

PROGRAMMING THE WEB**Subject Code: 10CS73****Hours/Week : 04****Total Hours : 52****I.A. Marks : 25****Exam Hours: 03****Exam Marks: 100****PART - A****UNIT – 1****6 Hours**

Fundamentals of Web, XHTML – 1: Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security, The Web Programmers Toolbox.

XHTML: Basic syntax, Standard structure, Basic text markup, Images, Hypertext Links.

UNIT – 2**7 Hours**

XHTML – 2, CSS: XHTML (continued): Lists, Tables, Forms, Frames CSS: Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The box model, Background images, The and <div> tags, Conflict resolution.

UNIT – 3**6 Hours**

Javascript: Overview of Javascript, Object orientation and Javascript, Syntactic characteristics, Primitives, operations, and expressions, Screen output and keyboard input, Control statements, Object creation and modification, Arrays, Functions, Constructors, Pattern matching using regular expressions, Errors in scripts, Examples.

UNIT – 4**7 Hours**

Javascript and HTML Documents, Dynamic Documents with Javascript: The Javascript execution environment, The Document Object Model, Element access in Javascript, Events and event handling, Handling events from the Body elements, Button elements, Text box and Password elements, The DOM 2 event model, The navigator object, DOM tree traversal and modification. Introduction to dynamic documents, Positioning elements, Moving elements, Element visibility, Changing colors and fonts, Dynamic content, Stacking elements, Locating the mouse cursor, Reacting to a mouse click, Slow movement of elements, Dragging and dropping elements.

PART - B**UNIT – 5****6 Hours**

XML: Introduction, Syntax, Document structure, Document type definitions, Namespaces, XML schemas, Displaying raw XML documents, Displaying XML documents with CSS, XSLT style sheets, XML processors, Web services.

UNIT – 6**7 Hours**

Perl, CGI Programming: Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays, Hashes, References, Functions, Pattern matching, File input and output; Examples.

The Common Gateway Interface; CGI linkage; Query string format; CGI.pm module; A survey example; Cookies. Database access with Perl and MySQL

UNIT – 7**6 Hours**

PHP: Origins and uses of PHP, Overview of PHP, General syntactic characteristics, Primitives, operations and expressions, Output, Control statements, Arrays, Functions, Pattern matching, Form handling, Files, Cookies, Session tracking, Database access with PHP and MySQL.

UNIT – 8**7 Hours**

Ruby, Rails: Origins and uses of Ruby, Scalar types and their operations, Simple input and output, Control statements, Arrays, Hashes, Methods, Classes, Code blocks and iterators, Pattern matching.

Overview of Rails, Document requests, Processing forms, Rails applications with Databases, Layouts.

Text Books:

1. Robert W. Sebesta: Programming the World Wide Web, 4 Edition, Pearson Education, 2008. (Listed topics only from Chapters 1 to 9, 11 to 15)

Reference Books:

1. M. Deitel, P.J. Deitel, A. B. Goldberg: Internet & World Wide Web How to Program, 4th Edition, Pearson Education, 2004.

2. Chris Bates: Web Programming Building Internet Applications, 3rd Edition, Wiley India, 2007.

3. Xue Bai et al: The web Warrior Guide to Web Programming, Cengage Learning, 2003.

INDEX SHEET

PART A :		Page no.
UNIT 1:	FUNDAMENTALS OF WEB, XHTML – 1:	4 - 21
UNIT 2:	XHTML – 2, CSS: XHTML (CONTINUED):	22 – 43
UNIT 3:	JAVASCRIPT:	44 – 62
UNIT 4:	JAVASCRIPT AND HTML DOCUMENTS,	63 – 91
PART – B		Page no.
UNIT 5:	XML:	92 – 110
UNIT 6:	PERL, CGI PROGRAMMING:	111 – 156
UNIT 7:	PHP:	
UNIT 8:	RUBY, RAILS:	

UNIT - 1

Syllabus: UNIT - 1 Fundamentals of Web, XHTML – 1

Internet, WWW, Web Browsers and Web Servers; URLs; MIME;
HTTP; Security; The Web Programmers Toolbox.
XHTML: Origins and evolution of HTML and XHTML
Basic syntax
Standard XHTML document structure;
Basic text markup. Images, Hypertext Links.

