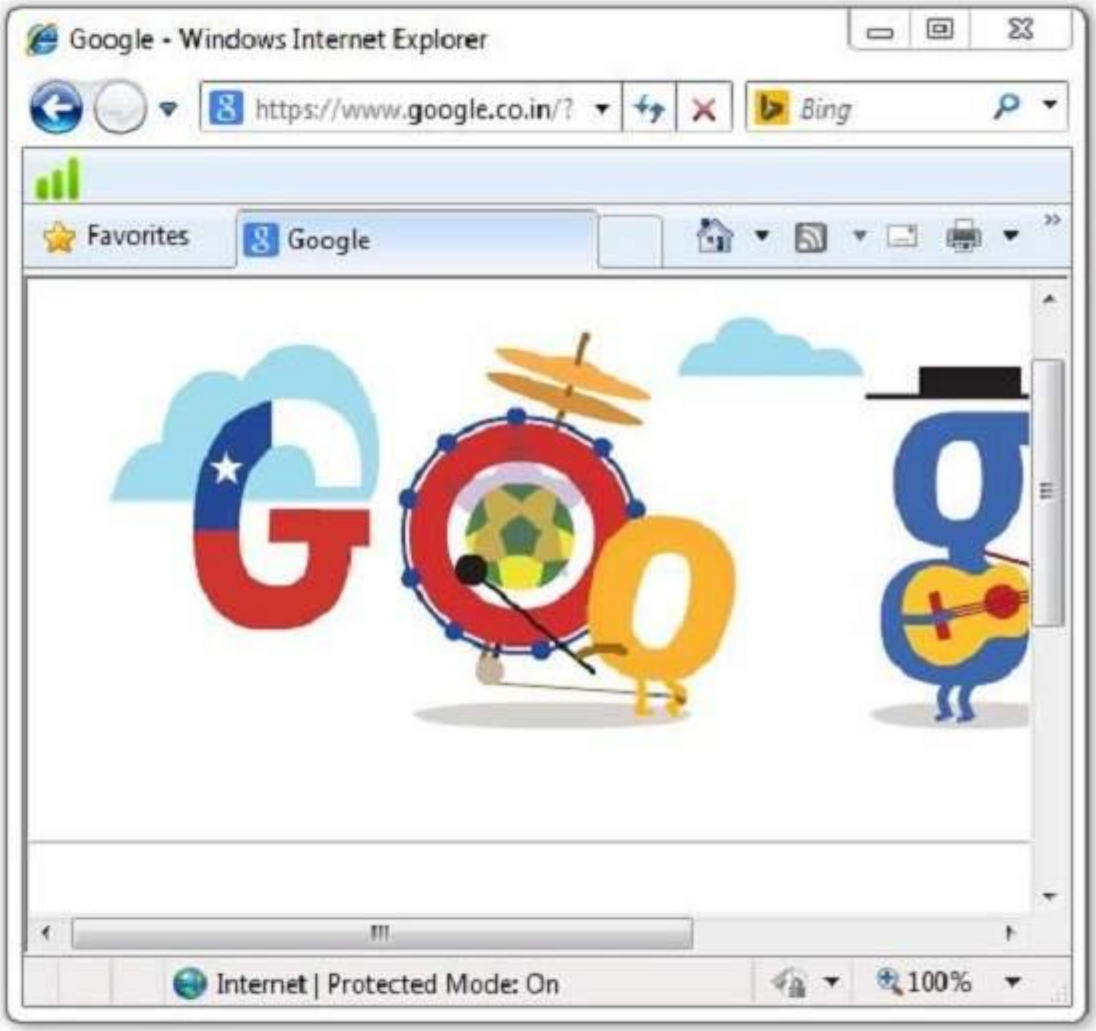
Finds and displays information and Web sites on the Internet.

