Behind the scenes:

- Introduction, benefit and application of ASP;
- Introduction to IIS: Features, properties and application of IIS and MMC, Virtual directory properties;
- ASP requirements: Need for ASP, Scripting capabilities, Recognizing individuals, Database access, State maintenance, ASP extensibility.

HTTP

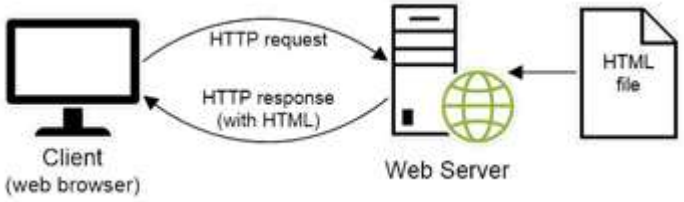
The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers.

Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other. HTTP specification specifies how clients' request data will be constructed and sent to the server, and how the servers respond to these requests.

Basic Features

There are three basic features that make HTTP a simple but powerful protocol:

- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client disconnects from the server and waits for a response. The server processes the request and re-establishes the connection with the client to send a response back.
- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.
- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.



Comparison of Server-side vs Client-side scripting

COMPARISON	SERVER-SIDE SCRIPTING	CLIENT-SIDE SCRIPTING
Basic	Works in the back end which could not be visible at the client end.	Works at the front end and script are visible among the users.
Processing	Requires server interaction.	Does not need interaction with the server.
Languages involved	PHP, ASP.net, Ruby on Rails, ColdFusion, Python, etcetera.	HTML, CSS, JavaScript, etc.
Affect	Could effectively customize the web pages and provide dynamic websites.	Can reduce the load to the server.
Security	Relatively secure.	Insecure

Comparison of Static and Dynamic Websites

We have discussed the basics of both static and dynamic websites. Here is the complete comparison of static and dynamic website features based on each parameters from usability to search engine visibility.

Parameters	Static Webpage	Dynamic Webpage
Definition	• Loads the same content every time the page is reloaded.	• Loads different content each time the page is loaded or refreshed. • Provides interactive features within the page without re-loading.
Browser	• Browser sends the HTTP request and receives the response from the server. • Interprets the markups in the received HTML document and display it as a webpage.	• Client side scripting works in the same manner like a static page and the browser process the script code. • Server side scripting involves processing of the script code at server side before sending it to the client's browser.

Browser Settings	• No special settings required.	• JavaScript is to be enabled in all type of browsers to see dynamic behaviors of a site.
Complexity	• Less complex	• Client side scripting involves medium complexity. • Server side scripting involves high complexity.
Cost	• In most of the cases it costs only a domain name. • Free hosting is available with most of the website builder tools.	• Separate hosting account is required to access server side, hence sites involving server side scripting needs high cost compared to the static or sites involving client side scripting.
Google Analytics	• Installing Analytics code is simple. • Copy the Analytics script code and paste it in the header section of a page.	• Needs to follow the difficult process using Tracking Code wizard for installing Analytics code in a PHP or ASP page.
Interactive Features	• No interactive features are provided to the visitors.	• Simple features like form validations are done with JavaScript. • Complex features like login module are created with PHP.
Language	• Static pages are generally created with Hyper Text Markup Language (HTML). • It is not necessary that all HTML pages are static. Scripts can be embedded in a static HTML document.	• Dynamic pages are created with scripting languages. • Client side scripting language includes JavaScript, ActionScript and Flash. • Server side scripting language includes PHP, ASP, JSP, ASP.NET, ColdFusion, Perl and WebDNA.
Layout and Content	• Both content and layout of a static webpages are fixed.	• Layout and content can be changed independently in a dynamic webpage.
Loading	• Static page loads very fast since no script processing at client and server side is required.	• Dynamic page loads slowly compared to a static page since it involves processing of client or server side scripts.
Multimedia	• Simple video or audio elements can be added to a static site.	• Flash objects can be added to a dynamic site. • Flash objects respond to the user inputs and provide more interactive features.
Page Name	• Static page name mostly ends with .html or .htm.	• Dynamic page name ends with .php or .asp.
Page Speed	• Loads very fast.	• Loads slow due to script processing. • Sometimes script becomes non-responsive and forces the browser to close.
Personalized	• Content can't be personalized for a specific user.	• Content can be personalized for a specific user based on login or any other parameter.
SEO	• Search engines easily index the static pages.	• Since the content of a same page is changing, search engines finds it difficult to index dynamic pages. • Webmasters can redirect dynamic pages to a static one so that the search engines can index it easily.
Server	• Server receives the request and sends the HTML document as it is.	• Server processes the script code if required before sending the document to the client.
Setup	• It is easy to setup a static website using any website builder tools.	• It is difficult to setup a dynamic site since the setting up of content management system is more time consuming.
Some Examples	• Any fixed content site can be a static site.	• Simple form validations using JavaScript. • Complex server side activities like login, session id tracking and payment gateway for credit card processing.
Source Code	• Source code of a static page will show the HTML content along with embedded client side scripts if any. • Right click on any webpage to see the source code.	• Source code of a dynamic page will only show the HTML content and does not show any server side script code. • For example, check the source code of any .php page and you will not find any PHP codes.