Unit1 Fundamentals

1.1 A Brief Intro to the Internet

- Internet History
- Internet Protocols

Internet History

1.1.1 Origins

In the 1960s the U.S Department of Defense (DoD) became interested in developing a new large-scale computer network.

The purposes of this network were communications, program sharing and remote computer access. One fundamental requirement was that the network be sufficiently Robust so that even if some network nodes were lost due to damage or some more reason the network could continue to function.

The DoD's Advanced Research Projects Agency (ARPA) funded the construction of the first such network, and the network the first such network, and the network was named as ARPAnet in 1969.

The primary use of ARPAnet was simple text-based communications through e-mail.

A number of other networks were developed during the late 1970's and early 1980's with BITNET and CSNET among them.

BITNET, which is an acronym for Because It's Time Network, developed at City University of NewYork. It was built initially to provide electronic mail and file transfers CSNET, which is an acronym for Computer Science Network, connected the university of Delaware, Purdue

University, RAND corporation and many more universities with initial purpose was to provide Electronic mail.

For the variety of reasons, neither BITNET not CSNET became a dominant national network.

A new national network, NSFnet was created in 1986. It was funded by National Science Foundation (NSF). NSFnet initially connected NSF supercomputer centers.

By 1990, NSFnet had replaced ARPAnet for most nonmilitary uses. By 1992 NSFnet connected more than 1 million computers around the world.

In 1995 a small part of NSFnet returned to being a research network. The rest is known as the Internet.

As a Summary:

- ARPAnet - late 1960s and early 1970s
 - Network reliability
 - For ARPA-funded research organizations
- BITnet, CSnet - late 1970s & early 1980s
 - email and file transfer for other institutions
- NSFnet - 1986
 - Originally for non-DOD funded places
 - Initially connected five supercomputer centers
 - By 1990, it had replaced ARPAnet for non-military uses
 - Soon became the network for all (by the early 1990s)
- NSFnet eventually became known as the Internet

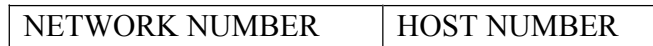
1.1.2 What the Internet is:

- Internet is a huge collection of computers connected in a communications network.
- It is a network of network rather than a network of computers.
- Using Internet many people can share resources and can communicate with each other.
- To have Internet service your computer must be connected to the Internet Service Providers (ISP) through cables modem, phone-line modem or DSL.
- The Internet employs a set of standardized protocols which allow for the sharing of resources. These standars are known by the Internet Protocol Suite.
- At the lowest level, since 1982, all connections use TCP/IP

1.1.3 Internet Protocols (IP) Addresses

- Internet Protocol (IP) Addresses

- Every node has a unique numeric address
- Form: 32-bit binary number
- IP address is divided into 2 main part:
 - Network number and
 - Host number
- IP addresses usually are written as four 8-bit numbers separated by dots



- Organizations are assigned groups of IPs for their computers
- There are 5 classes of IP address

Sl.no.	IP address Class	Format	Purpose
1.	Class A	N.H.H.H	Few large organization use this class addressing
2.	Class B	N.N.H.H	Medium size organizations use this addressing
3.	Class C	N.N.N.H	Relatively small organizations use this class

Here N stands for Network number and H stands for Host number. For example, a small organization may be assigned 256 IP addresses, such as 191.28.121.0 to 191.28.121.255

- Problem: By the mid-1980s, several different protocols had been invented and were being used on the Internet, all with different user interfaces (Telnet, FTP, Usenet, mailto)

1.1.4 Domain names

- Form: host-name.domain-names
- First domain is the smallest; last is the largest
- Last domain specifies the type of organization
- Fully qualified domain name - the host name and all of the domain names
- DNS servers - convert fully qualified domain names to IPs
- Few domains are:
 - Edu –Extension for Educational institutions
 - Com – Specifies a Company
 - Gov – Specifies government
 - Org – Other kind of organization
- Even Domain specifies the country name
 - in – India
 - pk – Pakistan

