The World Wide Web

One of the most significant additions to the resources available on the Internet (at least in terms of mass-use of this network) was the collection of sites that comprise the “World Wide Web”.

The World Wide Web (or the Web) is a conceptual organization of many distinct hosts’ resources under a single title. The underlying structure of the web was "released" in 1991 when Tim Berners-Lee posted a short summary on USENET:

http://groups.google.com/group/alt.hypertext/msg/395f282a67a1916c

Unlike other resources, this Internet entity gained widespread popular attention, and quickly became a part of our culture. Once the Internet became more integrated into our lives and homes, other resources that already existed (online library catalogs, electronic mail, text messaging) gained popularity as well.

There are several factors that facilitated this, including:

- the way in which web pages can be designed and how you connect web pages to one another
- the applications designed to access it
- the fact that it is a multi-media capable resource
- the way in which you locate and then access web pages
A web page is a typically a document designed according to the language known as HTML (or HyperText Markup Language). This is a document layout language that allows you to place and format text as well as pictures, and to make certain parts of the document links to other documents.

The use of HTML makes for a more customizable environment. In earlier systems (such as one called Gopher), you were limited to text menus. Also, the only information on a page was typically titles for resources, or other groups of resources, or file names. That was no longer the case when working with the Web. Documents created for the Web had formatted text and pictures (and eventually much more), and the page creator could specify which of these (text or images) could be selected by the user to move to a different page or resource. This type of document is referred to as a hypertext document.
Hypertext

The concept of a hypertext document is not unique to the World Wide Web, though this is the only example of its use that we will look at in this class in great detail. The basic idea behind hypertext is to display a certain set of information, and to allow the user to choose to see more information related to one part of the original document if they want to.

One of the first (if not the first) project that explored the ideas behind hypertext was the HyperTIES project right here at the University of Maryland back in 1982.

http://www.cs.umd.edu/hcil/hyperties/

Another early use was Hypercard Stacks, released in the 80s for the Apple Macintosh. In this system, you had individual cards, each of which could have links to other individual cards.

In fact, when viewed in Adobe's Acrobat viewer, this very document is a hypertext document – it contains text links to other documents.

However, the inspiration for hypertext and the web predates both of these innovative applications by almost 40 years. In 1945, Vaneveer Bush's article "As We May Think" was published in The Atlantic Monthly (an expanded version of an earlier paper of his) that contained descriptions of things similar to hypertext documents, the web, digital cameras, and much more.
For example, let’s look at the following information:

**Name:** Evan Golub  
**Department:** Computer Science  
**Position:** Lecturer

**Courses:**
- CMSC 102
- CMSC 103
- CMSC 150
- CMSC 214
- CMSC 434

**Interests:**
- Babylon 5
- Skiing

The above is a basic summary of information. If it was hypertext, a user could perhaps select my name or the word “skiing” to get some more detailed information. Let’s look at the same information on a Web page.....

Web Browsers

It is important to note that the Web is simply one of many Internet resources. We will soon discuss the Internet Protocols and TCP ports and see how there is nothing special about the web from the point of view of the Internet itself. We use a **client** (such as Mosaic, Internet Explorer, Firefox, Opera, Lynx) to connect to web **server(s)** using the Internet as our communications medium. Lynx is a text-based browser, while Firefox, Mosaic, Netscape, Opera and Internet Explorer are graphical interface based browsers. However, they all share one thing in common; they are clients that request information from servers, and then interpret that information to accomplish tasks such as constructing the web pages they display.
Multi-Media On The Web

Unlike other systems (such as file transfers which we will see soon) where a GUI client didn’t add anything to the information immediately available to the user, pages on the Web can have a variety of visual and audio components. It should be noted that a user who accesses the web via a text browsers such as Lynx or those on some cell phones will only see a very narrow subset of the information on a page.

World Wide Web pages can have:

- formatted text (different fonts, bolded, underlined, flashing, ...)
- pictures (GIFs, JPEGs, ...)
- embedded movies (MPEG, AVI, QuickTime, FLV, RealVideo, ...)
- interactive applications (via DHTML, SWFlash, Java, Silverlight…)
- embedded sound (WAV, AU, MP3, RealAudio, ...)
- forms (fill-ins, check boxes, pull down menus, buttons, ...)
- more and more every year it seems...

These are being used in some situations (such as Google Documents or Photoshop Online) to create desktop-like applications that can run within your web browser, and that can store your files remotely.

Additionally, web pages can be used to direct you to other Internet-based sources of media that use different transmission protocols.
Using a Web Browser to Retrieve and View Information

The World Wide Web is essentially a collection of Hyper Text Transfer Protocol (http) servers running on machines connected to the Internet, and the files (html and others) available via these servers. You can actually obtain files from these servers using a direct request to the appropriate port on the appropriate machine using a `telnet` client (we will discuss later this semester). There are programs that you can use that will request a specific file from a web server, and then store it directly to your hard drive.

