



01<sup>st</sup> of September 2009

## Web Terminology

The web is an evolving medium where new technologies are introduced practically every new day. The below shows some of the common terminologies associated with the Internet and the web.

### 1. Server

This is a computer where the application/website resides. Every server has a unique **IP** (a combination of 4 number strings separated by a dot e.g. 12.0.0.1) associated with it. Every server on the Internet can be traced by its unique IP. A website hosted on that server is accessed directly via the IP or more commonly via a **domain** (e.g. [www.google.com](http://www.google.com)). Every domain points to a specific IP and therefore from a domain, one can trace on which server the website is hosted. One server may host multiple domains. The server is programmed in such a way that requests for a particular domain are channeled to correctly show the data pertaining to that domain. So when one types out [www.sapagroup.com](http://www.sapagroup.com), it will query the server with Hetzner in Nuremeberg, Germany. The server will then understand that it needs to show the sapnagroup website and the same is displayed on the user's browser.

A server runs on an operating system similar to how a PC runs on an operating system like Windows. But unlike a PC, where Windows rules the roost, many servers run on operating systems like **Linux, Unix besides Windows**. All our servers run on Linux, which is one of the popular operating system for servers.

### 2. Client

A client is typically a software application which queries the server. The server could be accessed over the internet or through a private network. The typical clients would be browsers like **Internet Explorer, Firefox, Safari** or mail applications like **Outlook, Eudora** etc. As detailed above, the client accesses the server via the IP or through a website domain.

### 3. Server side scripts

To explain this concept, let us look at an example. You want to log into your Gmail account. You fill in your username and password. This is called a web form. Web forms are typically used to gather data and pass it to the server. Typical examples include login screens, registration forms, order forms etc. Once you have filled in the details and clicked submit, you will notice that the page submits and momentarily you will see a blank screen. If you see the url box on top where you have typed out [www.gmail.com](http://www.gmail.com), you will notice that now it shows something else. Essentially, the page gets submitted and the data that you have entered is passed to the server which hosts [www.gmail.com](http://www.gmail.com).

At this point, the server now needs to check if the username and password entered is correct or not. Come in the **server script**. This script takes the submitted data and checks it with the data that resides in the database. If there is a match for the username and password entered, it gets the details (in this case the emails in Gmail) of that user and shows it on the screen. Typically, a page refreshes when you click on a link or submit a form. On every refresh, the server side script comes into play.

Server side scripts are written in programming languages like **PHP, ASP, JSP** etc. PHP is one of the most popular server side scripting languages and is used by us in all our scripts. The data is stored in databases. Popular databases are **Mysql, Oracle, SQL Server**. Mysql is typically the database used in conjunction with PHP.

#### 4. Client side scripts

You must have noticed that when you fill a registration form and forget to fill a key entry like name, a prompt box comes up telling you that the name is not filled in. Similarly, when you mouseover a button on the website, the button changes colour or you see a pull down menu on mouseover. In all these instances, you notice a change on the page without the page refreshing. This is possible using client side scripting. **Javascript** is one of the most popular client side scripts. It is used for various purposes from showing scrollers on pages to showing pull down menus on mouseover on prompting for mandatory fields in forms. The biggest advantage of client side scripting is that the page does not refresh and hence the user doesn't need to wait for the page to load again.

#### 5. Hybrid scripting

As the web evolved, more and more applications started moving from the desktop to the web. A classic example was email where web mail clients like Gmail, Hotmail gained popularity over a desktop application like Outlook. The biggest advantage was the "anywhere available" nature of the web. This put pressure on developers to come up with technologies which speeded up the web experience. The biggest nuisance for a web user was the slow loading of pages on every refresh compared to a desktop application. **Ajax** was one such language which sprouted up to better users' web experience. Although Ajax was conceptualized much earlier, it was used extensively first in Gmail. In Gmail, when you compose a mail, it is automatically saved after every 5 seconds or so without you noticing it. The page does not load nor is a blank screen shown. This is done using Ajax. From switching folders to allocating labels to starring mails, all of this is handled through Ajax.

Ajax uses a combination of client side scripting language like **Javascript** to send data to the server in the background without refreshing the page. A typical case is the registration form which we discussed above. A conventional form saves the filled in data only once you click on the submit button. An ajax based form would save as you type. So if you type your name and close the browser window, the name would still be saved despite you not having clicked the submit button. This makes the entire web experience faster. With the use of Ajax, a lot of functions like tabbing, drag and drop etc. have now been ported to the web. It also opens up opportunities to develop applications which look and work the same way as a desktop application.

#### 6. Open source

Open source describes a broad general type of software license that makes source code available to the general public with relaxed or non-existent copyright restrictions. Source code is basically the code which is written to make your software work. When the software is bundled for distribution the source code is not a part of it in most commercial softwares.

Early examples of open source initiatives were for the development for "ARPNET" (which later developed into the Internet) in the 60's. The developers used a process called "Request for comment" to develop network protocols over the phone lines. IBM too in the early 50's and 60's shared its source code for its Operating System. The term open source was coined during a session in California, in reaction to the source code of Navigator by Netscape was being released.

Perhaps one of the best success stories of open source is the GNU/Linux operating system.

With the code being open source there are many advantages.

- Many peer minds together work across the globe to ensure a bug free software with strong security and functionality. Think about it this way, imagine working on a report and then posting it on the internet asking for comments/idea, you are bound to get some spellings you missed pointed out, some new concepts you can add, or even feedback from experts in the field helping your report to be better.
- Many alternatives/flavours, add-ons to the software to tune it as per your requirements. Linux OS itself has many flavours, there is a light weight version if you want a basic setup which takes as less space as possible, a version which does not install anything on your system, but rather works through a cd, a version for servers, a version more popular for Desktops.
- Software patches, security updates are available instantly after an issue is noted, with so many people working across the globe to test, the chances of a problem being missed out is highly unlikely.

