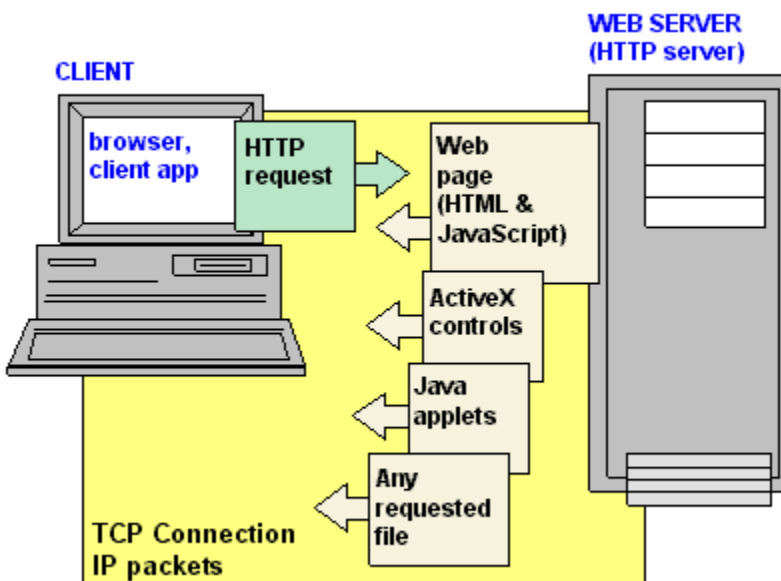


Unit 4

HTTP and Web Services

4.1 Web Servers and Web Access

A Web server is a program that, using the client/server model and the World Wide Web's Hypertext Transfer Protocol (HTTP), serves the files that form Web pages to Web users (whose computers contain HTTP clients that forward their requests). Every computer on the Internet that contains a Web site must have a Web server program. Two leading Web servers are Apache , the most widely-installed Web server, and Microsoft's Internet Information Server (IIS).



As is the case with any client-server communication, in this case also the client (i.e., the Web Browser) and the server (i.e., HTTP/Web Server) should be able to communicate with each other in a defined way. These pre-defined sets of rules which form the basis of the communication are normally termed as a protocol and in this case the underlying protocol will be HTTP.

Irrespective of how the client or the server has been implemented, there will always be a way to form a valid HTTP Request for the client to work and similarly the server needs to be capable of understanding the HTTP Requests sent to it and form valid HTTP Responses to all the arrived HTTP Requests. Both the client and the server machines should also be equipped with the capability of establishing the connection to each other (in this case it'll be a TCP reliable

connection) to be able to transfer the HTTP Requests (client -> server) and HTTP Responses (server -> client).

The following steps explain how a web server and web application server work together to process a page request:

1. The user requests a page by typing a URL in a browser, and the web server receives the request.
2. The web server looks at the file extension to determine whether a web application server must process the page. Then, one of the following actions occur:
 1. If the user requests a file that is a simple web page (often one with an HTM or HTML extension), the web server fulfills the request and sends the file to the browser.
 2. If the user requests a file that is a page that a web application server must process (one with a CFM, PHP, JSP, or CFC extension for ColdFusion requests), the web server passes the request to the web application server. The web application server processes the page and sends the results to the web server, which returns those results to the browser.

Aside from its functions listed above, the Web server also has an additional number of responsibilities. Whereas a Web browser simply translates and displays data it is fed, a Web server is responsible for distinguishing between various error and data types. A Web server must, for example, designate the proper code for any sort of internal error and send that back to the browser immediately after it occurs. It also has to distinguish between various elements on a Web page (such as .GIFs, JPEGS and audio files) so that the browser knows which files are saved in which format. Depending on the site's function, a Web server may also have numerous additional tasks to handle, including logging statistics, handling security and encryption, serving images for other sites (for banners, pictures, etc), generating dynamic content, or managing e-commerce functions.

