

G. PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY

Accredited by NAAC with 'A' Grade of UGC, Approved by AICTE, New Delhi

Permanently Affiliated to JNTUA, Ananthapuramu

(Recognized by UGC under 2(f) and 12(B) & ISO 9001:2008 Certified Institution)

Nandikotkur Road, Venkayapalli, Kurnool – 518452

Department of Computer Science and Engineering

Lecture Notes

On

WEB AND INTERNET TECHNOLOGIES

For III YEAR – II Sem B.Tech CSE (R15 Regulations)

Internet

- The Internet is a global network of networks that enables computers of all kinds to directly and transparently communicate and share services throughout the world.
- In 1969 the precursor of Internet is born: ARPAnet.
- ARPA = Advanced Research Projects Agency sponsored by the American Department of Defense (DOD).
- Designed to connect military research centers.
- Distributed computer system able to survive a nuclear attack.
- Problem: ARPAnet could connect only networks of the same type.
- In 1970, ARPA starts developing the Transmission Control Protocol / Internet Protocol (TCP/IP), a technology for connecting networks of different types (produced by different companies).
- Other networks appear, such as CSNET and BITNET.
- The Internet = a network of networks.
- People around the world share ideas, information, comments, and stories.

Popular services:

- Email (electronic mail) is the most popular service.
- You can send and receive mail (electronic text), faster and cheaper than surface mail.
 - Example email addresses: [III](mailto:III@CSErocks@gpcet.ac.in)
CSErocks@gpcet.ac.in
- Web browsing to find information.
 - Example: www.gpcet.ac.in

Features of the Internet

- **Mailinglists**: you can subscribe to lists on various subjects to receive emails, you can post (send) messages, you can unsubscribe. Some lists are moderated.
- Newsgroups are collections of messages on various subjects
- FTP (File Transfer Protocol). You can copy files from one computer to another over the Internet.
- **Telnet or remote login**. Permits your computer to log onto another computer and use it as if you were there.

- You need to provide your username and password, for security reasons.
- **Chatrooms.** You can exchange messages with other
- people, anonymously (using a nickname).
- Internet services for companies: e-commerce, etc.

World Wide Web (WWW)

- The World Wide Web allows computer users to locate and view multimedia-based documents (i.e., documents with text, graphics, animations, audios or videos) on almost any subject.
- Even though the Internet was developed more than three decades ago, the introduction of the World Wide Web is a relatively recent event. In 1990, Tim Berners-Lee of CERN (the European
- Laboratory for Particle Physics) developed the World Wide Web and several communication protocols that form the backbone of the Web.
- The Internet and the World Wide Web surely will be listed among the most important and profound creations of humankind.
- In the past, most computer applications executed on “stand-alone” computers (i.e., computers that were not connected to one another).
- The W3C is also a standardization organization.
- Web technologies standardized by the W3C are called Recommendations. W3C Recommendations include the Extensible Hyper-Text Markup Language (XHTML), Cascading Style Sheets (CSS), Hypertext Markup Language (HTML; now considered a “legacy” technology) and the Extensible Markup Language (XML).
- The W3C homepage (www.w3.org) provides extensive resources on Internet and Web technologies.

Web Contents

- Web content is the textual, visual or aural content that is encountered as part of the user experience on websites.
- It may include, among other things: text, images, sounds, videos and animations.

HTML web content

- Even though we may embed various protocols within web pages, the “web page” composed of “html” (or some variation) content is still the dominant way whereby we share content. And while there are many web pages with localized proprietary structure (most usually, business websites), many millions of websites abound that are structured according to a common core idea.

- A web search engine is designed to search for information on the World Wide Web.
- The search results are generally presented in a list of results and are often called hits
- The information may consist of web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories.
- Unlike Web directories, which are maintained by human editors, search engines operate algorithmically or are a mixture of algorithmic and human input.

Types of Website Content

- Static Web Site
- Dynamic Web Site

Static Web Site

- A static web page (sometimes called a flat page) is a web page that is delivered to the user exactly as stored.
- Dynamic web pages which are generated by a web application
- Consequently a static web page displays the same information for all users, from all contexts, subject to modern capabilities of a web server to negotiate content-type or language of the document where such versions are available and the server is configured to do so.
- Static web pages are often HTML documents stored as files in the file system and made available by the web server over HTTP.
- However, loose interpretations of the term could include web pages stored in a database, and could even include pages formatted using a template and served through an application server, as long as the page served is unchanging and presented essentially as stored.
- Static Web pages are very simple in layout and informative in context. Creation of static website content requires great level of technical expertise and if a site owner is intended to create static
- web pages, they must be very clear with their ideas of creating such pages since they need to hire a web designer.

Advantages

- No programming skills are required to create a static page.
- Inherently publicly cacheable (i.e. a cached copy can be shown to anyone).
- No particular hosting requirements are necessary.
- Can be viewed directly by a web browser without needing a web server or application server, for example directly from a CDROM or USB Drive.

Disadvantages

- Any personalization or interactivity has to run client-side (ie. In the browser), which is restricting.
- Maintaining large numbers of static pages as files can be impractical without automated tools

Application areas of Static Website:

- Need of Static web pages arise in the following cases.
- Changes to web content is infrequent
- List of products / services offered is limited Simple e-mail based ordering system should suffice
- No advanced online ordering facility is required
- Features like order tracking, verifying availability of stock,online credit card transactions, are not needed
- Web site not required to be connected to back-end system.

Dynamic Web Sites

- A dynamic web page is a kind of web page that has been prepared with fresh information (content and/or layout), for each individual viewing.
- It is not static because it changes with the time (ex. a news content), the user (ex. preferences in a login session), the user interaction (ex. web page game), the context (parametric customization), or any combination of the foregoing.

Two types of dynamic web sites

- Client-side scripting and content creation
- Server-side scripting and content creation

Client-side scripting and content creation

- Client-side scripting and content creation Using client-side scripting to change interface behaviors within a specific web page, in response to mouse or keyboard actions or at specified timing events. In this case the dynamic behavior occurs within the presentation.
- Such web pages use presentation technology called rich interfaced pages. Client-side scripting languages like JavaScript or ActionScript, used for Dynamic HTML (DHTML) and Flash technologies respectively, are frequently used to orchestrate media types (sound, animations, changing text, etc.) of the presentation.

- The scripting also allows use of remote scripting, a technique by which the DHTML page requests additional information from a server, using a hidden Frame, XMLHttpRequests, or a Web service.
- The Client-side content is generated on the user's computer.
- The web browser retrieves a page from the server, then processes the code embedded in the page (often written in JavaScript) and displays the retrieved page's content to the user.
- The innerHTML property (or write command) can illustrate the client-side dynamic page generation: two distinct pages, A and B, can be regenerated as document.
- innerHTML = A and document.innerHTML = B; or "on load dynamic" by document.write(A) and document.write(B)

Server-side scripting and content creation

- Using server-side scripting to change the supplied page source between pages, adjusting the sequence or reload of the web pages or web content supplied to the browser.
- Server responses may be determined by such conditions as data in a posted HTML form, parameters in the URL, the type of browser being used, the passage of time, or a database or server state.
- Such web pages are often created with the help of serverside languages such as PHP, Perl, ASP, ASP.NET, JSP, ColdFusion and other languages.
- These server-side languages typically use the Common Gateway Interface (CGI) to produce dynamic web pages. These kinds of pages can also use, on the client-side, the first kind (DHTML, etc.).

Application areas of Dynamic Website

Dynamic web page is required when following necessities arise:

- Need to change main pages more frequently to encourage clients to return to site.
- Long list of products / services offered that are also subject to upgradation
- Introducing sales promotion schemes from time to time
- Need for more sophisticated ordering system with a wide variety of functions
- Tracking and offering personalized services to clients.
- Facility to connect Web site to the existing back-end system

The fundamental difference between a static Website and a dynamic Website is a static website is no more than an information sheet spelling out the products and services while a dynamic website has wider functions like engaging and gradually leading the client to online ordering. But both static web site design and dynamic websites design can be designed for search engine optimization. If the

purpose is only to furnish information, then a static website should suffice. Dynamic website is absolutely necessary for e-commerce and online ordering.