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



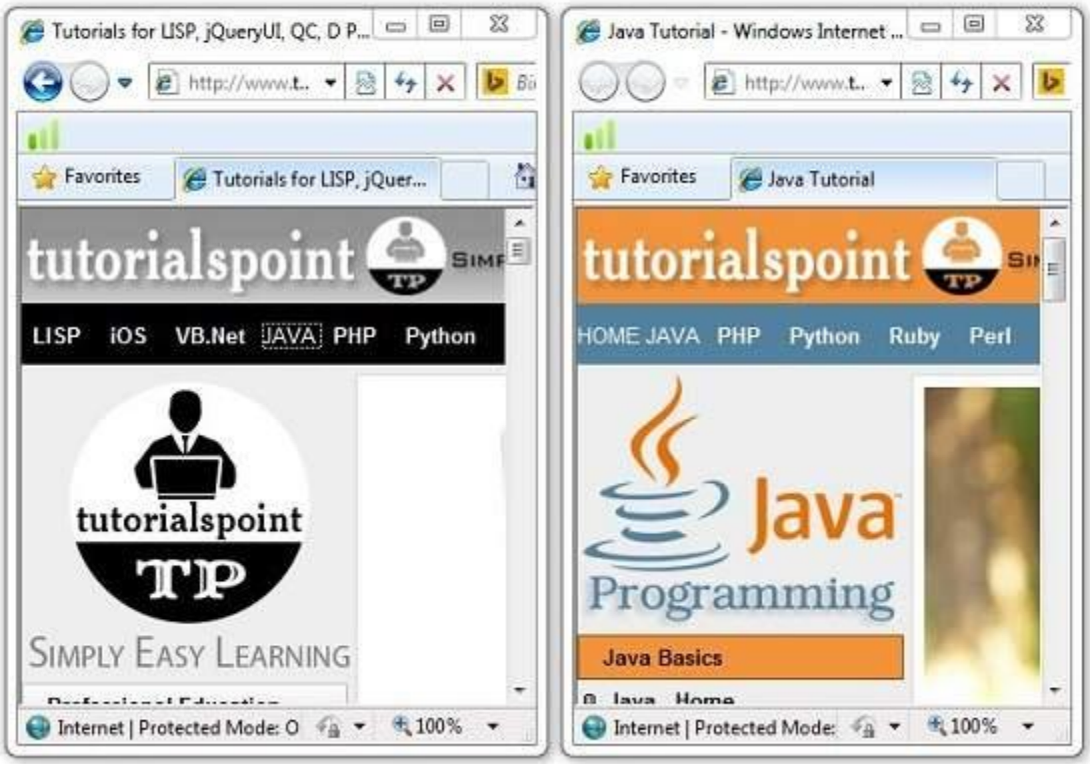
Accessing Web Page

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


Google - Windows Internet Explorer
https://www.google.co.in/?
Bing
Favorites
Google
Internet | Protected Mode: On
100%

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.


Tutorials for LISP, jQueryUI, QC, D P...
http://www.t...
Favorites
Tutorials for LISP, jQuer...
tutorialspoint
LISP iOS VB.Net JAVA PHP Python
tutorialspoint
TP
SIMPLY EASY LEARNING
Internet | Protected Mode: O
100%
Java Tutorial - Windows Internet ...
http://www.t...
Favorites
Java Tutorial
tutorialspoint
HOME JAVA PHP Python Ruby Perl
Java
Programming
Java Basics
Internet | Protected Mode:
100%

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:

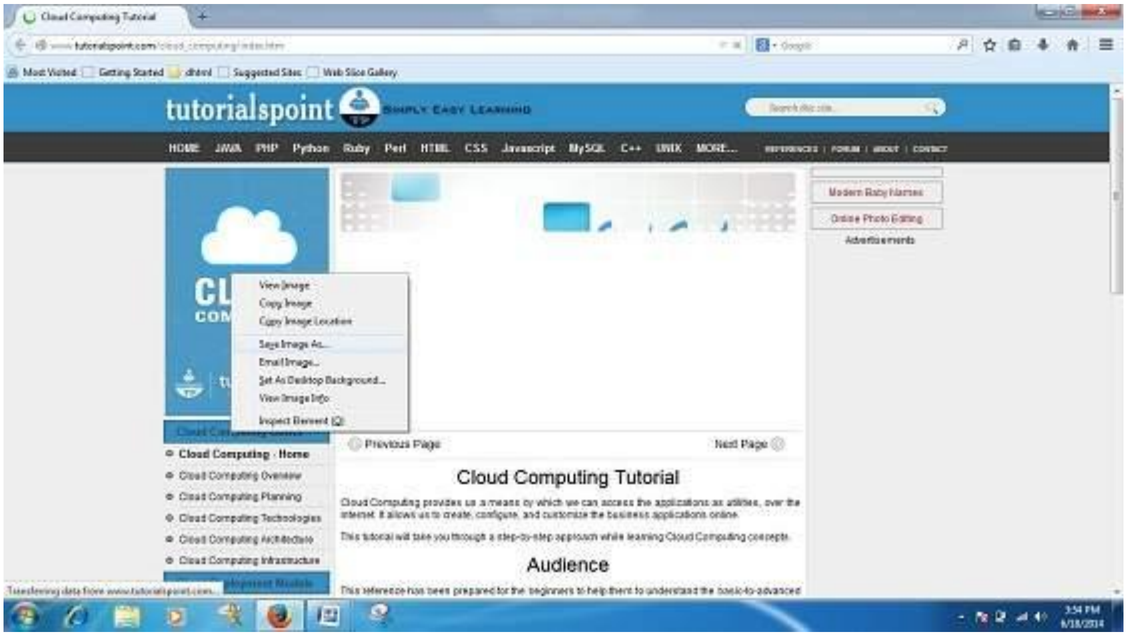
- Click **File > Save As.** Save Webpage dialog box appears.
- Choose the location where you want to save your webpage from **save in:** list box. Then choose the folder where you want to save the webpage.
- Specify the file name in the **File name** box.
- Select the type from **Save as** type list 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.