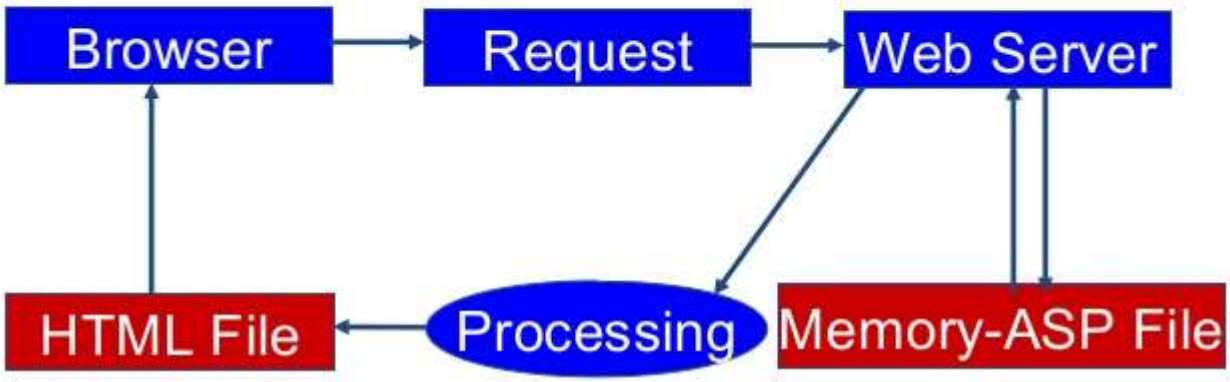
Suitablility	• More suitable for distributing fixed information created and maintained by the site owner.	• Suitable for sites providing more interactive and customized features based on user login or other inputs.
User Friendliness	• Less user friendly due to the fixed content.	• More user friendly by providing customized content.

Introduction, benefit and application of ASP;

- Classic ASP or ASP Classic, is Microsoft's first server-side script engine for dynamically generated web pages.
- ASP 2.0 provides six built-in objects: Application, ASPError, Request, Response, Server, and Session.
- VBScript is the default scripting language used for writing ASP, although other scripting languages can be used.
- ASP is now obsolete and replaced with ASP.NET.

Processing of an ASP Page

When a browser requests an ASP file, IIS passes the request to the ASP engine. The ***ASP engine reads the ASP file***, line by line, and executes the scripts in the file. Finally, the ASP file is returned to the browser as plain HTML.



Benefits of ASP

Language Independence

ASP is a scripting engine enabling you to develop in virtually any language of your choice. The two languages available by default are VBScript and JScript (Microsoft's version of JavaScript); however, modules for Perl, Python and other languages already exist and there are virtually no limits for support for other languages to be implemented. This enables the novice ASP developer to utilize his or her previous programming experience. If you have ever programmed in Visual Basic or VBA (the version of Visual Basic used in the MSOffice suite), you will have no problems starting with VBScript. If you're familiar with JavaScript, then JScript is your choice. Unix gurus will find that Perl can be used readily.

Short Learning Curve

As you have seen, you may use your current expertise in some programming language or technology to jump into ASP in short time. Even if you know only HTML, it will not be difficult for you to learn how to insert ASP commands into your HTML files. Why, you ask? Just keep reading....

Tons of Information

There are currently more than 150 sites listed in Open Directory's ASP category. This is more than for any other server-side development engine or language. Lots of online magazines will deliver new articles on ASP to your mailbox on a weekly or even daily basis. There's a lot of resources out there for ASP, folks.

Huge Community

There are plenty of professional ASP developers ready to answer your questions in numerous ASP newsgroups and forums. For instance, on the day I write this article, there are more than 50 new threads in the microsoft.public.inetserver.asp.general newsgroup alone.

Low Cost of Ownership

This mainly applies to site owner, not directly to the developer. Nowadays it's easier to find an NT administrator and ASP developer than a UNIX guru. You may think that for you as a developer, this isn't a compelling reason to use ASP. However, low cost and abundance of support is just one more reason you can give to your employer/customer as to why they should

do business with you. You can explain that they won't get left high and dry if one day you decide to part company, so while you're with them they will feel safe and satisfied.