- au – Australia
- us – United states

Domain Name Conversion

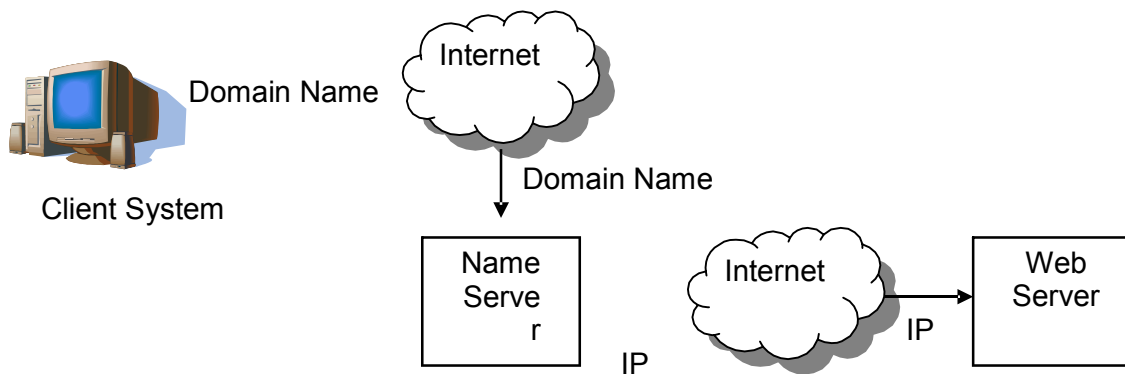


Fig. Domain Name Conversion

IP addresses are the address used internally by the Internet, the fully qualified domain name of the destination for a message, which is given browser, must be converted to an IP address before the message can be transmitted on the internet to the destination. These conversions are done by system software called Name Servers.

Name Servers server a collection of machines on the Internet and are operated by organizations that are responsible for the part of the Internet to which those machines are connected.

All documents requested from the browsers are routed to the nearest name server. If the name server can convert the fully qualified domain name to an IP address. If it cannot, the name server sends the fully qualified domain name to another name server for conversion.

The figure 1 shows how fully qualified domain names requested by a browser are translated into IPs before they are routed to the appropriate web server.

One way to determine the IP address of the website by using telnet.

If we want to know the IP address of [www. Google.co.in](http://www.google.co.in), go to Dos prompt and type telnet www.google.co.in
PROTOCOLS

By the mid – 198s, a collection of different protocols that run on top of TCP/IP had been developed to support a variety of Internet users. Among those the most common were telnet, ftp, usenet, mailto

Uses:

- telnet – which was developed to allow a user on one computer on the Internet to log on to and use another computer on the Internet.[Remote Login]
- ftp[file transfer protocol] - which was developed to transfer file among computers on the Internet.
- usenet – Which was developed to serve as an electronic bulletin board.
- mailto – which was developed to allow messages to be sent from the user of one computer on the Internet to other users on other computer on the Internet.

Client and Server

- Clients and Servers are programs that communicate with each other over the Internet
- A Server runs continuously, waiting to be contacted by a Client
 - Each Server provides certain services
 - Services include providing web pages
- A Client will send a message to a Server requesting the service provided by that server
 - The client will usually provide some information, parameters, with the request

1.2 The World-Wide Web

- A possible solution to the proliferation of different protocols being used on the Internet

1.2.1 Origins

- Tim Berners-Lee at CERN proposed the Web in 1989
 - Purpose: to allow scientists to have access to many databases of scientific work through their own computers
- Document form: hypertext
- Pages? Documents? Resources?
 - We'll call them documents
- Hypermedia – more than just text – images, sound, etc.

1.2.2 Web or Internet?

- The Web uses one of the protocols, http, that runs on the Internet--there are several others (telnet, mailto, etc.)
- The Internet is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both

connected to the Internet. Information that travels over the Internet does so via a variety of languages known as protocols.

- The World Wide Web, or simply Web, is a way of accessing information over the medium of the Internet. The Web uses the HTTP protocol. The Web also utilizes browsers, such as Internet Explorer or Firefox, to access Web documents called Web pages that are linked to each other via hyperlinks. Web documents also contain graphics, sounds, text and video.
- The Internet is the large container, and the Web is a part within the container.
- But to be technically precise, the Net is the restaurant, and the Web is the most popular dish on the menu.
- Browsers are used to connect to the www part of the internet.