Save Picture As: This option let you save the picture at specific location with its name. When you click this option, a dialog box is opened where you can sepcify its name and location.

Favourites

The Favourites option helps to save addresses of the webpages you visited oftenly. Hence you need not to remember long and complex address of websites you visit often.

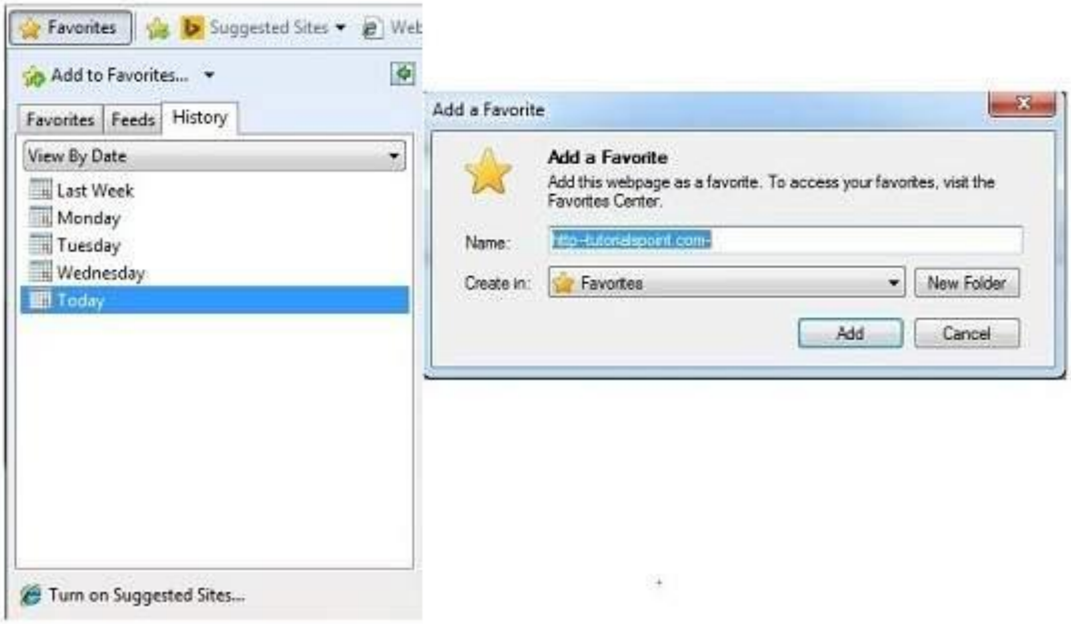
In order to open any webpage, you just need to double click on the webpage that you have marked from bookmarks list.

ADDING A WEB PAGE TO YOUR FAVOURITES

In ordered to add website to your favourite list, follow the steps given below:

- Open webpage that you want to add to your favourite.
- Click on **favourite menu** and then click on **Add to Favourites** opton. **Addfavourites** dialog box appears.

You can also click **Favourites** button available in the toolbar. Favourites panel will open in the left corner of the internet explorer window.
Click **add** button, **AddFavourites** dialog box will apppear.



Add this webpage as a favorite. To access your favorites, visit the Favorites Center.