4.2 Universal Naming with URLs

A uniform resource locator, abbreviated URL, also known as web address, is a specific character string that constitutes a reference to a resource. In most web browsers, the URL of a web page is displayed on top inside an address bar. An example of a typical URL would be "http://en.example.org/ ". A URL is technically a type of uniform resource identifier (URI), but in many technical documents and verbal discussions, URL is often used as a synonym for URI, and this is not considered a problem

URL strings consist of three parts (substrings):

1. Network protocol
2. Host name or address
3. File or resource location

These substrings are separated by special characters as follows:

protocol :// host / location

URL Protocol

The 'protocol' substring defines a network protocol to be used to access a resource. These strings are short names followed by the three characters '://' (a simple naming convention to denote a protocol definition). Typical URL protocols include http://,ftp://.

URL Host

The 'host' substring identifies a computer or other network device. Hosts come from standard Internet databases such as DNS and can be names or IP addresses. For example, google.com is the host for the Web page.

URL Location

The 'location' substring contains a path to one specific network resource on the host. Resources are normally located in a host directory or folder. For example,/od/internetaccessbestuses/bldef-url.htm is the location of this Web page including two subdirectories and the file name.

Absolute vs. Relative URLs

Full URLs featuring all three substrings are called absolute URLs. In some cases such as within Web pages, URLs can contain only the one location element. These are called relative URLs. Relative URLs are used for efficiency by Web servers and a few other programs when they already know the correct URL protocol and host.

4.3. WWW Technology: HTML, DHTML, WML, XML

HTML

Hypertext Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser. HTML defines the structure and layout of a Web document by using a variety of tags and attributes. The correct structure for an HTML document starts with <HTML><HEAD>(enter here what document is about)<BODY> and ends with </BODY></HTML>. All the information you'd like to include in your Web page fits in between the <BODY> and </BODY> tags.

The definition of HTML is Hypertext Markup Language.

- **Hypertext** is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.
- **Markup** is what HTML tags do to the text inside them. They mark it as a certain type of text (italicized text, for example).
- HTML is a **Language**, as it has code-words and syntax like any other language.

HTML is made up of elements (often called tags) that build the contents of a web page. The differences between HTML and other programming languages include:

- HTML is not compiled. It is written and used without any changes being done to it. It starts out a text file, and is still a text file when a browser or user agent interprets it.
- HTML is human readable. While some other programming languages can be read by people (and not just machines), many times you have to learn the language to really understand it.

DHTML (Dynamic HTML)

Dynamic HTML is not really a new specification of HTML, but rather a new way of looking at and controlling the standard HTML codes and commands. DHTML is not a programming language, and technically not a markup language either. You don't need to add a .dhtml extension of a DHTML enabled page. DHTML pages are written in regular HTML/XHTML markup.

Dynamic HTML is only a marketing term invented by Netscape and Microsoft for various technologies abilities included in their version 4 browsers.

Designed to enhance a Web user's experience, DHTML includes the following features:

- Dynamic content, which allows the user to dynamically change Web page content
- Dynamic positioning of Web page elements
- Dynamic style, which allows the user to change the Web page's color, font, size or content

While DHTML enhances the website user's experience, the technology may also be frustrating for users when it is used incorrectly. For example, a website menu with flashy DHTML animations can easily confuse user navigation. Another DHTML issue occurs when Web developers attempt to create cross-browser DHTML, which is very difficult.

There are four parts to DHTML:

- Document Object Model (DOM) (definition)
- Scripts
- Cascading Style Sheets (CSS)
- XHTML

DOM

The DOM is what allows you to access any part of your Web page to change it with DHTML. Every part of a Web page is specified by the DOM and using its consistent naming conventions you can access them and change their properties.

Scripts

Scripts written in either JavaScript or ActiveX are the two most common scripting languages used to activate DHTML. You use a scripting language to control the objects specified in the DOM.

Cascading Style Sheets

CSS is used in DHTML to control the look and feel of the Web page. Style sheets define the colors and fonts of text, the background colors and images, and the placement of objects on the page. Using scripting and the DOM, you can change the style of various elements.

XHTML

XHTML or HTML 4.x is used to create the page itself and build the elements for the CSS and the DOM to work on. There is nothing special about XHTML for DHTML - but having valid XHTML is even more important, as there are more things working from it than just the browser.