Here is a conceptual diagram of the Internet and how it contains many forms of online communications



The Internet and the Web work together, but they are not the same thing. The Internet provides the underlying structure, and the Web utilizes that structure to offer content, documents, multimedia, etc.

The Internet is at its most basic definition an electronic communications network. It is the structure on which the World Wide Web is based.

1.3 Web Browsers

- Browsers are clients - always initiate, servers react (although sometimes servers require responses)
- Mosaic - NCSA (Univ. of Illinois), in early 1993
 - First to use a GUI, led to explosion of Web use
 - Initially for X-Windows, under UNIX, but was ported to other platforms by late 1993
- Most requests are for existing documents, using HyperText Transfer Protocol (HTTP)
 - But some requests are for program execution, with the output being returned as a document

1.4 Web Servers

- Provide responses to browser requests, either existing documents or dynamically built documents
- All communications between browsers and servers use Hypertext Transfer Protocol (HTTP)
- Apache, Microsoft internet information server (IIS)

1.4.1 Web Server Operation

- Web servers run as background processes in the operating system
 - Monitor a communications port on the host, accepting HTTP messages when they appear
- All current Web servers came from either
 1. The original from CERN
 2. The second one, from NCSA

1.4.2 Web Server Operation Details

- Web servers have two main directories:
 1. Document root (servable documents)
 2. Server root (server system software)
- Document root is accessed indirectly by clients
 - Its actual location is set by the server configuration file
 - Requests are mapped to the actual location
 - Path /admin/web/topdocs/xyz.html
- Server root – stores server and its support software
- Virtual document trees : many servers allow part of the servable document collection to be stored outside the directory of document root. The secondary areas from which document can be served are called virtual document trees.
- Proxy servers : Some servers can serve documents that are in the document root of other machines on the web and those servers are called proxy servers.

Difference between apache and IIS

Apache Web Server	IIS web Server
1. It is an open source product. 2. Apache web server is useful on both UNIX based systems and on windows platform. 3. The apache web server can be	1. It is a vendorspecific product and can be used on windows only. 2. IIS web server is used on windows platform. 3. The IIS server can be controlled by modifying the window based management programs called IIS span-in. We can access 115 span-in by

controlled by editing the configuration file http.conf	going to control panel → Administrative tools → IIS admin
--	---

1.5 URLs (uniform resource locators)

1.5.1 General form:

scheme: object-address

- The scheme is often a communications protocol, such as telnet or ftp
- For the http protocol, the object-address is: fully qualified domain name/doc path
- For the file protocol, only the doc path is needed
- Host name may include a port number
- URLs cannot include spaces or any of a collection of other special characters (semicolons, colons, ...)
- The doc path may be abbreviated as a partial path
 - The rest is furnished by the server configuration
- If the doc path ends with a slash, it means it is a directory

1.6 Multipurpose Internet Mail Extensions (MIME)

- Originally developed for email
- Used to specify the form of a file returned by the server
- Type specifications
 - Form:

type/subtype

- Examples: text/plain, text/html, image/gif, image/jpeg, video/mpeg, video/rm, video/quicktime

Browser gets the type explicitly from the server

1.7 The HyperText Transfer Protocol

- The protocol used by ALL Web communications
- Current version of HTTP is 1.1
- Consists of 2 phases
 - request phase
 - response phase

http communication[request or response] between a browser and a web server consists of 2 parts → header-consists information about communication and

→ body – consists data of communication

1.7.1 Request Phase

☐ Form:

1. HTTP method domain part of URL HTTP ver.
2. Header fields
3. blank line semantics
4. Message body

1.7 The HyperText Transfer Protocol: Methods

- GET - Fetch a document
- POST - Execute the document, using the data in body
- HEAD - Fetch just the header of the document
- PUT - Store a new document on the server
- DELETE - Remove a document from the server

☐ An example of the first line of a request:

GET /degrees.html HTTP/1.1

■ Format of second line header field (optional)

☐ Field name followed by a colon and the value of the field.