Extensibility

There are virtually no limits to what can be done with ASP thanks to unlimited extensibility provided via COM components. This approach, in my opinion, is a key success factor of ASP. For example, there's no way to send email using standard ASP functions but there are lots of components (both free and commercial) enabling you to do this, as well as chosing the methods and features you want implemented. Recent introduction of Java Server Pages - an ASP style scripting language developed to work in conjunction with server-side Java, shows that the approach is recognized as powerful by one of the biggest Microsoft competitors, Sun Microsystems.

Hosting

Hosting companies widely support ASP. A search on HostIndex returned 889 matches for hosting companies supporting ASP with hosting prices starting below $10. This is quite enough to find the host that meets all your needs for a reasonable price and you will be able to switch hosting companies easily if your current doesn't satisfy you.

Tools

Microsoft has two tools supporting ASP: their most popular WYSIWYG editor - FrontPage (though I would not personally recommend it for ASP editing) and Visual InterDev. Other vendors have also implemented ASP support into their products, including the popular HTML coding package HomeSite from Allaire. There's also scheduled support for ASP in upcoming versions of other widely-used code based editors, such as HotDog Pro from Sausage Software.

You Can Get a Good Job

A search for ASP on high tech job search engine dice.com returned more than 10000 matches with average salary around $70K. I think it will be hard for you to chose the best from such a quantity, but very simple to find good one.

Advantages of using Asp.Net Framework

1: Keep you Asp.net applications secured with the built-in Windows authentication and per-application configuration.
2: Asp.Net has reduced the long lines of code required to develop large applications.
3: Asp.Net and Html, together generate dynamic web pages smoothly.
4: Being an ideal server-side scripting technology, Asp.Net code first runs on Windows server before displaying on the web browser.
5: Asp.Net framework is language independent, means you can choose any programming language which best suited to you application.
6: With the built-in configuration information, Asp.Net is easy to deploy.
7: The windows web server thoroughly monitors the web pages, multiple components, and applications running over it.
8: The Dot Net Framework quickly gives an alert for memory leaks, unbounded loops, and other wrong behaviors, immediately killing them and restart them over again.
9: Asp.Net features like early binding, JIT compilation, caching services and native optimization supports gives your application the high level of performance.
10: All the Asp.Net applications are highly monitored and managed to help application available to handle requests.
11: The best part of Dot Net Framework is it has its own built-in caching features.
12: The content and the program logic are separated in the .Net Framework, thus reducing the program inconveniences.

Introduction to IIS: Features, properties and application of IIS and MMC, Virtual directory properties;

Internet Information Services (**IIS**, formerly **Internet Information Server**) is an extensible web server created by Microsoft for use with the Windows NT family. IIS supports HTTP, HTTP/2, HTTPS, FTP, FTPS, SMTP and NNTP.
IIS also known as Windows web server is available on most versions of Microsoft Windows operating systems and takes second place in overall usage behind Apache HTTP Server on the internet. It will host websites, web applications and services needed by users or developers
IIS 6.0 and higher support the following authentication mechanisms:

- **Anonymous authentication**: clients can access the server without requiring a user name and password
- **Basic access authentication:** an HTTP user agent (e.g. a web browser) to provide a user name and password when making a request.
- Digest access authentication: This can be used to confirm the identity of a user before sending sensitive information, such as online banking transaction history. It applies a hash function to the username and password before sending them over the network
- Integrated Windows Authentication: enables users to log in with their Windows credentials, using Kerberos or NTLM. The client sends credentials in the Authorization header. Windows authentication is best suited for an intranet environment
- UNC authentication: UNC authentication allows you to configure IIS to use a specified user account when accessing resources on a remote share. When creating a virtual directory (or web**application**) that points to a UNC (Universal Naming Convention) share,**credentials** can be provided for accessing that share.

- .NET Passport Authentication (Removed in Windows Server 2008 and IIS 7.0): **Passport authentication** is a centralized **authentication** service provided by Microsoft that offers a single logon and core profile services for member sites.**Passport** benefits users because they do not need to log on to new limited-access resources or sites.
- Certificate authentication: Has the Digital Certificate been issued/signed by a Trusted CA? Is the Certificate Expired – checks both the start and end dates. Has the Certificate been revoked? (Could be OCSP or CRL check). Has the client provided proof of possession?

IIS 7.0 has a modular architecture. Modules, also called extensions, can be added or removed individually so that only modules required for specific functionality have to be installed. IIS 7 includes native modules as part of the full installation. These modules are individual features that the server uses to process requests and include the following:

- **Security modules**: Used to perform many tasks related to security in the request-processing pipeline, such as specifying authentication schemes, performing URL authorization, and filtering requests.
- **Content modules:** Used to perform tasks related to content in the request-processing pipeline, such as processing requests for static files, returning a default page when a client does not specify a resource in a request, and listing the contents of a directory.
- **Compression modules**: Used to perform tasks related to compression in the request-processing pipeline, such as compressing responses, applying Gzip compression transfer coding to responses, and performing pre-compression of static content.
- **Caching modules**: Used to perform tasks related to caching in the request-processing pipeline, such as storing processed information in memory on the server and using cached content in subsequent requests for the same resource.
- **Logging and Diagnostics modules**: Used to perform tasks related to logging and diagnostics in the request-processing pipeline, such as passing information and processing status to HTTP. sys for logging, reporting events, and tracking requests currently executing in worker processes.