Web Technologies

- HTML, which stands for Hypertext Markup Language, is the predominant markup language for web pages. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content.
- It allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behavior of HTML web pages.
- HTML can also be used to include Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both HTML and CSS standards, encourages the use of CSS over explicit presentational markup.

Introduction to Web Servers

- A web Server is a program that automatically downloads pages from the Web
- An application server works with a Web server to handle requests for dynamic content, such as servlets, from Web applications. A Web server uses a Web server plug-ins to establish and maintain persistent HTTP and HTTPS connections with an application server.
- The web server software offers access to documents stored on the server.
- Clients can browse the documents in a web browser.
- The documents can be for example static Hypertext Markup Language (HTML) files, image files or various script files, such as Common Gateway Interface (CGI), Javascript or Perl files. The communication between clients and server is based on the Hypertext Transfer Protocol (HTTP)

We have the Following Web Servers

- **1. Apache** **2. WAMP** **3. XAMPP**

Apache

- Introduced in 1995 and based on the popular NCSA httpd 1.3, Apache is now the most used web server in the world.
- One of the reasons to its popularity is that it is free to use. Also, since the source code is free, it is possible to modify the web server
- Being threaded (threaded or process-driven depending on the operating system, on Unix, Apache uses processes, while threads are used in Win32 environments) means that Apache maintains a pool of software threads ready to serve incoming requests.
- When a request enters the web server, it is assigned one of the free threads, that serves it

throughout the requests' lifetime.

- Apache puts a limit on the number of threads that are allowed to run simultaneously.
- If that number has been reached, requests are rejected.
- Over 56% of Internet Web servers run Apache or an Apache derivative
- Apache is the most commonly used Web Server on Linux systems. Web Servers are used to serve Web Pages requested by client computers. Clients typically request and view Web Pages using Web Browser applications such as **Firefox, Opera, or Mozilla**

What's "Open-Source"?

- Open-Source refers to software that is not only free, but includes the source as well
- Users are free to make whatever modifications they like to make the software work better for them
- Users are generally encouraged to submit improvements for inclusion in the master distribution
- Use `<http://www.apache.org/dist/>`

Modules in Apache

- What makes Apache so attractive is also its architecture.
- The software is arranged in a kernel part and additional packages called modules.
- The kernel is responsible for opening up sockets for incoming TCP connections, handling static files and sending back the result.
- Whenever something else than a static file is to be handled, one of the designated modules takes over
- Modules are convenient when new functionality should be added to a web server, because nothing has to be changed in the kernel
- In Apache, every request goes through a life-cycle, that consists of a number of phases, as shown in Figure below
- To install the Apache2 module for MySQL authentication, you can run the following command from a terminal prompt:

```
sudo apt-get install libapache2-mod-auth-mysql
```

- Apache is a very complex web server, mainly because of the vast number of features provided. Fortunately, most of this functionality stays in clearly separated and independent program modules, which facilitates program understanding and maintenance.

<http://www.apache.org/httpd>

- Apache is a public domain Web server developed by a loosely knit group of programmers. Public domain refers to any program that is not copyrighted.
- Public-domain software is free and can be used without restrictions.

- The first version of Apache, based on the NCSA httpd Web server, was developed in 1995.
- Because it was developed from existing NCSA code plus various patches, it was called a patchy server - hence the name Apache Server.
- The home page of the Apache HTTP server project, also known as the Apache.
- Here you can find official information about the Apache web server. Any new Apache release is announced on this site.
- Whenever a security problem occurs, you can find the details (and fixes) there.
- The site also provides information about the people behind Apache.
- It is under direct control of the Apache Group core team and is located in San Francisco, California.
- As a result of its sophisticated features, excellent performance, and low price - free, Apache has become the world's most popular Web server.
- Source code for the apache is freely available, anyone can adapt the server for specific needs, and there is a large public library of Apache add-ons. Add ons' refers to a product designed to complement another product.
- The original version of Apache was written for UNIX, but there are now versions that run under OS/2, Windows and other platforms.
- Apache has been shown to be substantially faster, more stable, and more feature-full than many other web servers
- The Apache Group maintains rigorous standards before releasing new versions of their server, and the server runs without a hitch on over one half of all WWW servers available on the Internet
- The Apache Group and the free Apache HTTP server stand as an appealing throwback to the early, less-commercial
- Free and open standards supported and maintained by the people who use them are still a good thing for the Web
- The Apache httpd server is a powerful, flexible, HTTP/1.1 compliant webserver.
- It implements the latest protocols, including HTTP/1.1 (RFC2616) and is highly configurable and extensible with third-party modules.
- Apache can handle run-time execution of external applications (scripts)
- Server-side includes allow the server to modify pages at run-time before sending them
- Java servlet modules are available, but not yet part of the base package (still under development)
- SSL-enabled versions of Apache are available, both from commercial vendors and under

open-source licence

- It can be customized by writing 'modules' using the Apache module API provides full source code and comes with an unrestrictive license runs on Windows NT/9x, Netware 5.x and above, OS/2, and most versions of Unix, as well as several other operating systems is actively being developed encourages user feedback through new ideas, bug reports and patches implements many frequently requested features, including.

DBM Databases for Authentication

- It allows you to easily set up password-protected pages with enormous numbers of authorized users, without bogging down the server.

Customized Responses to Errors and Problems

Allows you to set up files, or even CGI scripts, which are returned by the server in response to errors and problems, e.g. setup a script to intercept 500 Server Errors and perform on-the-fly diagnostics for both users and yourself.

Support For CGI Scripting

Allows you to script web applications in PHP, Perl, Python and many more languages.

Multiple Directory Index Directives

Allows you to say Directory Index index.html index.cgi, which instructs the server to either send back index.html or run index.cgi when a directory URL is requested, whichever it finds in the directory. Unlimited flexible URL rewriting and aliasing. Apache has no fixed limit on the numbers of Aliases and Redirects, which may be declared in the config files. In addition, a powerful rewriting engine can be used to solve most URL manipulation problems.

Content Negotiation

The ability to automatically serve clients of varying sophistication and HTML level compliance, with documents which offer the best representation of information that the client is capable of accepting.

Virtual Hosts

A much requested feature, sometimes known as multi-homed servers. This allows the server to distinguish between requests made to different IP addresses or names (mapped to the same machine). Apache also offers dynamically configurable mass-virtual hosting.

Configurable Reliable Piped Logs

You can configure Apache to generate logs in the format that you want. In addition, on most Unix architectures, Apache can send log files to a pipe, allowing for log rotation, hit filtering, real-time splitting of multiple vhosts into separate logs, and asynchronous DNS resolving on the fly. Taking your first steps into Apache territory are a lot easier than you think. If you are a regular Unix user, you will be well prepared to set up Apache. And the Windows version has the same install procedure as any other Windows application.

The first step is to obtain a copy of Apache HTTP Server. Because it's free there's no reason not to

give it a try <http://httpd.apache.org> . The download files are 4.2MB for the Windows version, and a little over 4.7MB for the Unix version. It is possible to write slim, high-end applications.

Install Apache Web Server on Ubuntu

Install Apache

Open Terminal in UBUNTU and open Terminal by using CTRL+T

Type the following

```
Step -1: $ sudo apt-get install apache2
```

Test it

Surf to your own server on your own local computer:

```
$ firefox "http://localhost"
```

Find out how your server is found on the net. Anyone can use your IP address or DNS name to connect to your server:

```
$ ip addr
```

Use your own IP address (eth0, first IP number, does not end .255)

```
$ firefox "http://1.2.3.4"
```

If DNS is working on your network, you can see your fully qualified domain name with:

```
$ host 1.2.3.4
```

User homepages

With user homepages, anything you put in public_html in your home directory is published on the web.