HTTP Headers

■ Four categories of header fields:

General, request, response, & entity

■ Common request fields:

Accept: text/plain

Accept: text/*

If-Modified_since: date

■ Common response fields:

Content-length: 488

Content-type: text/html

- Can communicate with HTTP without a browser

➤ telnet blanca.uccs.edu http

➤ Creates connection to http port on server

http command eg:

GET /respond.html HTTP/1.1

Host: blanca.uccs.edu

1.7.2 Response phase

■ Form:

1. Status line
2. Response header fields
3. blank line
4. Response body

■ Status line format:

HTTP version status code explanation

■ Example: HTTP/1.1 200 OK

(Current version is 1.1)

- Status code is a three-digit number; first digit specifies the general status

1 => Informational

2 => Success

3 => Redirection

4 => Client error

5 => Server error

- The header field, Content-type, is required

HTTP Response Example

HTTP/1.1 200 OK

Date: Tues, 18 May 2004 16:45:13 GMT

Server: Apache (Red-Hat/Linux)

Last-modified: Tues, 18 May 2004 16:38:38 GMT

Etag: "841fb-4b-3d1a0179"

Accept-ranges: bytes

Content-length: 364

Connection: close

Content-type: text/html, charset=ISO-8859-1

- Both request headers and response headers must be followed by a blank line

1.8 Note on security?

1.9 The Web Programmer's Toolbox

- Document languages and programming languages that are the building blocks of the web and web programming
- XHTML
- Plug-ins
- Filters
- XML
- Javascript
- Java, Perl, Ruby, PHP

1.9.1 XHTML

- To describe the general form and layout of documents
- An XHTML document is a mix of content and controls
 - Controls are tags and their attributes
 - Tags often delimit content and specify something about how the content should be arranged in the document
 - Attributes provide additional information about the content of a tag

1.9.2 Creating XHTML documents

- XHTML editors - make document creation easier
 - Shortcuts to typing tag names, spell-checker,
- WYSIWYG XHTML editors
 - Need not know XHTML to create XHTML documents

1.9.3 Plugins and Filters

- Plug ins
 - Integrated into tools like word processors, effectively converting them to WYSIWYG XHTML editors
- Filters
 - Convert documents in other formats to XHTML

Plugins and Filters: Advantages and Disadvantages

- Advantages of both filters and plug-ins:
 - Existing documents produced with other tools can be converted to XHTML documents
 - Use a tool you already know to produce XHTML
- Disadvantages of both filters and plug-ins:
 - XHTML output of both is not perfect - must be fine tuned
XHTML may be non-standard
 - You have two versions of the document, which are difficult to synchronize

- A meta-markup language

1.9 XML

- Designed to create a new markup language for a particular purpose or area
- Because the tags are designed for a specific area, they can be meaningful
- No presentation details
- A simple and universal way of representing data of any textual kind

1.9.5 JavaScript

- A client-side HTML-embedded scripting language
- Only related to Java through syntax
- Dynamically typed and not object-oriented
- Provides a way to access elements of HTML documents and dynamically change them

1.9.6 Java

- General purpose object-oriented programming language
- Based on C++, but simpler and safer
- Our focus is on applets, servlets, and JSP

1.9.7 Perl

- Provides server-side computation for HTML documents, through CGI
- Perl is good for CGI programming because:
 - Direct access to operating systems functions
 - Powerful character string pattern-matching operations
 - Access to database systems
- Perl is highly platform independent, and has been ported to all common platforms
- Perl is not just for CGI

1.9.8 PHP

- A server-side scripting language

- An alternative to CGI

- Similar to JavaScript
- Great for form processing and database access through the Web

1.10 Origins and Evolution of HTML

- HTML was defined with SGML
- Original intent of HTML: General layout of documents that could be displayed by a wide variety of computers
- Recent versions:
 - HTML 3.2 – 1997
 - Introduced many new features and deprecated many older features
 - HTML 4.01 - 1999 - A cleanup of 4.0
 - XHTML 1.0 - 2000
 - Just 4.01 defined using XML, instead of SGML
 - XHTML 1.1 – 2001
 - Modularized 1.0, and drops frames
 - We'll stick to 1.1, except for frames
- **Reasons to use XHTML, rather than HTML:**
 1. HTML has lax syntax rules, leading to sloppy and sometime ambiguous documents
 - XHTML syntax is much more strict, leading to clean and clear documents in a standard form
 2. HTML processors do not even enforce the few syntax rule that do exist in HTML
 3. The syntactic correctness of XHTML documents can be validated