IIS 7.5 includes the following additional or enhanced security features:

- Client certificate mapping
- IP security
- Request filtering
- URL authorization

Authentication changed slightly between IIS 6.0 and IIS 7, most notably in that the anonymous user which was named "IUSR_{machinename}" is a built-in account in Vista and future operating systems and named "IUSR". Notably, in IIS 7, each authentication mechanism is isolated into its own module and can be installed or uninstalled.

IIS 8.0 offers new features targeted at performance and easier administration. The new features are:

- **Application Initialization**: a feature that allows an administrator to configure certain applications to start automatically with server startup. This reduces the wait time experienced by users who access the site for the first time after a server reboot.
- **Splash page during application initialization**: the administrator can configure a splash page to be displayed to the site visitor during an application initialization.
- **ASP.NET 4.5 support**: With IIS 8.0, ASP.NET 4.5 is included by default, and IIS also offers several configuration options for running it side-by-side with ASP.NET 3.5.
- **Centralized SSL certificate support**: a feature that makes managing certificates easier by allowing the administrator to store and access the certificates on a file share.
- **Multicore scaling on NUMA hardware**: IIS 8.0 provides several configuration options that optimize performance on systems that run NUMA, such as running several worker processes under one application pool, using soft or hard affinity and more.
- **WebSocket Protocol Support**
- **Server Name Indication (SNI):** SNI is an extension to Transport Layer Security, which allows binding of multiple websites with different hostnames to one IP address (similar to how Host Headers are used for non-SSL sites).
- **Dynamic IP Address Restrictions**: a feature that enables an administrator to dynamically block IPs or IP ranges that hit the server with a large number of requests
- **CPU Throttling**: a set of controls that allow the server administrator to control CPU usage by each application pool in order to optimize performance in a multi-tenant environment

IIS 8.5 has several improvements related to performance in large-scale scenarios, such as those used by commercial hosting providers and Microsoft's own cloud offerings. It also has several added features related to logging and troubleshooting. The new features are:

- **Idle worker-Process page-out**: a function to suspend idle site to reduce the memory footprint of idle sites
- **Dynamic Site Activation**: a feature that registers listening queues only to sites that have received requests
- Enhanced Logging: a feature to allow collection of Server variables, request headers and response headers in the IIS logs
- **ETW logging**: an ETW provider which allows collecting real-time logs using various Event-tracing tool

- **Automatic Certificate Rebind**: a feature that detects when a site certificate has been renewed, and automatically rebinds the site to it

IIS Express

IIS Express, a lightweight (4.5–6.6 MB) version of IIS, is available as a standalone freeware server and may be installed on Windows XP with Service Pack 3 and subsequent versions of Microsoft Windows.

Extensions

IIS releases new feature modules between major version releases to add new functionality. The following extensions are available for IIS 7.5:

- **FTP Publishing Service**: Lets Web content creators publish content securely to IIS 7 Web servers with SSL-based authentication and data transfer.
- **Administration Pack**: Adds administration UI support for management features in IIS 7, including ASP.NET authorization, custom errors, FastCGI configuration, and request filtering.
- **Application Request Routing:** Provides a proxy-based routing module that forwards HTTP requests to content servers based on HTTP headers, server variables, and load balance algorithms.
- **Database Manager**: Allows easy management of local and remote databases from within IIS Manager.
- **Media Services**: Integrates a media delivery platform with IIS to manage and administer delivery of rich media and other Web content.
- **URL Rewrite Module**: Provides a rule-based rewriting mechanism for changing request URLs before they are processed by the Web server.
- **WebDAV**: Lets Web authors publish content securely to IIS 7 Web servers, and lets Web administrators and hosters manage WebDAV settings using IIS 7 management and configuration tools.
- **Web Deployment Tool**: Synchronizes IIS 6.0 and IIS 7 servers, migrates an IIS 6.0 server to IIS 7, and deploys Web applications to an IIS 7 server.

MMC

- Provides system administrators and advanced users an interface for configuring and monitoring the system.

Virtual Directory

- A virtual directory is a path or alias within a website that refers users to another directory where the actual data is hosted.
- Virtual directories are used if website admins need to put files in directories other than the home directory and publish from them

ASP requirements: Need for ASP, Scripting capabilities, Recognizing individuals, Database access, State maintenance, ASP extensibility.