Enabling Userdir on Server

For newer Ubuntu such as 8.04 Hardy, you have to explicitly enable userdir module. On older ones such as 6.06 LTS Dapper, userdir is enabled by default.

```
$ sudo a2enmod userdir  
$ sudo /etc/init.d/apache2 restart
```

Test homepages

Go to your home directory

```
$ cd
```

Create a folder for public html files, note the underscore `_'

```
$ mkdir public_html
```

Check your name

```
$ whoami
```

Surf to ~yourname:

```
$ firefox "http://localhost/~tkarvine/"
```

You should see a directory listing of an empty directory. You can also try it using your IP address and a different machine.

If you see your homepage, well done. You have successfully installed a server with user homepages. You can create some web pages with OpenOffice and save them to your public_html directory.

Second Method

- Open Terminal (Application -> Accessories -> Terminal) and execute the following command:
- `sudo apt-get install apache2`
- When the setup is complete you can check if the Apache is working properly by pointing your browser to <http://localhost>. If you see the text "It works!", it means Apache is working just fine.
- Towards the end of the installation if you see a message like this inside Terminal, "*Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName*", you can fix this by executing the following command. It will open Gedit (text editor).
- `gksu gedit /etc/apache2/conf.d/fqdn`
- When Gedit opens, type "ServerName localhost" inside the file and click Save. Then close the file

Configuration

Start, Stop and Restart Apache

- After you have installed Apache, it will be added to the init.d list and will autostart

whenever you boot up your computer. The following commands allow you to

Start,
Restart,
Stop Apache.

```
sudo /etc/init.d/apache2 start    #start apache
sudo /etc/init.d/apache2 stop    #stop apache
sudo /etc/init.d/apache2 restart #restart
apache
```

- Apache is configured by placing directives in plain text configuration files.
- The main configuration file is called `apache2.conf`.
- In addition, other configuration files may be added using the `Include` directive, and wildcards can be used to include many configuration files.
- Any directive may be placed in any of these configuration files. Changes to the main configuration files are only recognized by Apache2 when it is started or restarted.
- The server also reads a file containing mime document types; the filename is set by the `TypesConfig` directive, and is `mime.types` by default.
- The default Apache2 configuration file is `/etc/apache2/apache2.conf`. You can edit this file to configure the Apache2 server. You can configure the port number, document root, modules, log files, virtual hosts, etc.

HTTPS Configuration

- The `mod_ssl` module adds an important feature to the Apache2 server - the ability to encrypt communications. Thus, when your browser is communicating using SSL, the `https://` prefix is used at the beginning of the Uniform Resource Locator (URL) in the browser navigation bar.
- The `mod_ssl` module is available in `apache2-common` package. If you have installed this package, you can run the following command from a terminal prompt to enable the `mod_ssl` module:
- `sudo a2enmod ssl`
- In order for Apache to provide HTTPS, a certificate and key file are also needed. For information on generating a key and obtaining a certificate.
-

XAMPP

- XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make

it the ideal tool for students developing and testing applications in PHP and MySQL.

- XAMPP is available as a free download in two specific packages: full and lite.
- While the full package download provides a wide array of development tools, this article will focus on using XAMPP Lite which contains the necessary technologies that meet the Ontario

Skills Competition standards. As the name implies, the light version is a small package containing Apache HTTP Server, PHP, MySQL, phpMyAdmin, Openssl, and SQLite.

Obtaining and Installing XAMPP

- As previously mentioned, XAMPP is a free package available for download and use for various web development tasks. All XAMPP packages and add-ons are distributed through the Apache Friends website at the address: <http://www.apachefriends.org/>.
- Once on the website, navigate and find the Windows version of XAMPP Lite and download the self-extracting ZIP archive. After downloading the archive, run and extract its contents into the root path of a hard disk or USB drive.
- If extracted properly you will notice a new xampplite directory in the root of your installation disk.
- In order to test that everything has been installed correctly, first start the Apache HTTP Server by navigating to the xampplite directory and run the apache_start.bat batch file.
- The below Screen Appears
-



- In order to stop all Apache processes do not close the running terminal application, but instead run another batch file in the xampplite directory called apache_stop.bat.
- The next step is to write a simple “Hello World” program in PHP that will test the configuration of PHP under XAMPP.
- In the previous pages we installed and ran the Apache HTTP Server with success. The next step is to write a simple “Hello World” program in PHP that will test the configuration of PHP under XAMPP.
- In a text editor, such as TextPad, write the following lines of code:

Example:

```
<HTML>
```

```
<Head>
```

```
<Title>Hello World</Title>
```

```

</Head>

    <Body>

?> echo "Hello World": ?>

/Body> />Html

```



FIGURE 2: Resulting output for helloWorld.php.

- The above code example starts a new XHTML document with the page title set as “Hello World” and then prints a single line of text, Hello World, using the echo PHP language construct.
- In order to run and test this PHP program, save it into your xampplite\htdocs directory with the file name of helloWorld.php. Then, to view it in your browser, simply type in http://localhost/helloWorld.php into the address bar. If done correctly, you will see your Hello World line of text in the web browser as shown in FIGURE 2.

Windows Install

- Installing PHP/MySQL on Windows using XAMPP requires nothing more than mastering the following skills:
- Using an Internet Browser, such Internet Explorer, Mozilla, or Firebird Locating the xampp folder that was created by the extract Double-clicking on a few files
- If you've got these basic skills, you're ready to go!
- Click on this link <http://www.apachefriends.org/en/xampp-windows.html> to download the latest version of XAMPP for Windows.. The EXE (7-zip) is recommended. It is smaller (almost half the ZIP version) and also self-extracting.
- Download it to the C:\ directory, double-clion it and a dialogue box appears.

- Click the Extract button.
- Everything will be extracted to a created folder called "xampp" under the C:\ directory. (You could, if you like, place it in another subdirectory of your choosing.)
- Double-click on the file setup_xampp.bat which is in the subdirectory containing the XAMPP installation.
- Follow the directions it displays on the screen.

Configuration

- The "Apache" and "MySQL" folders created by the XAMPP installation will have batch files for installing Apache and MySQL as services. Run these two batch files.
- Then create a shortcut in your Windows Startup folder
- that points to the file "ApacheMonitor.exe" which is located in the \apache\bin subdirectory. Finally, go to the Windows Control Panel and configure these services - Apache and MySQL - to start manually.
- You want to minimize the services that run automatically for security purposes.
- The next time you start up your Windows machine, the Apache Monitor will appear in your system tray.
- Double-click on it to open it up, click on the button to start up Apache.
- Click on the Services button to bring up the Services window to start MySQL.
- The ApacheMonitor is handy for when you want to restart Apache after making changes to the php.ini.
 - A few helpful notes:
- The php.ini file in \xampp\apache\bin is the one to edit when PHP runs as an apache module
- Web pages go in the \xampp\htdocs directory (the DocumentRoot) Files for the XAMPP splash page are in \xampp\htdocs\xampp
- Default installation of XAMPP is "wide open", i.e., MySQL has no root password and register_globals is On
- If you can't see the flag icons to switch from German to English, change scrolling to "yes" in index.html
- After installing MySQL as a service, remember to copy \xampp\mysql\my_example.cnf to C:\my.cnf
- The file type may indicate a "SpeedDial" shortcut, but MySQL will treat it as a configuration file

WAMP

- WAMP is an acronym for **Windows, Apache, MySQL and PHP**. It is a combination of independently created software's bundled together into a SINGLE installation package to set up a SERVER on your machine with out any hassles. It comes with the GPL license.