1.11 Basic Syntax

- Elements are defined by tags (markers)
 - Tag format:
 - Opening tag: <name>
 - Closing tag: </name>
 - The opening tag and its closing tag together specify a container for the *content* they enclose
- Not all tags have content
 - If a tag has no content, its form is <name />
- The container and its content together are called an *element*
- If a tag has attributes, they appear between its name and the right bracket of the opening tag
- Comment form: <!-- ... -->
- Browsers ignore comments, unrecognizable tags, line breaks, multiple spaces, and tabs
- Tags are suggestions to the browser, even if they are recognized by the browser

1.12 HTML Document Structure

- <html>, <head>, <title>, and <body> are required in every document
- Every XHTML document must begin with:

```
<?xml version = "1.0"?>
```

```
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
```

```
http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>
```

- The whole document must have <html> as its root
- html must have the xmlns attribute:

```
<html xmlns = "http://www.w3.org/1999/xhtml"
```

- A document consists of a head and a body
- The <title> tag is used to give the document a title, which is normally displayed in the browser's window title bar (at the top of the display)
- Prior to XHTML 1.1, a document could have either a body or a frameset

1.13 Basic Text Markup

- Text is normally placed in paragraph elements
- *Paragraph Elements*

- The <p> tag breaks the current line and inserts a blank line - the new line gets the beginning of the content of the paragraph
- The browser puts as many words of the paragraph's content as will fit in each line

```
<?xml version = "1.0"?>
```

```
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
```

```
http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>
```

```
<!-- greet.html
```

```
A trivial document
```

```
-->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head> <title> Our first document </title>
```

```
</head>
```

```
<body>
```

```
<p>
```

```
Greetings from your Webmaster!
```

```
</p>
```

```
</body>
```

```
</html>
```

- W3C HTML Validation Service
<http://validator.w3.org/file-upload.html>
- Line breaks

- The effect of the
 tag is the same as that of <p>, except for the blank line

- No closing tag!

- Example of paragraphs and line breaks

On the plains of hesitation <p> bleach the
bones of countless millions </p>

who, at the dawn of victory
 sat down
to wait, and waiting, died.

- Typical display of this text:

On the plains of hesitation

bleach the bones of countless millions
who, at the dawn of victory
sat down to wait, and waiting, died.

- *Headings*

- Six sizes, 1 - 6, specified with <h1> to <h6>
- 1, 2, and 3 use font sizes that are larger than the default font size
- 4 uses the default size
- 5 and 6 use smaller font sizes

```
<!-- headings.html
  An example to illustrate headings
  -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Headings </title>
  </head>
  <body>
    <h1> Aidan's Airplanes (h1) </h1>
    <h2> The best in used airplanes (h2) </h2>
    <h3> "We've got them by the hangarful" (h3)
    </h3>
    <h4> We're the guys to see for a good used
      airplane (h4) </h4>
    <h5> We offer great prices on great planes
      (h5) </h5>
    <h6> No returns, no guarantees, no refunds,
      all sales are final (h6) </h6>
  </body>
</html>
```



- **Blockquotes**
 - Content of `<blockquote>`
 - To set a block of text off from the normal flow and appearance of text
 - Browsers often indent, and sometimes italicize
- **Font Styles and Sizes (can be nested)**
 - Boldface - ``
 - Italics - `<i>`
 - Larger - `<big>`
 - Smaller - `<small>`
 - Monospace - `<tt>`

The `<big>` sleet `<big>` in `<big>` `<i>` Crete

`</i>``
` lies `</big>` completely `</big>`

in `</big>` the street

The sleet in *Crete*

lies completely in the street

- These tags are not affected if they appear in the content of a `<blockquote>`, unless there is a conflict (e.g., italics)