For Web developers, DHTML poses the following problems:

- It can be difficult to develop and debug because of lack of Web browser and technological support.
- DHTML scripts may not work correctly in various Web browsers.
- The Web page layout may not display correctly when it is developed to display in different screen size combinations and in different browsers.

WML (Wireless Markup Language)

Wireless Markup Language (WML), based on XML, is a markup language intended for devices that implement the Wireless Application Protocol (WAP) specification, such as mobile phones. It provides navigational support, data input, hyperlinks, text and image presentation, and forms, much like HTML (Hypertext Markup Language). It preceded the use of other markup languages now used with WAP, such as HTML itself, and XHTML (which are gaining in popularity as processing power in mobile devices increases).

WML is analogous to HTML in several ways because it is written in plain text format. However, because wireless devices are not the same in terms of display, processing power and button layout, some features are specific to devices that are incorporated in WML.

The following are some key features of WML as compared to HTML:

- WML is a markup language for small, wireless computing devices.
- In WML, variables can be defined that store data in string format. In HTML, variables cannot be stored.
- WML uses WML script for client-side scripting, which is stored in a separate file. HTML uses JavaScript.
- The supported image format for WML is WBMP. HTML supports JPEG, GIF and BMP.
- A micro-browser is used to run WML markup. A regular browser, such as Internet Explorer, Firefox or Chrome, is used to run HTML markup.
- WML follows XHTML specification and is therefore case sensitive. HTML is not case sensitive.
- WML has fewer tags compared to HTML.

WML-equipped devices have the following characteristics:

- **Display Size:** Devices have a small screen size and low resolution; therefore, WML has to be capable of rendering content regardless of display size.
- **Input:** Small computing devices do not have a mouse or pointer-based navigation devices. They may have a small numeric keypad or a QWERTY keypad based on whether the device is simple or sophisticated. WML has to be capable of obtaining necessary user input regardless of the limitations of the device.
- **Processing:** They have limited-capacity rechargeable batteries with a low-power CPU and low memory. WML browsers should act like thin clients and perform minimal processing on the device.
- **Network Capabilities:** Small computing devices have a low bandwidth and high network latency. WML has to ensure maximum efficiency in fetching requested Web pages from the server

XML (Extensible Markup Language)

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality, and usability over the Internet.[6] It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.

XML, a formal recommendation from the World Wide Web Consortium (W3C), is similar to the language of today's Web pages, the Hypertext Markup Language (HTML). Both XML and HTML contain markup symbols to describe the contents of a page or file. HTML, however, describes the content of a Web page (mainly text and graphic images) only in terms of how it is to be displayed and interacted with.

XML is "**extensible**" because, unlike HTML, the markup symbols are unlimited and self-defining. XML is actually a simpler and easier-to-use subset of the Standard Generalized Markup Language (SGML), the standard for how to create a document structure. It is expected that HTML and XML will be used together in many Web applications. XML markup, for example, may appear within an HTML page.

XML documents use a self-describing and simple syntax:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

The first line is the XML declaration. It defines the XML version (1.0) and the encoding used (ISO-8859-1 = Latin-1/West European character set).

The next line describes the root element of the document (like saying: "this document is a note"):

```
<note>
```

The next 4 lines describe 4 child elements of the root (to, from, heading, and body):

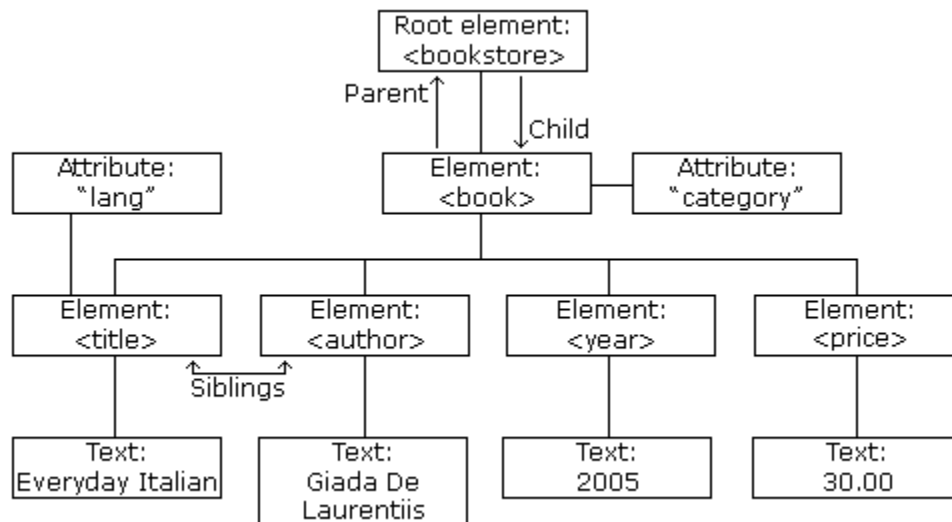
```
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
```