Name: http--tutorialspoint.com-

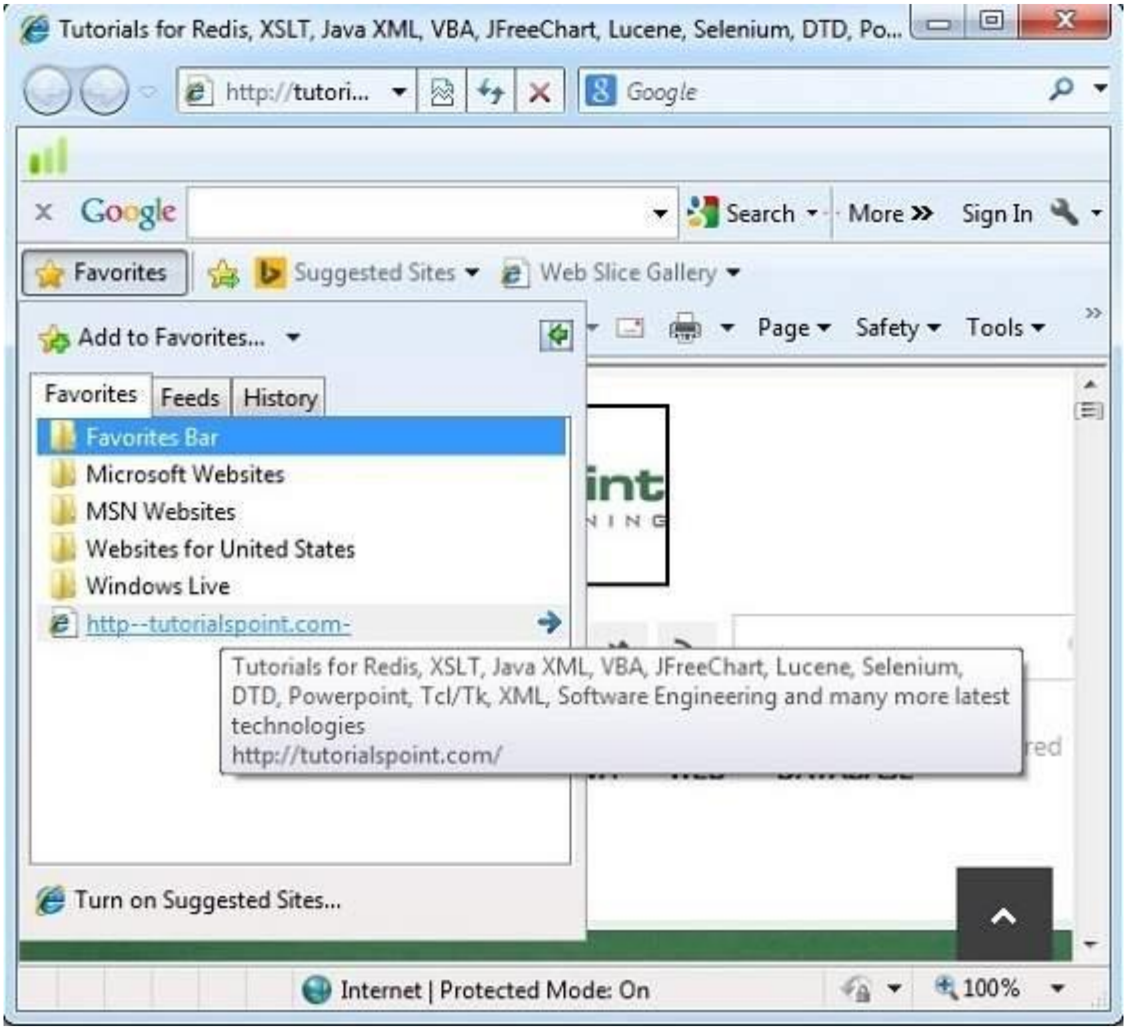
Create in: Favorites

- In **AddFavourites** dialog box, the **Name:** text box will contains the name of the web page that you want to add to favourites.
- Click the **Create in** button, Favoutites folder will appear. Move to the folder where you want to store the favourites by clicking on the folder name.
- Now click **OK** button to save the favourites.