Contents Of WAMP Server (Package)

- Apache
- My Sql
- Paperphp
- MYAdmin
- SQLite_Manager
- Zend Optimizer

OBTAINING THE RESOURCES

- So you need a Windows desktop server ... but what EXACTLY do you need?
- For a basic Windows server you need the following:
 - Apache (The most commonly used web server software)
 - MySQL (A relational database for holding all the information)
 - PHP (a scripting language used for producing dynamic web pages)
- This combination of software is known as WAMP server software (Windows, Apache, MySQL, PHP) and that's what you will need if you are using a Windows computer.
- If you have a Linux computer, it's known as LAMP server software. (If you're into Linux, then you probably already know how to install a LAMP server). This eBook concentrates on the vast majority of people, who will be using Windows.
- So ... you need Apache, MySQL, and PHP ... The good news ... they're all FREE and they're all easy to obtain.
- Now you could search the internet for all three pieces of software and install them separately But a much better way is to install a single package that already contains all three in a single application (quicker and easier).
- There are several such packages available but we're going to use one called WampServer which is available from <http://www.wampserver.com/en/index.php> From the home page, click on the "Download the latest release of Wampserver" link.
- There are two versions available, depending upon whether you are using 32bit or 64 bit Windows (If you're unsure go to your Windows Control Panel, click on "System" and it'll tell you what bit you have).
- Once you're sure which version to download (32 bit or 64 bit) simply click on the appropriate download link and download WampServer to your desktop.

INSTALLING THE SERVER SOFTWARE

- Once the software has finished downloading, find it on your desktop (Figure 1)



Figure 1

- Double click on it to start the installation process. (Figure 2)

- - Figure-2
- Close any other applications you've got open, then click on the “Next” button and read the License Agreement (Figure 3)



- Click on “I accept the agreement”, then click on the “Next” button.
- You can choose where WampServer2 is installed. (Figure 4)



Figure 3



Figure 4

-
- I would recommend leaving the default of “C:\wamp”, then click on the “Next” button.
- The next choice (Figure 5) is a matter of personal preference. Leave blank or make your choice, then click on the “Next” button.



Figure 5

-
- Click on “Install” (Figure 6)



Figure 6

- Wait (Figure 7)! Press the wall switch to “ON” and boil the kettle. Make yourself a coffee!



Figure 7

-
- WampServer2 may be able to tell what browser you usually use. If not, it will display a page like this (Figure 8) and ask you to choose.

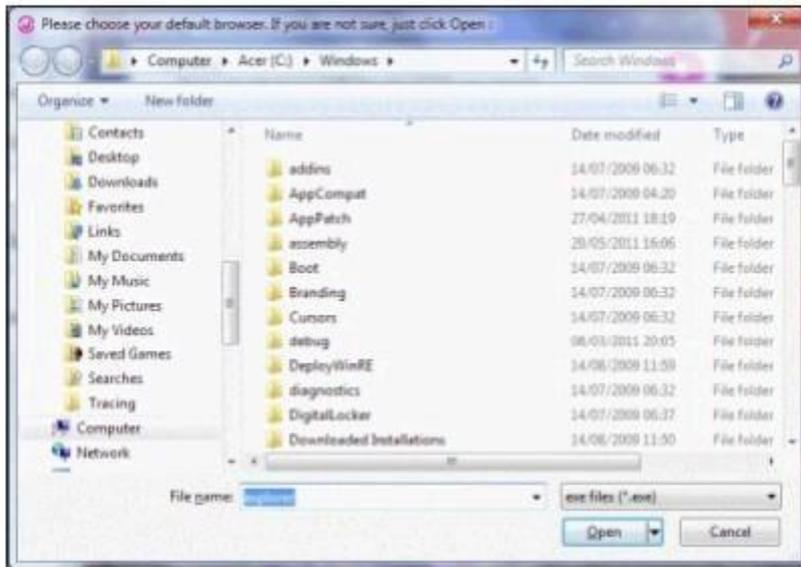


Figure 8

-
- Choose your browser then click on the “Open” button. If you’re not sure, just click on the “Open” button.
- Depending on your firewall settings, you may be asked to allow the Apache Server to access your networks. (Figure 9)



Figure 9

-
- Click on “Allow access” to your private network.
- Leave the default values. (Figure 10) Then click on the “Next” button.

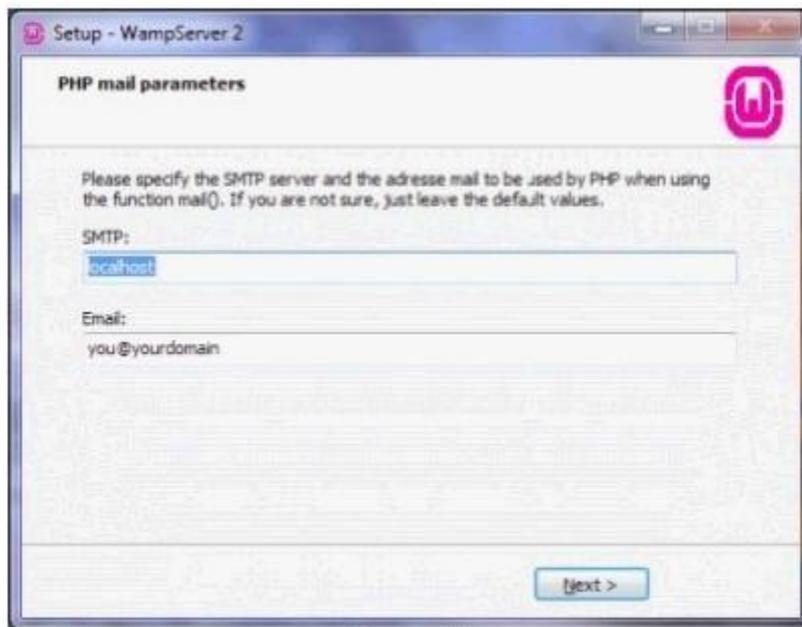


Figure 10

-
- Click on the “Launch WampServer 2 now” box to enter a tick. (Figure 11). Then click on the “Finish” button



Figure 11

LAUNCHING YOUR SERVER

- When WampServer2 has launched, you should see an icon like this (Figure 12) at the bottom right of your computer screen.



Figure 12

If you can't see it, go to the small triangle symbol at the bottom of your screen. (Figure 13)



Figure 13

Clicking on the white triangle will display the hidden icons. (Figure 14)



Figure 14

- Then click on the green “W” WampServer2 icon. Clicking the WampServer2 icon displays the Wamp Server menu. (Figure 15). From where you can access all the various parts of the server.

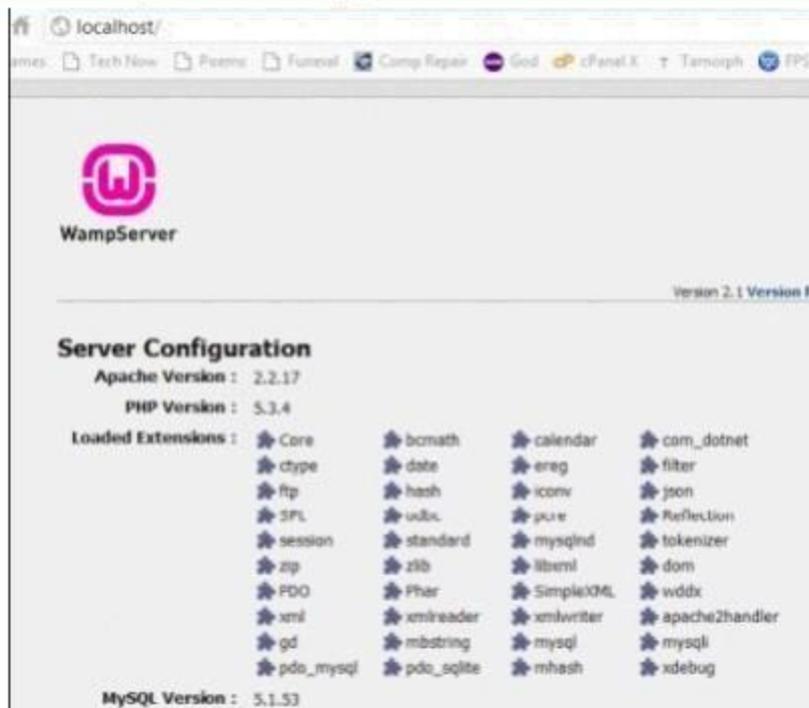


Click on the top part of the menu, where it says “Localhost” (Figure 16)



Figure 16

- If your installation was completed correctly, your browser should look like this (Figure 17)



-
- Your WampServer2 will close down whenever you close Windows, and it won't re-start automatically when you re-start windows.
- To re-start the server simply click on the icon on your desktop or launch bar, or go to "Start", "All Programs".