And finally the last line defines the end of the root element:

```
</note>
```

XML documents must contain a **root** element. This element is "the parent" of all other elements.

The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree. The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters).



The image above represents one book in the XML below:


```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

The root element in the example is `<bookstore>`. All `<book>` elements in the document are contained within `<bookstore>`.

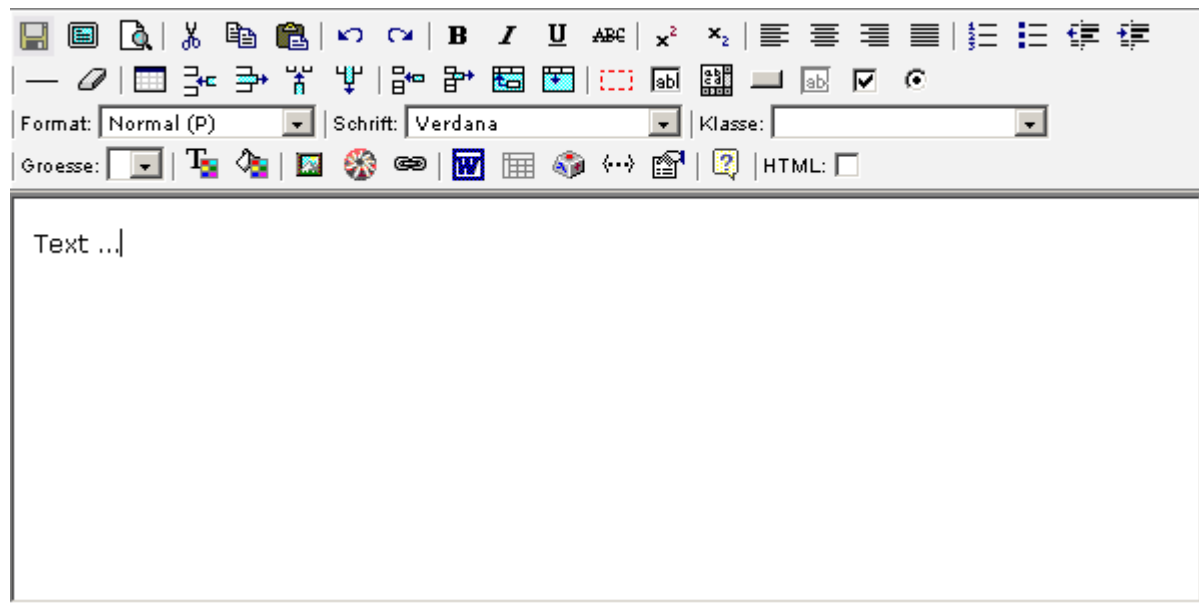
The `<book>` element has 4 children: `<title>`, `<author>`, `<year>`, `<price>`.

4.4. WYSIWYG (What You See Is What You Get)

WYSIWYG implies a user interface that allows the user to view something very similar to the end result while the document is being created. In general WYSIWYG implies the ability to directly manipulate the layout of a document without having to type or remember names of layout commands.

Typically, the design goals of a WYSIWYG application may include the following:

- Provide high-quality printed output on a particular printer
- Provide high-quality printed output on a variety of printers
- Provide high-quality on-screen output
- Allow the user to visualize what the document will look like when printed



Example of WYSIWYG

4.5. Helper applications: CGI; PERL, JAVA, JAVA SRIPTS, PHP, ASP, .NET Applications

PERL

Perl, sometimes referred to as **Practical Extraction and Reporting Language**, is an interpreted programming language with a huge number of uses, libraries and resources. Arguably one of the most discussed and used languages on the internet; it is often referred to as the Swiss army knife, or duct tape, of the web. Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages.

Advantages of PERL:

- 1. Portability:** Perl code that doesn't use system specific features can be run on any platform, and these days almost every operating systems support it. So, for example, if you write an utility that renames files in a complete directory tree according to some rule, you will be able to run it on Windows and Unix (Linux, Solaris, AIX, BSD, etc), and probably Mac OS too. But if you write an utility that changes file permissions, you might run into problems, because file permission are implemented differently on different systems.
- 2. String processing and especially Regular expression support:** Perl is a winner in everything related to string processing. Its regular expression support is the most versatile in existence and seamlessly integrated into the language. (In fact, it is so good, that Perl has set a new standard for regular expression, which is now emulated in many other programs and languages).
- 3.** It is a beautiful language that combines the best features from many other languages and is very **easy to learn** if you approach it properly. But that's also a matter of taste.

JAVA

Java is a programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. It can also be used to build a small application module or applet for use as part of a Web page. Applets make it possible for a Web page user to interact with the page.

Java differs from other programming languages in a couple of significant ways. The following sections describe the most important differences.

Platform independence

One of the main reasons Java is so popular is its platform independence, which means that Java programs can be run on many different types of computers. A Java program runs on any computer with a Java Runtime Environment, also known as a JRE, installed. A JRE is available for

almost every type of computer — PCs running Windows, Macintosh computers, Unix or Linux computers, huge mainframe computers, and even cell phones.