OPENING FAVOURITES

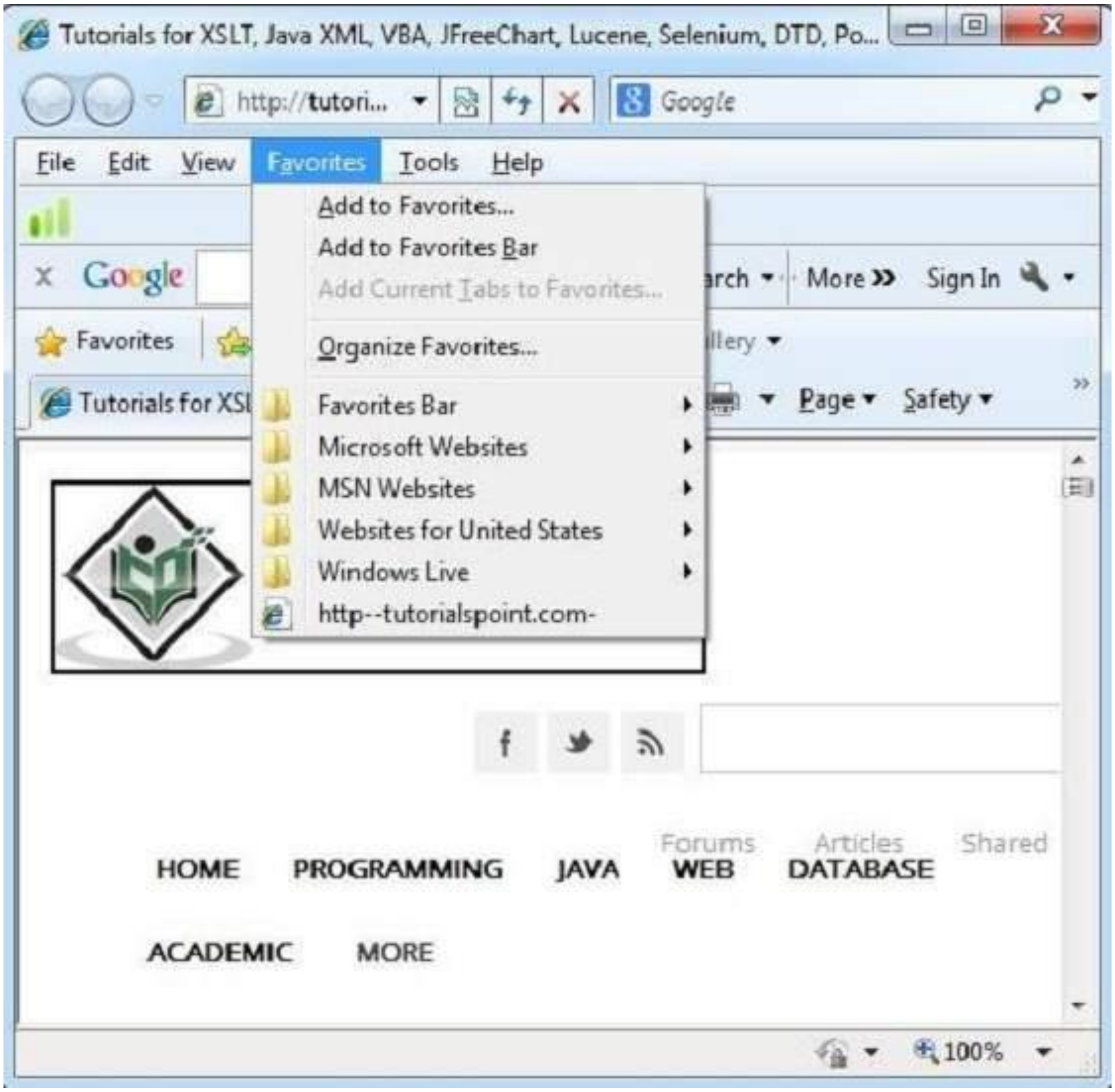
In order to open favourites, follow the steps given below:

- In the Favourite Panel, take the mouse over the site that you want to open. Now click on the address to open that site.



Tutorials for Redis, XSLT, Java XML, VBA, JFreeChart, Lucene, Selenium, DTD, Powerpoint, Tcl/Tk, XML, Software Engineering and many more latest technologies
http://tutorialspoint.com/