HTTP messages

- A HTTP message consists of a header and an optional body.
- The message header of an HTTP request consists of a request line and a collection of header fields.
- The message header of an HTTP response consists of a status line and a collection of header fields.
- All HTTP messages must include the protocol version. Some HTTP messages can optionally enclose a content body.
- `HttpCore` defines the HTTP message object model to closely follow this definition and provides extensive support for serialization (formatting) and deserialization (parsing) of HTTP message elements.

HTTP request message

- HTTP request is a message sent from the client to the server. The first line of that message includes the method to be applied to the resource, the identifier of the resource, and the protocol version in use.

```
o HttpRequest request = new BasicHttpRequest("GET", "/",  
o     HttpVersion.HTTP_1_1);  
o  
o System.out.println(request.getRequestLine().getMethod());  
o System.out.println(request.getRequestLine().getUri());  
o System.out.println(request.getProtocolVersion());  
o System.out.println(request.getRequestLine().toString());
```

HTTP response message

- HTTP response is a message sent by the server back to the client after having received and interpreted a request message. The first line of that message consists of the protocol version followed by a numeric status code and its associated textual phrase.

```
➤ HttpResponse response = new  
BasicHttpResponse(HttpVersion.HTTP_1_1,  
➤     HttpStatus.SC_OK, "OK");  
➤  
➤ System.out.println(response.getProtocolVersion());  
➤ System.out.println(response.getStatusLine().getStatusCode());  
➤ System.out.println(response.getStatusLine().getReasonPhrase());  
➤ System.out.println(response.getStatusLine().toString());
```

HTML Basics

Welcome to HTML Basics. This workshop leads you through the basics of Hyper Text Markup Language (HTML). HTML is the building block for web pages. You will learn to use HTML to author an HTML page to display in a web browser.

Objectives:

By the end of this workshop, you will be able to:

- Use a text editor to author an HTML document.
- Be able to use basic tags to denote paragraphs, emphasis or special type.
- Create hyperlinks to other documents.
- Create an email link.
- Add images to your document.
- Use a table for layout.
- Apply colors to your HTML document.

Prerequisites:

You will need a text editor, such as Notepad and an Internet browser, such as Internet Explorer or Netscape.

Q: What is Notepad and where do I get it?

A: Notepad is the default Windows text editor. On most Windows systems, click your Start button and choose Programs then Accessories. It should be a little blue notebook.

Mac Users: SimpleText is the default text editor on the Mac. In OSX use TextEdit and change the following preferences: Select (in the preferences window) Plain text instead of Rich text and then select Ignore rich text commands in HTML files. This is very important because if you don't do this HTML codes probably won't work.

One thing you should avoid using is a word processor (like Microsoft Word) for authoring your HTML documents.

What is an html File?

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension

Try It?

Open your text editor and type the following text:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

Save the file as **mypage.html**. Start your Internet browser. Select **Open** (or Open Page) in the **File** menu of your browser. A dialog box will appear. Select **Browse** (or Choose File) and locate the html file you just created - **mypage.html** - select it and click **Open**. Now you should see an address in the

dialog box, for example **C:\MyDocuments\mypage.html**. Click **OK**, and the browser will display the page. To view how the page should look, visit this web page:
<http://profdevtrain.austincc.edu/html/mypage.html>

Example Explained

What you just made is a skeleton html document. This is the minimum required information for a web document and all web documents should contain these basic components. The first tag in your html document is `<html>`. This tag tells your browser that this is the start of an html document. The last tag in your document is `</html>`. This tag tells your browser that this is the end of the html document.

The text between the `<head>` tag and the `</head>` tag is header information. Header information is not displayed in the browser window.

The text between the `<title>` tags is the title of your document. The `<title>` tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

The text between the `<body>` tags is the text that will be displayed in your browser.

The text between the `` and `` tags will be displayed in a bold font.

HTM or HTML Extension?

When you save an HTML file, you can use either the .htm or the .html extension. The .htm extension comes from the past when some of the commonly used software only allowed three letter extensions. It is perfectly safe to use either .html or .htm, but be consistent. **mypage.htm** and **mypage.html** are treated as different files by the browser.

How to View HTML Source

A good way to learn HTML is to look at how other people have coded their html pages. To find out, simply click on the View option in your browsers toolbar and select Source or Page Source. This will open a window that shows you the actual HTML of the page. Go ahead and view the source html for this page.

HTML Tags

What are HTML tags?

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters `<` and `>`
- The surrounding characters are called angle brackets
- HTML tags normally come in pairs like `` and ``
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start and end tags is the element content
- HTML tags are not case sensitive, `` means the same as ``

Logical vs. Physical Tags

In HTML there are both logical tags and physical tags. Logical tags are designed to describe (to the browser) the enclosed text's meaning. An example of a logical tag is the `` `` tag. By placing text in between these tags you are telling the browser that the text has some greater importance. By default all browsers make the text appear bold when in between the `` and `` tags.

Physical tags on the other hand provide specific instructions on how to display the text they enclose. Examples of physical tags include:

- ``: Makes the text bold.
- `<big>`: Makes the text usually one size bigger than what's around it.
- `<i>`: Makes text italic.

Physical tags were invented to add style to HTML pages because style sheets were not around, though the original intention of HTML was to not have physical tags. Rather than use physical tags to style your HTML pages, you should use style sheets.

HTML Elements

Remember the HTML example from the previous page:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

This is an HTML element:

```
<b>This text is bold</b>
```

The HTML element begins with a start tag: ``

The content of the HTML element is: This text is bold

The HTML element ends with an end tag: ``

The purpose of the `` tag is to define an HTML element that should be displayed as bold.

This is also an HTML element:

```
<body>
This is my first homepage. <b>This text is bold</b>
</body>
```

This HTML element starts with the start tag `<body>`, and ends with the end tag `</body>`. The purpose of the `<body>` tag is to define the HTML element that contains the body of the HTML document.

Nested Tags

You may have noticed in the example above, the `<body>` tag also contains other tags, like the `` tag. When you enclose an element in with multiple tags, the last tag opened should be the first tag closed. For example:

```
<p><b><em>This is NOT the proper way to close nested tags.</em></b></p>
```

```
<p><b><em>This is the proper way to close nested tags.</em></b></p>
```

Note: It doesn't matter which tag is first, but they must be closed in the proper order.

Why Use Lowercase Tags?

You may notice we've used lowercase tags even though I said that HTML tags are not case sensitive. `` means the same as ``. The World Wide Web Consortium (W3C), the group responsible for developing web standards, recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) requires lowercase tags.

Tag Attributes

Tags can have attributes. Attributes can provide additional information about the HTML elements on your page. The `<tag>` tells the browser to do something, while the attribute tells the browser how to do it. For instance, if we add the `bgcolor` attribute, we can tell the browser that the background color of your page should be blue, like this: `<body bgcolor="blue">`.

This tag defines an HTML table: `<table>`. With an added border attribute, you can tell the browser that the table should have no borders: `<table border="0">`. Attributes always come in name/value pairs like this: `name="value"`. Attributes are always added to the start tag of an HTML element and the value is surrounded by quotes.

Quote Styles, "red" or 'red'?