Object orientation

Java is inherently object-oriented, which means that Java programs are made up of programming elements called objects. Simply put, an object is a programming entity that represents either some real-world object or an abstract concept.

All objects have two basic characteristics:

- Objects have data, also known as state. For example, an object that represents a book has data such as the book's title, author, and publisher.
- Objects also have behavior, which means that they can perform certain tasks. In Java, these tasks are called methods. For example, an object that represents a car might have methods such as start, stop, drive, or crash. Some methods simply allow you to access the object's data. For example, a book object might have a getTitle method that tells you the book's title.

Classes are closely related to objects. A class is the program code you write to create objects. The class describes the data and methods that define the object's state and behavior. Then, when the program executes, classes are used to create objects.

For example, suppose you're writing a payroll program. This program needs objects to represent the company's employees. So, the program includes a class (probably named Employee) that defines the data and methods for each Employee object. Then, when your program runs, it uses this class to create an object for each of your company's employees.

The Java API

The Java language itself is very simple. However, Java comes with a library of classes that provide commonly used utility functions that most Java programs can't do without. This class library, called the Java API, is as much a part of Java as the language itself. In fact, the real challenge of learning how to use Java isn't learning the language; it's learning the API. The Java language has only 50 keywords, but the Java API has several thousand classes — with tens of thousands of methods you can use in your programs.

JAVASCRIPT

JavaScript is scripting language used for client side scripting. JavaScript developed by Netscape in 1995 as a method for validating forms and providing interactive content to web site.

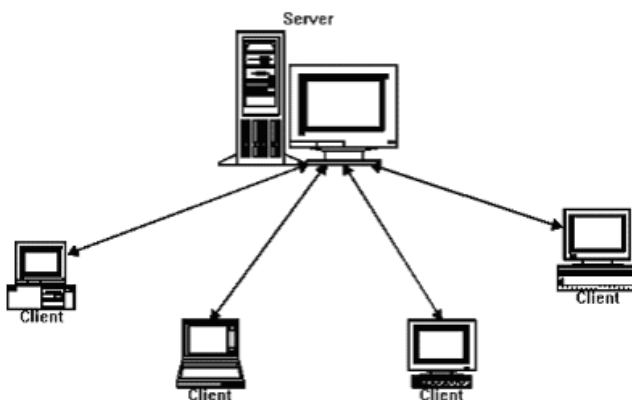
JavaScript is an interpreted client-side scripting language that allows a web designer the ability to insert code into their web page. JavaScript is commonly placed into an HTML, ASP, or a similar file and run directly from the web page to perform tasks such as printing the time and date, creating a calendar, or other tasks that are not possible through plain HTML.

Advantages of JavaScript

- Javascript is executed on the client side
This means that the code is executed on the user's processor instead of the web server thus saving bandwidth and strain on the web server.
- Javascript is a relatively easy language
The Javascript language is relatively easy to learn and comprises of syntax that is close to English. It uses the DOM model that provides plenty of prewritten functionality to the various objects on pages making it a breeze to develop a script to solve a custom purpose.
- Javascript is relatively fast to the end user
As the code is executed on the user's computer, results and processing is completed almost instantly depending on the task (tasks in javascript on web pages are usually simple so as to prevent being a memory hog) as it does not need to be processed in the site's web server and sent back to the user consuming local as well as server bandwidth.
- Extended functionality to web pages
Third party add-ons enable Javascript developers to write snippets of Javascript which can execute on desired web pages to extend its functionality.

Hypertext PreProcessor (PHP)