- Superscripts and subscripts*

- Subscripts with `<sub>`

- Superscripts with `<sup>`

Example: `x₂³`

Display: x₂³

- Inline versus block elements
- All of this font size and font style stuff can be done with style sheets, but these tags are not yet deprecated
- Character Entities

Char. Entity Meaning

`&` `&` Ampersand

<	<	Less than
>	>	Greater than
”	"	Double quote
'	'	Single quote
¼	¼	One quarter
½	½	One half
¾	¾	Three quarters
°	°	Degree
(space) 	 	Non-breaking space

- Horizontal rules

- `<hr />` draws a line across the display, after a line break

- The meta element (for search engines) Used to provide additional information about a document, with attributes, not content

Images

- GIF (Graphic Interchange Format)

- 8-bit color (256 different colors)

- JPEG (Joint Photographic Experts Group)

- 24-bit color (16 million different colors)

- Both use compression, but JPEG compression is better

- Images are inserted into a document with the `` tag with the `src` attribute

- The `alt` attribute is required by XHTML

- Purposes:

1. Non-graphical browsers
2. Browsers with images turned off

```
<img src = "comets.jpg"
```

```
alt = "Picture of comets" />
```

- The `` tag has 30 different attributes, including width and height (in pixels)

- Portable Network Graphics (PNG)

- Relatively new
 - Should eventually replace both gif and jpeg

Eg:

```
<!-- image.html
```

```
  An example to illustrate an image
```

```
-->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
  <head> <title> Images </title>
```

```
  </head>
```

```
  <body>
```

```
<h1> Aidan's Airplanes </h1>
<h2> The best in used airplanes </h2>
<h3> "We've got them by the hangarful"
</h3>
<h2> Special of the month </h2>
<p>
  1960 Cessna 210 <br />
  577 hours since major engine overhaul
  <br />
  1022 hours since prop overhaul
  <br /><br />
  <img src = "c210new.jpg"
    alt = "Picture of a Cessna 210"/>
  <br />
  Buy this fine airplane today at a
  remarkably low price <br />
  Call 999-555-1111 today!
</p>
</body>
</html>
```



2.2 Hypertext Links

- **Hypertext is the essence of the Web!**
- **A link is specified with the href (*hypertext reference*) attribute of <a> (the anchor tag)**
 - **The content of <a> is the visual link in the document**
 - **If the target is a whole document (not the one in which the link appears), the target need not be specified in the target document as being the target**
- **Note: Relative addressing of targets is easier to maintain and more portable than absolute addressing**

<!-- link.html

An example to illustrate a link

-->

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head> <title> Links </title>
```

```
</head>
```

```
<body>
```

```
<h1> Aidan's Airplanes </h1>
```

```
<h2> The best in used airplanes </h2>
```

```
<h3> "We've got them by the hangarful"
```

```
</h3>
```

```
<h2> Special of the month </h2>
```

```
<p>
```

```
1960 Cessna 210 <br />
```

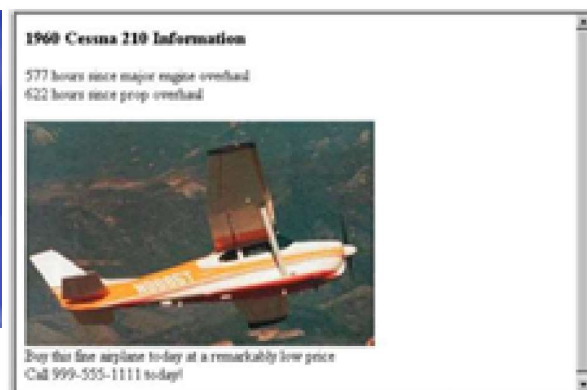
```
<a href = "C210data.html">
```