However, to get the full functionality of the Web, you really need to use a web browser as your interface to these sites. While there are some browsers that are text only, using one technically only limits how we can view things via the browser, but not what we can retrieve.

Some servers are `https`, which means that the information being transmitted is being encrypted to add a level of protection against certain risks.
Getting to a Web Page

The location of a page in the World Wide Web is specified using a URL (Uniform Resource Locator). The initial idea was to create a "networked extension" to the concept of a file's name and location.

An example of a URL is:


This URL is divided into three parts:

- resource type identifier [http:]
- host name [/www.cs.umd.edu]
- directory (and file) name [/users/egolub/index.html]
Resource Type Identifier

You can actually access many other parts of the Internet via a typical Web Browser by giving a URL for a different resource. The first step in identifying a request as being for a non-Web resource is through the type (or protocol) identifier in the URL. The following are among the ones available:

- **telnet** - Connect via telnet (default of 23)
- **gopher** - Contact the Gopher server on the given site (*little use anymore*)
- **ftp** - Request a file via the FTP file transfer standard
- **file** - Request a file from your local machine
- **wais** - Connect to the given WAIS server (*little use anymore*)
- **https** - A **secure** connection to a web server (*high use*)
- **news** - Read a USENET newsgroup (*far less use nowadays*)
- **mailto** - Launch an e-mail client to compose a message (*common use*)
- **aim** - Open a window to the specified buddy (*little use*)
- **rtmp** - Streaming protocol (Real Time Messaging Protocol) often used for FLV
- **imap** - A possible way to access your e-mail inbox (*little use*)

Many of these are “hidden” behind the scenes to most users. Depending on which you start a URL with, the client application you are using will act in the appropriate way. It might connect to the given host via the appropriate port and send the required commands once connected to request the resource. It might launch a helper application with some initial settings information, which makes it feel like certain things are “on the web” when they really aren’t.
Identifying a File’s Type

When a file is requested, the browser will typically use the file name’s suffix to *guess* what type of file it is going to be. Then based on the anticipated type, it will decide how to “display” it.

<table>
<thead>
<tr>
<th>File Name Suffix</th>
<th>File Type Usually</th>
</tr>
</thead>
<tbody>
<tr>
<td>.txt</td>
<td>text file</td>
</tr>
<tr>
<td>.html</td>
<td>hypertext document</td>
</tr>
<tr>
<td>.shtml</td>
<td>html document with server-side processing</td>
</tr>
<tr>
<td>.gif</td>
<td>GIF image</td>
</tr>
<tr>
<td>.jpg</td>
<td>JPEG image</td>
</tr>
<tr>
<td>.pdf</td>
<td>PDF document</td>
</tr>
<tr>
<td>.swf</td>
<td>Shockwave Flash applet</td>
</tr>
<tr>
<td>.ps</td>
<td>PostScript file</td>
</tr>
<tr>
<td>.au</td>
<td>AU format sound file</td>
</tr>
<tr>
<td>.wav</td>
<td>WAVE format sound file</td>
</tr>
<tr>
<td>.ram</td>
<td>RealAudio audio/video format (typically streaming)</td>
</tr>
<tr>
<td>.mp3</td>
<td>MP3 compressed sound file</td>
</tr>
<tr>
<td>.mpg</td>
<td>MPEG movie (more than one mpg level possible)</td>
</tr>
<tr>
<td>.mp4</td>
<td>Movie (common use on iPlatform)</td>
</tr>
<tr>
<td>.avi</td>
<td>AVI movie (many codecs for avi format family)</td>
</tr>
<tr>
<td>.zip</td>
<td>ZIP compressed file</td>
</tr>
<tr>
<td>.flv</td>
<td>Flash Video file (commonly used by YouTube)</td>
</tr>
<tr>
<td>.m4p</td>
<td>iTunes (DRM-protected) music file</td>
</tr>
</tbody>
</table>
Helper (External) Applications

A browser might allow the user to specify what external application is appropriate for “displaying” a given file type. It is pre-configured to have certain files displayed within the browser (such as GIFs), but also allows you to let it know how to handle other types of files.

Consider PostScript files; this is a file that is typically created to be sent to a specific type of printer. The PostScript language is designed to tell printers how to create an image on the printer. However, it has become a common practice to place PostScript versions of papers on the Web, in order to allow others to get a copy of the paper (with all diagrams, etc...) from the Web. A user will typically want a file with a .ps suffix to be saved to their local disk, and then they will send it to a PostScript printer. However, it is also possible to use a PostScript viewer to see (rather than print) the file.

It has become more common for embedded helper applications (plug-ins) to be made available, to make the browsing experience more seamless.
What is the CGI-BIN directory?