PHP is an open source programming language primarily implemented on the Internet in websites. Its main purpose is to bring dynamic content to the web pages on which it is implemented.



Unlike traditional programming languages, PHP is not as concerned with optimal performance or operation. Rather, the language is designed to be relatively simple to use by even novice developers. Easy-to-use functions provide a great deal of power, allowing scripts to be easy to make and implement. On the other hand, PHP is

also dynamic enough that it can be implemented even more effectively by professionals and experts who do desire high levels of performance.

PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, IIS, and many others. And this includes any web server that can utilize the FastCGI PHP binary, like lighttpd and nginx. PHP works as either a module, or as a CGI processor.

There are three main areas where PHP scripts are used.

- **Server-side scripting.** This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server.
- **Command line scripting.** You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on *nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.
- **Writing desktop applications.** PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution

Active Server Page (ASP)

Microsoft's Active Server Pages (ASP) is a server-side scripting environment that you can use to create and run dynamic, interactive Web server applications. With ASP, you can combine HTML pages, script commands, and COM components to create interactive Web pages and powerful Web-based applications that are easy to develop and modify.

ASP programming involves scripting in Visual Basic Script, Jscript, Perl, Python, or other languages. Certain modifications are necessary, but the programmer who has written code in these other languages will find ASP programming to be familiar indeed. The two languages that work the best for ASP programming are VBScript and Jscript.

Advantages of ASP

Support for Tools: With ASP you can drag and drop your web controls just like how it is done in Visual Basic controls. All you have to do is just double click to write a server program for the control.

Ease of Use: ASP pages are just HTML pages with ASP programs encapsulated between <% and %> tags. To run an asp file you just have to place it into a directory in your server and run it provide you have proper authorization. All you have to do to make some changes is just edit your asp file and the modification will be executed.

Flexibility: With ASP you can code your scripts in both VBScript and Microsoft Jscript. There are also existing modules to support languages like Python, perl along with solutions to incorporate other languages. So even if you are new to ASP you can use your previous knowledge in other programming languages like VB and JavaScript while migrating to ASP. This makes it easier to learn ASP even with just html knowledge.

Disadvantages of ASP

Higher costs: To process asp pages you need to install IIS on a Windows platform server. You have to buy MS-SQL to connect to your database in ASP. And you need to add some components as there are no inbuilt features like in PHP. All these activities will cost you money.

Lower Speed: ASP is slower than PHP, as it is based on COM architecture that is actually an overhead to the server.

Not Platform Compatible: ASP does not have platform compatibility like PHP. This is because it is mostly dependent on Microsoft products and needs a Windows platform along with an ASP-Apache installation at the server side.

Needs VB knowledge: The syntax of ASP has a lot of similarity with that of Visual Basic. Hence some VB knowledge is needed to program in ASP.

.NET Application

The .NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages.