Attribute values should always be enclosed in quotes. Double style quotes are the most common, but single style quotes are also allowed. In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes:

```
name='George "machine Gun" Kelly'
```

Note: Some tags we will discuss are deprecated, meaning the World Wide Web Consortium (W3C) the governing body that sets HTML, XML, CSS, and other technical standards decided those tags and attributes are marked for deletion in future versions of HTML and XHTML. Browsers should continue to support deprecated tags and attributes, but eventually these tags are likely to become obsolete and so future support cannot be guaranteed.

For a complete list of tags, visit W3C.org.

Basic HTML Tags

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Basic HTML Tags

Tag	Description
<code><html></code>	Defines an HTML document
<code><body></code>	Defines the document's body
<code><h1></code> to <code><h6></code>	Defines header 1 to header 6
<code><p></code>	Defines a paragraph
<code>
</code>	Inserts a single line break
<code><hr></code>	Defines a horizontal rule
<code><!--></code>	Defines a comment

Headings

Headings are defined with the `<h1>` to `<h6>` tags. `<h1>` defines the largest heading while `<h6>` defines the smallest.

`<h1>This is a heading</h1>`

`<h2>This is a heading</h2>`

`<h3>This is a heading</h3>`

`<h4>This is a heading</h4>`

`<h5>This is a heading</h5>`

`<h6> This is a heading</h6>`

HTML automatically adds an extra blank line before and after a heading. A useful heading attribute is `align`.

```
<h5 align="left">I can align headings </h5>
```

```
<h5 align="center">This is a centered heading </h5>
```

```
<h5 align="right">This is a heading aligned to the right </h5>
```

Paragraphs

Paragraphs are defined with the `<p>` tag. Think of a paragraph as a block of text. You can use the align attribute with a paragraph tag as well.

```
<p align="left">This is a paragraph</p>
```

```
<p align="center">this is another paragraph</p>
```

Important: You must indicate paragraphs with `<p>` elements. A browser ignores any indentations or blank lines in the source text. Without `<p>` elements, the document becomes one large paragraph. HTML automatically adds an extra blank line before and after a paragraph.

Line Breaks

The `
` tag is used when you want to start a new line, but don't want to start a new paragraph. The `
` tag forces a line break wherever you place it. It is similar to single spacing in a document.

This Code	Would Display
<pre><p>This
 is a para
 graph with line breaks</p></pre>	This is a para graph with line breaks

The `
` tag has no closing tag.

Horizontal Rule

The `<hr>` element is used for horizontal rules that act as dividers between sections, like this:

The horizontal rule does not have a closing tag. It takes attributes such as align and width. For instance:

This Code	Would Display
<pre><hr width="50%" align="center"></pre>	<hr style="width: 50%; margin: auto;"/>

Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment can be placed anywhere in the document and the browser will ignore everything inside the brackets. You can use comments to write notes to yourself, or write a helpful message to someone looking at your source code.

This Code	Would Display
<pre><p> This html comment would <!-- This is a comment --> be displayed like this.</p></pre>	This HTML comment would be displayed like this.

Notice you don't see the text between the tags `<!--` and `-->`. If you look at the source code, you would see the comment. To view the source code for this page, in your browser window, select **View** and then select **Source**.

Note: You need an exclamation point after the opening bracket <!-- but not before the closing bracket -->.

HTML automatically adds an extra blank line before and after some elements, like before and after a paragraph, and before and after a heading. If you want to insert blank lines into your document, use the
 tag.

Try It Out!

Open your text editor and type the following text:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1 align="center">My First Webpage</h1>
<p>Welcome to my first web page. I am writing this page using a text editor and plain
old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br>
which I am of course.</p>
</body>
</html>
```

Save the page as **mypage2.html**. Open the file in your Internet browser. To view how the page should look, visit this web page: <http://profdevtrain.austincc.edu/html/mypage2.html>

Other HTML Tags

As mentioned before, there are logical styles that describe what the text should be and physical styles which actually provide physical formatting. It is recommended to use the logical tags and use style sheets to style the text in those tags.

Logical Tags

Tag	Description
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines an address element
<cite>	Defines a <i>citation</i>
<code>	Defines <code>computer code text</code>
<blockquote>	Defines a long quotation
	Defines text
<dfn>	Defines a <i>definition</i> term
	Defines <i>emphasized</i> text
<ins>	Defines inserted text
<kbd>	Defines keyboard text
<pre>	Defines preformatted text
<q>	Defines a short quotation
<samp>	Defines sample computer code
	Defines strong text
<var>	Defines a <i>variable</i>

Physical Tags

Tag	Description
	Defines bold text
<big>	Defines big text
<i>	Defines <i>italic</i> text
<small>	Defines small text
<sup>	Defines ^{superscripted} text
<sub>	Defines _{subscripted} text
<tt>	Defines teletype text
<u>	Deprecated. Use styles instead

Character tags like and produce the same physical display as and <i> but are more uniformly supported across different browsers.

Some Examples:

The following paragraph uses the `<blockquote>` tag. In the previous sentence, the blockquote tag is enclosed in the `<samp>` Sample tag.

We the people of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our Posterity, do ordain and establish this Constitution for the United States of America.

Although most browsers render blockquoted text by indenting it, that's not specifically what it's designed to do. It's conceivable that some future browser may render blockquoted text in some other way. However, for the time being, it is perfectly safe to indent blocks of text with the `<blockquote>`.

This Code	Would Display
<code><abbr title="World Wide Web">WWW</abbr></code>	<u>WWW</u>

When you hold your mouse pointer over the WWW, text in the title attribute will appear in.

HTML Character Entities

Some characters have a special meaning in HTML, like the less than sign (`<`) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in place of the actual characters themselves.

The Most Common Character Entities:

Result	Description	Entity Name	Entity Number
	non-breaking space	<code>&nbsp;</code>	<code>&#160;</code>
<code><</code>	less than	<code>&lt;</code>	<code>&#60;</code>
<code>></code>	greater than	<code>&gt;</code>	<code>&#62;</code>
<code>&</code>	ampersand	<code>&amp;</code>	<code>&#38;</code>
<code>"</code>	quotation mark	<code>&quot;</code>	<code>&#34;</code>
<code>'</code>	apostrophe	<code>&apos;</code> (does not work in IE)	<code>&#39;</code>

A character entity has three parts: an ampersand (`&`), an entity name or an entity number, and finally a semicolon (`;`). The `&` means we are beginning a special character, the `;` means ending a special character and the letters in between are sort of an abbreviation for what it's for. To display a less than sign in an HTML document we must write: **`<`** or **`<`**; The advantage of using a name instead of a number is that a name is easier to remember. The disadvantage is that not all browsers support the newest entity names, while the support for entity numbers is very good in almost all browsers.

Note: Entities are case sensitive.

Non-breaking Space

The most common character entity in HTML is the non-breaking space **` `**. Normally HTML will truncate spaces in your text. If you add 10 spaces in your text, HTML will remove 9 of them. To add spaces to your text, use the ` ` character entity.

This Code	Would Display
<code><p> This code would appear as this.</p></code>	This code would appear as this.