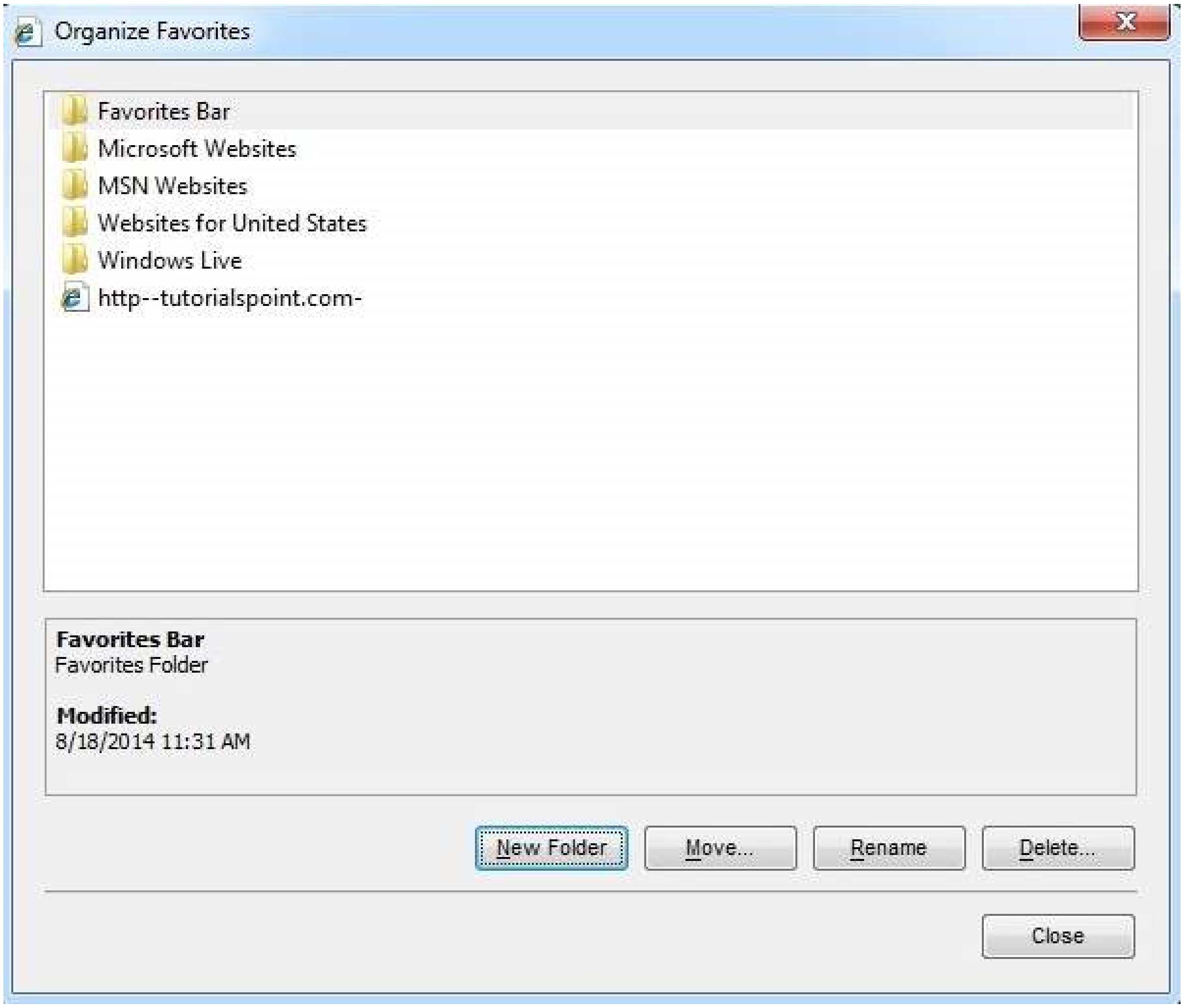
- Favourite can also be opened from the **Favourites** menu by selecting the appropriate one.



ORGANIZING FAVOURITES

Favourites can be organized by categorizing web pages, creating folder for each category and then storing web pages into them. In order to organize favourites, follow the steps given below:

- Click **Favourites menu > Organize Favourites.** Organize favourites dialog box will appears.
- In order to organize the webpages, drag the individual webpage to the respective folder. Similarly to delete a favourite, Click on **delete**button.



Favorites Bar
Microsoft Websites
MSN Websites
Websites for United States
Windows Live
http--tutorialspoint.com-