In addition to the files (text, html documents, images, etc...) which are accessible via the Web, most Web servers will also allow you to activate a program on the remote machine running the Web server. Many of these are Common Gateway Interface (CGI) scripts. A CGI script is a program written in a simple language that allows a Web user to execute a program on a machine running the Web server. Many Web sites have this feature turned off because of security risks attached to allowing a non-system administrator (the author of a Web page/CGI script) to give system access to an unknown user (the Web surfer) at will.

Typically, in order for a program to be run, the URL must point to a file in a specific cgi-enabled directory (or a subdirectory of that). An administrator can then allow either no users or only certain users the ability to run CGI scripts by controlling access to this special directory path.

Example: http://www.m-w.com/cgi-bin/dictionary?this

Some sites place this in a directory with the name "cgi-bin". However, it is very common to place them in other directories to make the URLs more logical over a large site. For example, the above can also appear as:

http://www.merriam-webster.com/dictionary/this
Plug-ins, JavaScript, Java, ActiveX...

A popular trend for pages on the Web is to have the browser request a program, and then to run that program on the user’s machine. Various companies are making products for use with browsers known as plug-ins that allow an application to run within the browser, using non-HTML “display” programs. Some commonly seen plug-ins are for running Java and Flash and Silverlight programs (often referred to as applets) within a web page, and also Adobe Reader for viewing PDF files directly within the browser. There are, however, many other plug-ins; some written by "regular people" with programming experience and little to no way to assign trust values to them.

On top of this, different browsers support different extensions to the HTML language as well as browser-specific instruction sets. Some are common in many browsers, such as a core part of JavaScript (which we will discuss in more detail towards the end of the semester). Others might be specific to one browser (or a very limited subset of browsers), such as Internet Explorer allowing page designers to use a language known as ActiveX.

There can be interactions between these - for example, when using the Internet Explorer v6 browser, Active X is used to launch the PDF viewer plug-in. This means the disabling one functionality of a browser could break seemingly-unrelated abilities.
Compatibility

Not all plug-ins "exist" for all browsers or even all versions of some browsers. When you upgrade your browser's version, older plug-ins might no longer work (this has been an issue with Firefox, for example). However, if you don't update your browser then other newer plug-ins might not work.

Many pages use **Flash** or **Java** but on the iOS platform (for example) the browser does not support either. Many pages use forms that might make use of newer features that a browser from a year or two ago didn't support and few companies seem to test their web pages on a wide enough variety of browsers as they probably should.

In theory, you should just need one browser installed. In reality, you might want to have (for example) a version of Internet Explorer, Firefox, and Safari all handy in case a site that you need to visit for a time-critical use doesn't support some browsers.
Some dangers of viewing web pages

Having code run on the client machine in this type of way removes risks associated with CGI programs from the machine running the server. However, it does add risks to the machine running the client (or browser). When you go to a page, if that page is set up to automatically execute a series of commands on your behalf or your computer, then it is conceivable that a malicious user will have put “bad things” onto their Web page.

As an example, some maliciously designed pages have commands that will alter the DNS lookup features on your machine, which then blocks your ability to access sites by hostname. Some popular anti-virus software now watches for these types of actions, so if you don't have anti-virus software installed, the campus suggests you visit the site listed below to get a free copy of something for your operating system.

   http://www.helpdesk.umd.edu/virus

You also might want to consider scanning your machine regularly using a free tool such as AdAware (http://www.lavasoftusa.com/software/adaware/) or Spybot (http://www.safer-networking.org/en/index.html) or MalwareBytes (http://www.malwarebytes.org/).

Computer viruses are nothing new (the first ones go back to at least the 1980s http://www.cknow.com/articles/14/1/Robert-Slade:-Chapter-7---(c)-Brain) but they are widespread and pose a variety of risks. They can also be used as a weapon in a variety of cyber-attacks against groups, companies, or individuals.
Later in the semester, we will explore certain topics in more depth, but some are worth mentioning early:

- Just because it's on the web, doesn't make it true. Just because it's on the web, doesn't make it false.
  - Actor/Comedian "Sinbad" is alive and well as of my writing these notes, though his Wikipedia entry once said otherwise: [http://www.msnbc.msn.com/id/17638311](http://www.msnbc.msn.com/id/17638311)
  - There actually was a "near miss" when an asteroid passed a mere 50,000 miles from Earth: [http://www.usnews.com/science/articles/2010/09/13/asteroids-miss-with-astronomers.html](http://www.usnews.com/science/articles/2010/09/13/asteroids-miss-with-astronomers.html)
  - There are cats the “glow in the dark” (actually they glow under certain lights).

- Just because it says that it's an anonymous comment, doesn't mean it can't be tracked back to you. Very often, your **IP address** will be recorded, and this can be used to trace the message's origins. Also, “fake name” accounts still leave a trail and data mining could link real people to their alias accounts.

- The rules and policies of web sites can change without warning, especially as those running the site look for new ways to make it interesting or marketable. Many sites have started to make APIs available to allow others to write applications that interact with the site. This could create privacy "holes". [http://www.ddj.com/security/206103547](http://www.ddj.com/security/206103547)