```

<html>
<head>
<title>My First Webpage</title>
</head>
<body background="http://profdevtrain.austincc.edu/html/graphics/clouds.gif"
bgcolor="#EDDD9E">
<h1 align="center">My First Webpage</h1>
<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text
editor and plain old html.</p>
<p>By learning html, I'll be able to create webpages like a <del>beginner</del>
pro....<br>
which I am of course.</p>
</body>
</html>

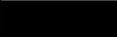
```

Save your page as **mypage3.html** and view it in your browser. To view how the page should look, visit this web page: **<http://profdevtrain.austincc.edu/html/mypage3.html>**
Notice we gave our page a background color as well as a background image. If for some reason the web page is unable to find the picture, it will display our background color.

HTML Colors

Color Values

Colors are defined using a hexadecimal notation for the combination of red, green, and blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00). The highest value is 255 (hex #FF). This table shows the result of combining red, green, and blue:

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

Color Names

A collection of color names is supported by most browsers. To view a table of color names that are supported by most browsers visit this web page:

http://profdevtrain.austincc.edu/html/color_names.htm

Note: Only 16 **color names** are supported by the W3C HTML 4.0 standard (aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow). For all other colors you should use the **Color HEX** value.

Color	Color HEX	Color Name
	#F0F8FF	AliceBlue
	#FAEBD7	AntiqueWhite
	#7FFFD4	Aquamarine
	#000000	Black
	#0000FF	Blue
	#8A2BE2	BlueViolet
	#A52A2A	Brown

Web Safe Colors

A few years ago, when most computers supported only 256 different colors, a list of 216 Web Safe Colors was suggested as a Web standard. The reason for this was that the Microsoft and Mac operating system used 40 different "reserved" fixed system colors (about 20 each). This 216 cross platform web safe color palette was originally created to ensure that all computers would display all colors correctly when running a 256 color palette. To view the 216 Cross Platform Colors visit this web page: <http://profdevtrain.austincc.edu/html/216.html>

16 Million Different Colors

The combination of Red, Green and Blue values from 0 to 255 gives a total of more than 16 million different colors to play with (256 x 256 x 256). Most modern monitors are capable of displaying at least 16,384 different colors. To assist you in using color schemes, check out <http://wellstyled.com/tools/colorscheme2/index-en.html>. This site lets you test different color schemes for page backgrounds, text and links.

HTML Lists

HTML provides a simple way to show unordered lists (bullet lists) or ordered lists (numbered lists).

Unordered Lists

An unordered list is a list of items marked with bullets (typically small black circles). An unordered list starts with the `` tag. Each list item starts with the `` tag.

This Code	Would Display
<pre> Coffee Milk </pre>	<ul style="list-style-type: none">▪ Coffee▪ Milk

Ordered Lists

An ordered list is also a list of items. The list items are marked with numbers. An ordered list starts with the `` tag. Each list item starts with the `` tag.

This Code	Would Display
<pre> Coffee Milk </pre>	<ol style="list-style-type: none">1. Coffee2. Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

Definition Lists

Definition lists consist of two parts: a **term** and a **description**. To mark up a definition list, you need three HTML elements; a container `<dl>`, a definition term `<dt>`, and a definition description `<dd>`.

This Code	Would Display
<pre><dl> <dt>Cascading Style Sheets</dt> <dd>Style sheets are used to provide presentational suggestions for documents marked up in HTML. </dd> </dl></pre>	<p>Cascading Style Sheets Style sheets are used to provide presentational suggestions for documents marked up in HTML.</p>

Inside a definition-list definition (the `<dd>` tag) you can put paragraphs, line breaks, images, links, other lists, etc

Try It Out

Open your text editor and type the following:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body bgcolor="#EDDD9E">
<h1 align="center">My First Webpage</h1>
<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text
editor and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br>
which I am of course.</p>
Here's what I've learned:
<ul>
<li>How to use HTML tags</li>
<li>How to use HTML colors</li>
<li>How to create Lists</li>
</ul>
</body>
</html>
```

Save your page as **mypage4.html** and view it in your browser. To see how your page should look visit this web page: **<http://profdevtrain.austincc.edu/html/mypage4.html>**

HTML Links

HTML uses the <a> anchor tag to create a link to another document or web page.

The Anchor Tag and the Href Attribute

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc. The syntax of creating an anchor:

```
<a href="url">Text to be displayed</a>
```

The <a> tag is used to create an anchor to link from, the href attribute is used to tell the address of the document or page we are linking to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

This Code	Would Display
<code>Visit ACC!</code>	Visit ACC!

The Target Attribute

With the target attribute, you can define **where** the linked document will be opened. By default, the link will open in the current window. The code below will open the document in a new browser window:

```
<a href=http://www.austincc.edu/ target="_blank">Visit ACC!</a>
```

Email Links

To create an email link, you will use mailto: plus your email address. Here is a link to ACC's Help Desk:

```
<a href="mailto:helpdesk@austincc.edu">Email Help Desk</a>
```

To add a subject for the email message, you would add ?subject=after the email address. For example:

```
<a href="mailto:helpdesk@austincc.edu?subject=Email Assistance">Email Help Desk</a>
```

The Anchor Tag and the Name Attribute

The name attribute is used to create a named anchor. When using named anchors we can create links that can jump directly to a specific section on a page, instead of letting the user scroll around to find what he/she is looking for. Unlike an anchor that uses href, a named anchor doesn't change the appearance of the text (unless you set styles for that anchor) or indicate in any way that there is anything special about the text. Below is the syntax of a named anchor:

```
<a name="top">Text to be displayed</a>
```

To link directly to the top section, add a # sign and the name of the anchor to the end of a URL, like this:

This Code	Would Display
<pre>Back to top of page </pre> <p>A hyperlink to the top of the page from within the file 10links.html will look like this:</p> <pre>Back to top of page </pre>	<p>Back to top of page</p> <p>Back to top of page</p>

Note: Always add a trailing slash to subfolder references. If you link like this: href="http://profdevtrain.austincc.edu/html", you will generate two HTTP requests to the server, because the server will add a slash to the address and create a new request like this: href="http://profdevtrain.austincc.edu/html/"

Named anchors are often used to create "table of contents" at the beginning of a large document. Each chapter within the document is given a named anchor, and links to each of these anchors are put at the top of the document. If a browser cannot find a named anchor that has been specified, it goes to the top of the document. No error occurs.

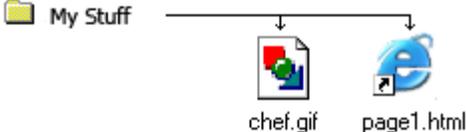
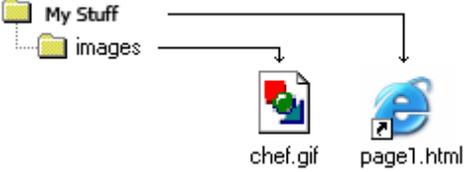
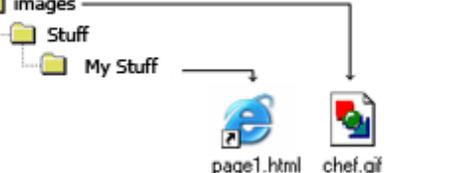
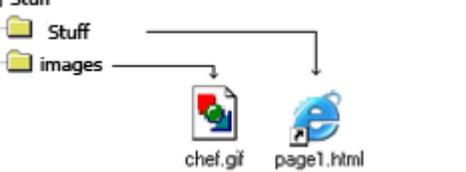
HTML Images

The Image Tag and the Src Attribute

The tag is empty, which means that it contains attributes only and it has no closing tag. To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page. The syntax of defining an image:

This Code	Would Display
<pre></pre>	

Not only does the source attribute specify what image to use, but where the image is located. The above image, graphics/chef.gif, means that the browser will look for the image name **chef.gif** in a **graphics** folder in the same folder as the html document itself.

	<p><code>src="chef.gif"</code> means that the image is in the same folder as the html document calling for it.</p>
	<p><code>src="images/chef.gif"</code> means that the image is one folder down from the html document that called for it. This can go on down as many layers as necessary.</p>
	<p><code>src="../chef.gif"</code> means that the image is in one folder up from the html document that called for it.</p>
	<p><code>src="../../chef.gif"</code> means that the image is two folders up from the html document that called for it.</p>
	<p><code>src="../images/chef.gif"</code> means that the image is one folder up and then another folder down in the images directory.</p>
	<p><code>src="../../../other/images/chef.gif"</code> means this goes multiple layers up.</p>

The browser puts the image where the image tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

The Alt Attribute

The alt attribute is used to define an alternate text for an image. The value of the alt attribute is author-defined text:

```

```

The alt attribute tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image. It is a good practice to include the alt attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers or use screen readers.