The .NET Framework consists of the **common language runtime** and the **.NET Framework class library**. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remote services, while also enforcing strict type safety and other forms of code accuracy that promote security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

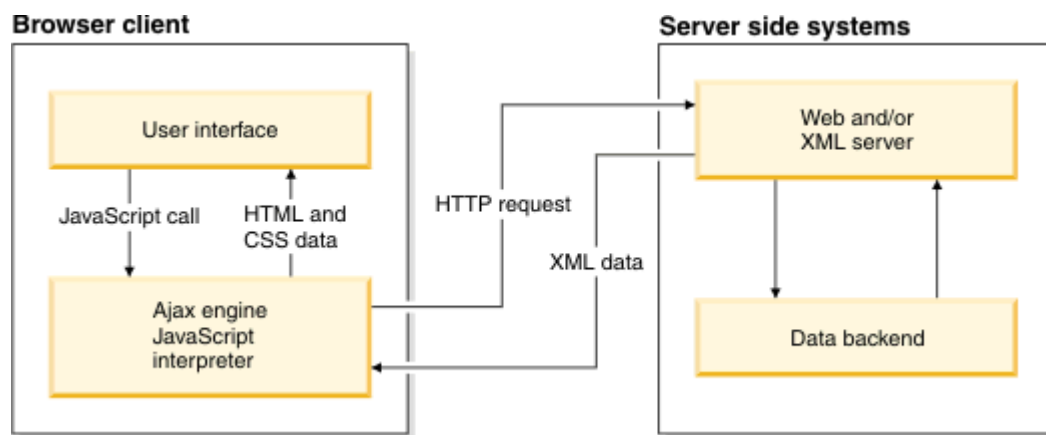
The principal design features of .NET are:

- **Interoperability:** This allows for .NET-developed programs to access functionalities in programs developed outside .NET.
- **Common Runtime Engine:** Also known as the common language runtime, this allows programs developed in .NET to exhibit common behaviors in memory usage, exception handling and security.
- **Language Independence:** Common language infrastructure specifications (CLI) allow for the exchange of data types between two programs developed in different languages.
- **Base Class Library:** A library of code for most common functions--used by programmers to avoid repetitive rewriting of code.
- **Ease of Deployment:** There are tools to ensure the ease of installing programs without interfering with previously installed applications.
- **Security:** Programs developed in .NET are based on a common security model.

Introduction to AJAX (Asynchronous JavaScript and XML)

AJAX is a technique for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

Asynchronous refers to events that are happening on the background independently of the main application flow. These events do not disturb the flow of the application, allowing the continuation of its normal process. A fairly good example of this happening is in your Facebook home page, when all of a sudden, without refreshing your browser window, you notice that there are new status feed updates from your friends.



JavaScript handles most of the basic tasks such as data validation and manipulation, as well as display rendering the Ajax engine handles without a trip to the server. At the same time that it is making display changes for the customer, it is sending data back and forth to the server. But the data transfer is not dependent upon actions of the customer.

XML is used to interchange the data presented in the web page and XMLHttpRequest is the main aspect behind Ajax that distinguishes its functionality from other web development techniques, it allows processing data using the web server asynchronously.

Ajax limitations

While Ajax is a Web application development technique that is designed to make Web pages more responsive and interactive with a user, Ajax has some limitations that should be considered before developing an Ajax-based application. The following limitations are some of the more prominent disadvantages:

- **Browser support** - Not all browsers support JavaScript or XMLHttpRequest object. Even among browsers that do have support for JavaScript and XMLHttpRequest, these objects can be treated differently. Each browser's implementation of Ajax must be considered.

- **Security and user privacy** - Not all concerns have been addressed. Issues surrounding security and user privacy need to be considered when developing an Ajax application.
- **Accessibility** - Because not all browsers have JavaScript or XMLHttpRequest object support, you must ensure that you provide a way to make the Web application accessible to all users.
- **Bookmark and navigation** - Since Ajax is used to asynchronously load bits of content into an existing page, some of the page information may not correspond to a newly loaded page. Browser history and bookmarks may not have the correct behavior since the URL was unchanged despite parts of the page changing.
- **Search engine** - Ajax applications are not searchable; however, it is possible to use Ajax features and elements within an application that is searchable.