#### 7. Linux

Its an open source operating system (similar to windows XP, windows Vista, Mac OS etc) which is which is Unix-like (Unix is another operating system), initially more popularly used for servers, its fast gaining foot on personal computers with flavours like Ubuntu which are easy to install and are very user friendly in its operation and look similar to Windows or Mac OS, the best part being alternatives to almost every software you need available, eg there is open office for your word process spreadsheets which can open and save Microsoft word format documents. It can be deployed on a variety of computer hardware like embedded devices, mobile phones,

watches, super computers. Linux is preferred because of its stability, security and better performance as compared to other operating systems. Its also an open source operating system which means, thousands of developers contribute to its development, and ensure its success.

Unix was developed at AT&T's Bell Laboratories in 1969, and released in 1971. The GNU Project was initiated in 1983 with the goal of creating a complete "Unix-compatible software system" composed entirely of free software. MINIX was a cheap minimal unix-like operating system designed for education in computer science. Linus Torvalds while attending the University of Helsinki developed a replacement for MINIX which became known as the Linux kernel.

Some advantages of Linux are

- The advantages of open source apply to Linux. Stable and Secure.
- Virus free environment
- Virtually alternatives of every software available as open source

## 8. PHP

PHP or PHP: Hypertext Preprocessor (recursive initialism used, refer to [http://en.wikipedia.org/wiki/Recursive\\_initialism](http://en.wikipedia.org/wiki/Recursive_initialism)), is one of the many languages a computer understands, and will process information/task we seek and give results. Just like how a translator in a meeting will take questions from English speaking members and translate it to German speaking members, get the answers from them and translates the results back to the English speaking members, similarly PHP (or any other language) works like a translator, which translates a set of instructions written by a programmer and sends them to the computer in a language it can understand, and gives the results back. The difference between PHP and other languages is that it's a server side language and hence resides on the server, and because of this can interact with a database like MySQL on the server to give you a dynamic system which can eg

- take form input like a contact us form submission
- store this information in a database for future reference
- send an email to the concerned person etc.

It is estimated that PHP is installed on over **20 million websites and 1 million web servers**.

Initially written in 1994 to maintain a personal homepage by Rasmus Lerdorf it was called "Personal Home Page". The newer versions could soon communicate with database leading to development of dynamic web applications. Its code was then released for bug tracing and development in 1995.

Some advantages of PHP are

- Better performance and reliability
- Development and maintenance costs are low
- Easily embedded into HTML code
- Compatible with servers like Apache and IIS

## 8. MySQL

MySQL stands for "My Structured Query Language" and is a database and can be thought of as an excel sheet which holds raw information. Calculations can be also performed like how we do in excel eg sum the rows etc. The advantage of any database is that we can flexibly ask questions to it and it will give us the information at the fastest possible way. Eg if we have a database which stores information of soccer matches, we can ask it questions like

- how many matches were played in total
- what was the maximum number of goals shot by any team, and in which match
- list all the matches played between 12/07/2009 to 20/08/2009

Etc

MySQL is known to be a web database which is fast.

The original development of MySQL started in 1994 and since then has been rapidly developing to provide more functionality. Support for windows was released soon.

The company MySQL AB (Swedish) now a subsidiary of Sun Microsystems owns and sponsors MySQL.

Recently Oracle Corporation began the process of acquiring Sun Microsystems.

Some of the advantages of using MySQL are

- Scalability on various platforms as compared to other RDBMS. Mysql can run on Windows, Mac OS X, Linux, BSD, UNIX, AmigaOS, and Symbian.
- Supports high traffic, popularity of MySQL can be judged by the various applications esp those that need high traffic. e.g. Include Google (not for search though), Nokia, Facebook, Wikipedia, YouTube and Flickr.
- Integration with PHP, MySQL is so commonly used by PHP that most 3<sup>rd</sup> party applications in PHP require MySQL as the backend.

## 9. Java Script

Javascript was designed to look like Java hence the name. It's a language like PHP, but unlike PHP it's a client side Language, i.e. It runs on your browser (in simple terms). It's useful for performing quick tasks like checking if the form fields have been submitted rather than asking PHP to do this. This helps in 2 ways, it reduces the load on your server as checks like these are done by your browser itself using javascript, and it also increases the speed of such actions as we don't have to fetch anything from the server. Menu effects etc are all done using javascript. Javascript was initially developed for Netscape in the 90s as Mocha, and later renamed to LiveScript before it got its name as Javascript. The introduction of Ajax has ensured Javascript's spotlight and increased its usage.

## 10. AJAX

AJAX stands for asynchronous JavaScript and XML and is a technique in web development on the client side to build interactive web applications. Many parts of a website/online system are common like the logo, the footer, the navigation. Also the formatting of heading, body text is common. In conventional system we have to re-download this information every time we click on a link. Eg if we have the "about us" page open and we click on the "contact us" page, it will re-download the logo, the footer etc all over again.

AJAX helps in reducing what we have to download by keeping information which is common as it is and changing only the blocks we need to change, so in case of the above example, it will keep the page as it is and quickly only download and change the heading text to "Contact us" and the general text below accordingly, and not re-download everything else, hence improving speed. The disadvantage of AJAX is that it's not SEO friendly.

Techniques to achieve similar results were started in the mid 90s with the introduction of Java applets, and IFrame. Benefits of Ajax are quite a few

- Faster more interactive Web application
- Lesser requests to the server as common elements like css, javascripts are already loaded and are not requested again.

Some drawbacks are

- Not SEO friendly
- Harder to develop
- Browser back button might not work
- Difficult to bookmark a page