Favorites Bar
Favorites Folder

Modified:
8/18/2014 11:31 AM

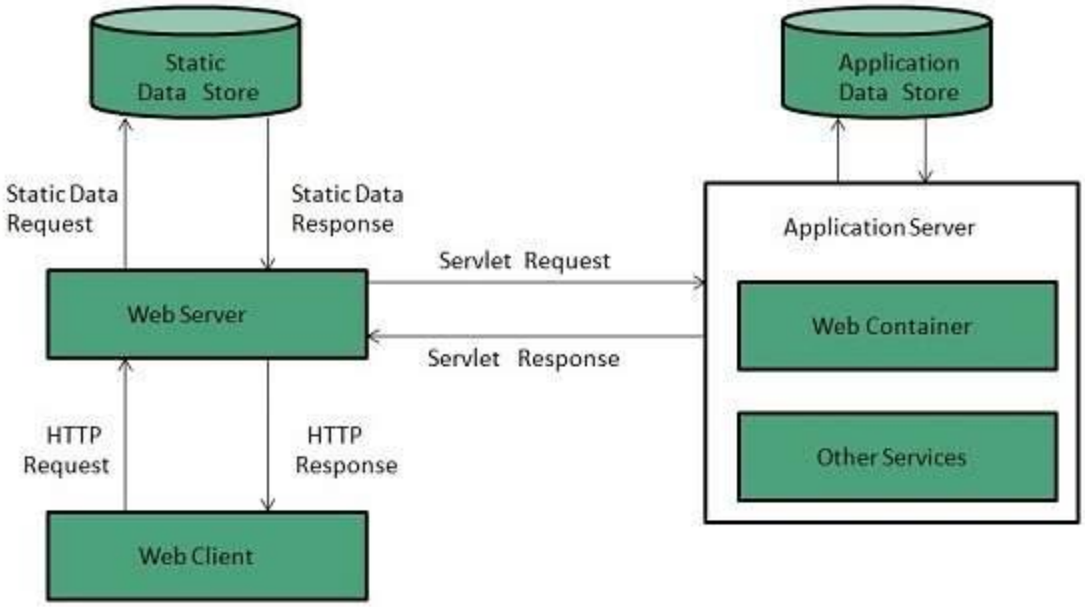
Web server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

Web site is collection of web pages whileweb server is a software that respond to the request for web resources.

Web Server Working

Web server respond to the client request in either of the following two ways:

- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database



Key Points

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an **HTTP response:Error 404 Not found.**
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Architecture

Web Server Architecture follows the following two approaches:

1. Concurrent Approach
2. Single-Process-Event-Driven Approach.

Concurrent Approach

Concurrent approach allows the web server to handle multiple client requests at the same time. It can be achieved by following methods:

- Multi-process
- Multi-threaded

- Hybrid method.

Multi-processing

In this a single process (parent process) initiates several single-threaded child processes and distribute incoming requests to these child processes. Each of the child processes are responsible for handling single request.

It is the responsibility of parent process to monitor the load and decide if processes should be killed or forked.

Multi-threaded

Unlike Multi-process, it creates multiple single-threaded process.

Hybrid

It is combination of above two approaches. In this approach multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in single process results in less load on system resources.

Examples

Following table describes the most leading web servers available today:

S.N.	Web Server Descriptino
1	**Apache HTTP Server** This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.
2.	**Internet Information Services (IIS)** The Internet Information Server (IIS) is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms (and may be on upcoming new Windows version also). IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.
3.	**Lighttpd** The lighttpd, pronounced lighty is also a free web server that is distributed

	with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.
4.	**Sun Java System Web Server** This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on Windows, Linux and UNIX platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, and Ruby on Rails, ASP and Coldfusion etc.
5.	**Jigsaw Server** Jigsaw (W3C's Server) comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, UNIX, Windows, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

web standards

Web standards are the technologies we use to build web sites. These standards exist as long technical documents called specifications, which detail exactly how the technology should work. These documents are not very useful for learning how to use the technologies they describe (this is why we have sites like MDN Web Docs), but instead are intended to be used by software engineers to implement these technologies (usually in web browsers).

For example, the HTML Living Standard describes exactly how HTML (all the HTML elements, and their associated APIs, and other surrounding technologies) should be implemented.

Web standards are created by standards bodies — institutions that invite groups of people from different technology companies to come together and agree on how the technologies should work in the best way to fulfill all of their use cases. The W3C is the best known web standards body, but there are others such as the WHATWG (who maintain the living standards for the HTML language), ECMA (who publish the standard for ECMAScript, which JavaScript is based on), Khronos (who publish technologies for 3D graphics, such as WebGL), and others.

"Open" standards

One of the key aspects of web standards, which TimBL and the W3C agreed on from the start, is that the web (and web technologies) should be free to both contribute and use, and not encumbered by patents/licensing. Therefore anyone can write the code to build a website for free, and anyone can contribute to the standards creation process, where the specs are written.