Image Dimensions

When you have an image, the browser usually figures out how big the image is all by itself. If you put in the image dimensions in pixels however, the browser simply reserves a space for the image, then

loads the rest of the page. Once the entire page is loads it can go back and fill in the images. Without dimensions, when it runs into an image, the browser has to pause loading the page, load the image, then continue loading the page. The chef image would then be:

```

```

Open the file **mypage2.html** in your text editor and add code highlighted in bold:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1 align="center">My First Web page</h1>
<p>Welcome to my first webpage. I am writing this page using a text editor and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br>
which I am of course.</p>
<!-- Who would have guessed how easy this would be :) -->
<p></p>
<p align="center">This is my Chef</p>
</body>
</html>
```

Save your page as **mypage5.html** and view it in your browser. To see how your page should look visit this web page: <http://profdevtrain.austincc.edu/html/mypage5.html>

Tables

Tables are defined with the `<table>` tag. A table is divided into rows (with the `<tr>` tag), and each row is divided into data cells (with the `<td>` tag). The letters td stands for table data, which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

This Code	Would Display
<pre><table> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr> </table></pre>	<pre>row 1, cell 1 row 1, cell 2 row 2, cell 1 row 2, cell 2</pre>

Tables and the Border Attribute

To display a table with borders, you will use the border attribute.

This Code	Would Display
<pre><table border="1"> <tr> <td>Row 1, cell 1</td> <td>Row 1, cell 2</td> </tr> </table></pre>	<pre>row 1, cell 1 row 1, cell 2</pre>

and....

This Code	Would Display		
<pre><table border="5"> <tr> <td>Row 1, cell 1</td> <td>Row 1, cell 2</td> </tr> </table></pre>	<table border="5"><tr><td>row 1, cell 1</td><td>row 1, cell 2</td></tr></table>	row 1, cell 1	row 1, cell 2
row 1, cell 1	row 1, cell 2		

Open up your text editor. Type in your `<html>`, `<head>` and `<body>` tags. From here on I will only be writing what goes between the `<body>` tags. Type in the following:

```
<table border="1">
<tr>
<td>Tables can be used to layout information</td>
<td>&nbsp;  &nbsp; </td>
</tr>
</table>
```

Save your page as **mytable1.html** and view it in your browser. To see how your page should look visit this web page: <http://profdevtrain.austincc.edu/html/mytable1.html>

Headings in a Table

Headings in a table are defined with the `<th>` tag.

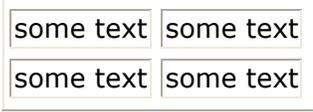
This code	Would Display						
<pre><table border="1"> <tr> <th>Heading</th> <th>Another Heading</th> </tr> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr> </table></pre>	<table border="1"><thead><tr><th>Heading</th><th>Another Heading</th></tr></thead><tbody><tr><td>row 1, cell 1</td><td>row 1, cell 2</td></tr><tr><td>row 2, cell 1</td><td>row 2, cell 2</td></tr></tbody></table>	Heading	Another Heading	row 1, cell 1	row 1, cell 2	row 2, cell 1	row 2, cell 2
Heading	Another Heading						
row 1, cell 1	row 1, cell 2						
row 2, cell 1	row 2, cell 2						

Cell Padding and Spacing

The `<table>` tag has two attributes known as cellspacing and cellpadding. Here is a table example without these properties. These properties may be used separately or together.

This Code	Would Display				
<pre><table border="1"> <tr> <td>some text</td> <td>some text</td> </tr> <tr> <td>some text</td> <td>some text</td> </tr> </table></pre>	<table border="1"><tr><td>some text</td><td>some text</td></tr><tr><td>some text</td><td>some text</td></tr></table>	some text	some text	some text	some text
some text	some text				
some text	some text				

Cellspacing is the pixel width between the individual data cells in the table (The thickness of the lines making the table grid). The default is zero. If the border is set at 0, the cellspacing lines will be invisible.

This Code	Would Display
<pre><table border="1" cellspacing="5"> <tr> <td>some text</td> <td>some text</td> </tr><tr> <td>some text</td> <td>some text</td> </tr> </table></pre>	

Cellpadding is the pixel space between the cell contents and the cell border. The default for this property is also zero. This feature is not used often, but sometimes comes in handy when you have your borders turned on and you want the contents to be away from the border a bit for easy viewing. Cellpadding is invisible, even with the border property turned on. Cellpadding can be handled in a style sheet.

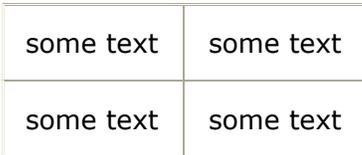
This Code	Would Display
<pre><table border="1" cellpadding="10"> <tr> <td>some text</td> <td>some text</td> </tr><tr> <td>some text</td> <td>some text</td> </tr> </table></pre>	

Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<colgroup>	Defines groups of table columns
<col>	Defines the attribute values for one or more columns in a table

Table Size

Table Width

The width attribute can be used to define the width of your table. It can be defined as a fixed width or a relative width. A fixed table width is one where the width of the table is specified in pixels. For example, this code, `<table width="550">`, will produce a table that is 550 pixels wide. A relative table width is specified as a percentage of the width of the visitor's viewing window. Hence this code, `<table width="80%">`, will produce a table that occupies 80 percent of the screen.

This table width is 250 pixels

This table width is 50%

There are arguments in favor of giving your tables a relative width because such table widths yield pages that work regardless of the visitor's screen resolution. For example, a table width of 100% will always span the entire width of the browser window whether the visitor has a 800x600 display or a 1024x768 display (etc). Your visitor never needs to scroll horizontally to read your page, something that is regarded by most people as being very annoying.

HTML Layout - Using Tables

One very common practice with HTML, is to use HTML tables to format the layout of an HTML page. A part of this page is formatted with two columns. As you can see on this page, there is a left column and a right column. This text is displayed in the left column.

An HTML <table> is used to divide a part of this Web page into two columns. The trick is to use a table without borders, and maybe a little extra cell-padding. No matter how much text you add to this page, it will stay inside its column borders.

Try It Out!

Let's put everything you've learned together to create a simple page. Open your text editor and type the following text:

```
<html>
<head>
<title>My First Web Page </title>
</head>
<body>
<table width="90%" cellpadding="5" cellspacing="0" >
  <tr bgcolor="#EDDD9E">
    <td width="200" valign="top"></td>
    <td valign="top"><h1 align="right">Janet Doeson</h1>
    <h3 align="right">Technical Specialist</h3></td>
  </tr>
  <tr>
    <td width="200">
      <h3>Menu</h3>
      <ul>
        <li><a href="home.html">Home</a></li>
        <li> <a href="faq.html">FAQ</a></li>
        <li> <a href="contact.html">Contact</a></li>
        <li> <a href="http://www.austincc.edu">Links</a> </li>
      </ul></td>
    <td valign="top"><h2 align="center">Welcome!</h2>
    <p>Welcome to my first webpage. I created this webpage without the assistance of a
webpage editor. Just my little text editor and a keen understanding of html.</p>
    <p>Look around. Notice I'm able to use paragraphs, lists and headings. You may not
be able to tell, but the layout is done with a table. I'm very clever. </p>
    <blockquote>
      <p>I always wanted to be somebody, but now I realize I should have been more
specific.</p>
      <cite>Lily Tomlin </cite> </blockquote>
    </td>
  </tr>
</table>
<hr width="90%" align="left">
<address>
  Janet Doeson<br>
  Technical Specialist<br>
  512.555.5555
</address>
<p>Contact me at <a href="mailto:jdoeson@acme.com">jdoeson@acme.com</a> </p>
```

```
</body>  
</html>
```

Save your page as **mytable2.html** and view it in your browser. To see how your page should look visit this web page: **<http://profdevtrain.austincc.edu/html/mytable2.html>**

I have indented some of the HTML code in the above example. Indenting the code can make your HTML document easier to read.

Create Your Own Page

It's time to create your own page. Use your text editor to create a page which contains the following:

- the required HTML page codes
- link to another web page
- an email link
- a picture/graphic
- a list of information

Save the file as **xyhtml_basics.html** where xy is your initials. Email the file to jmorales@austincc.edu.