Because web technologies are created openly, in collaboration between many different companies, it means that no one company gets to control them, which is a really good thing. You wouldn't want a single company suddenly deciding to put the entire web behind a paywall, or releasing a new version of HTML that everyone has to buy to continue making web sites, or worse still, just deciding they aren't interested any more and just turning it off.

This allows the web to remain a freely-available public resource.

Overview of modern web technologies/Standard

There are a number of technologies to learn if you want to be a front-end web developer. In this section we will describe them briefly. For a more detailed explanation of how some of them work together, read our article How the web works.

Browsers

You are probably reading these words inside a web browser in this very moment (unless you've printed it out, or are using assistive technology, such as a screen reader to read it out to you). Web browsers are the software programs people use to consume the web, and include Firefox, Chrome, Opera, Safari, and Edge.

HTTP

Hypertext Transfer Protocol, or HTTP, is a messaging protocol that allows web browsers to communicate with web servers (where web sites are stored). A typical conversation goes something like

```
"Hello web server. Can you give me the files I need to render bbc.co.uk"?

"Sure thing web browser — here you go"

[Downloads files and renders web page]
```

The actual syntax for HTTP messages (called requests and responses) is not that human-readable, but this gives you the basic idea.

HTML, CSS, and JavaScript

HTML, CSS, and JavaScript are the main three technologies you'll use to build a website:

- HyperText Markup Language, or **HTML**, is a markup language consisting of different elements you can wrap (mark up) content in to give it meaning (semantics) and structure. Simple HTML looks like this:

```
<h1>This is a top-level heading</h1>

<p>This is a paragraph of text.</p>

<img src="cat.jpg" alt="A picture of my cat">
```

Copy to Clipboard

If we adopted a house-building analogy, HTML would be like the foundations and walls of the house, which give it structure and hold it together.

- Cascading Style Sheets (**CSS**) is a rule-based language used to apply styles to your HTML — for example, setting text and background colors, adding borders, animating things, or laying out a page in a certain way. As a simple example, the following code would turn our HTML paragraph red:

```
p {
  color: red;
}
```

Copy to Clipboard

In the house analogy, CSS is like the paint, wallpaper, carpets and paintings you'd use to make the house look nice.

- **JavaScript** is the programming language we use to add interactivity to web sites, from dynamic style switching, to fetching updates from the server, right through to complex 3D graphics. The following simple JavaScript will store a reference to our paragraph in memory and change the text inside it:

```
let pElem = document.querySelector('p');
pElem.textContent =  'We changed the text!';
```

Copy to Clipboard

In the house analogy, JavaScript is like the cooker, TV, Microwave, or hairdryer — the things that give your house useful functionality

Tooling

Once you've learned the "raw" technologies that can be used to build web pages (such as HTML, CSS, and JavaScript), you'll soon start to come across various tools that can be used to make your work easier or more efficient. Examples include:

- The developer tools inside modern browsers that can be used to debug your code.
- Testing tools that can be used to run tests to show whether your code is behaving as you intended it to.
- Libraries and frameworks built on top of JavaScript that allow you to build certain types of web site much more quickly and effectively.

- So-called "Linters", which take a set of rules, look at your code, and highlight places where you haven't followed the rules properly.
- Minifiers, which remove all the whitespace from your code files to make it so that they are smaller and therefore download from the server more quickly.

Server-side languages and frameworks

HTML, CSS, and JavaScript are front-end (or client-side) languages, which means they are run by the browser to produce a website front-end that your users can use.

There are another class of languages called back-end (or server-side) languages, meaning that they are run on the server before the result is then sent to the browser to be displayed. A typical use for a server-side language is to get some data out of a database and generate some HTML to contain the data, before then sending the HTML over to the browser to display it to the user.

Example server-side languages include ASP.NET, Python, PHP, and NodeJS.

Web best practices

We have briefly talked about the technologies that you'll use to build websites. Now let's discuss the best practices you should employ to make sure you are using those technologies in the best way that you can.

When doing web development, the main cause of uncertainty comes from the fact that you don't know what combination of technology each user will use to view your web site:

- User 1 might be looking at it on an iPhone, with a small, narrow screen.
- User 2 might be looking at it on a Windows laptop with a widescreen monitor attached to it.
- User 3 might be blind, and using a screen reader to read the web page out to them.
- User 4 might be using a really old desktop machine that can't run modern browsers.

Because you don't know exactly what your users will use, you need to design defensively — make your web site as flexible as possible, so that all of the above users can make use of it, even if they might not all get the same experience. In short, we are trying to make the web work for all, as much as possible.