```
Information on the Cessna 210 </a>
```

```
</p>
```

```
</body>
```

```
</html>
```



- If the target is not at the beginning of the document, the target spot must be marked
- Target labels can be defined in many different tags with the id attribute, as in `<h1 id = "baskets"> Baskets </h1>`
 - The link to an id must be preceded by a pound sign (#); If the id is in the same document, this target could be ``
 **What about baskets? **
 - If the target is in a different document, the document reference must be included ` Baskets `
 - Style note: a link should blend in with the surrounding text, so reading it without taking the link should not be made less pleasant
 - Links can have images:
``
`<img src = "smallplane.jpg"`
`alt = "Small picture of an airplane " />`
 **Info on C210 **

UNIT – 2: XHTML – 2: CSS: XHTML (continued..)

Lists; Tables;
Forms; Frames;
CSS: Introduction; Levels of style sheets; Property value forms;
Style specification formats; Selector forms;
Font properties; List properties;
Color; Alignment of text;
The Box model; Background images; The and tags; Conflict resolution.

2.1 Lists

- *Unordered lists*
- The list is the content of the tag
- List elements are the content of the tag

```
<h3> Some Common Single-Engine Aircraft </h3>
```

```
<ul>
```

```
<li> Cessna Skyhawk </li>
```

```
<li> Beechcraft Bonanza </li>
```

```
<li> Piper Cherokee </li>
```

```
</ul>
```



- *Ordered lists*
 - The list is the content of the tag
 - Each item in the display is preceded by a sequence value

```
<h3> Cessna 210 Engine Starting Instructions </h3>
```

```
</h3>
```

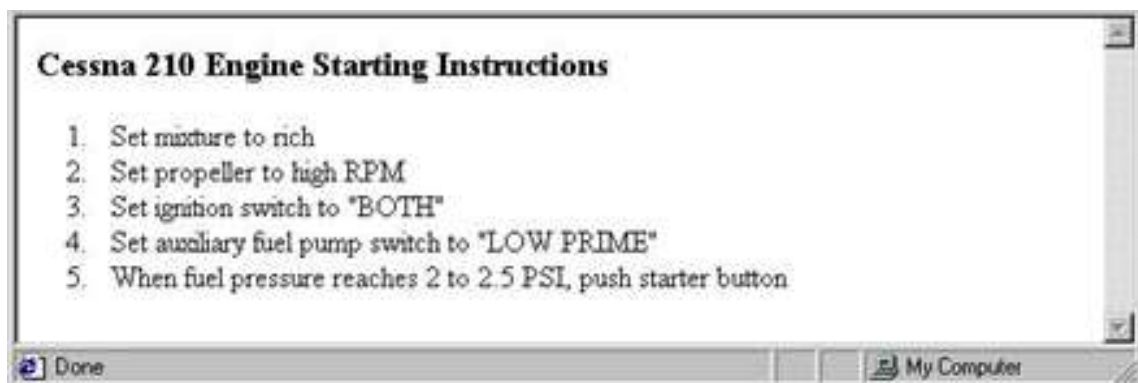
```
<ol>
```

```
<li> Set mixture to rich </li>
```

```

<li> Set propeller to high RPM </li>
<li> Set ignition switch to "BOTH" </li>
<li> Set auxiliary fuel pump switch to
  "LOW PRIME" </li>
<li> When fuel pressure reaches 2 to 2.5
  PSI, push starter button </li>
</ol>

```



■ *Definition lists (for glossaries, etc.)*

- List is the content of the <dl> tag
- Terms being defined are the content of the <dt> tag
- The definitions themselves are the content of the <dd> tag

```
<h3> Single-Engine Cessna Airplanes </h3>
```

```
<dl >
```

```
<dt> 152 </dt>
```

```
<dd> Two-place trainer </dd>
```

```
<dt> 172 </dt>
```

```
<dd> Smaller four-place airplane </dd>
```

```
<dt> 182 </dt>
```

```
<dd> Larger four-place airplane </dd>
```

```
<dt> 210 </dt>
```

```
<dd> Six-place airplane - high performance
```

```
</dd>
```

```
</dl>
```


The screenshot shows a web browser window with the title "Single-Engine Cessna Airplanes". The browser's status bar at the bottom indicates "Done" and "My Computer". The table content is as follows:

Model	Description
152	Two-place trainer
172	Smaller four-place airplane
182	Larger four-place airplane
210	Six-place airplane - high performance

2.2 Tables

- A table is a matrix of cells, each possibly having content
- The cells can include almost any element
- Some cells have row or column labels and some have data
- A table is specified as the content of a `<table>` tag
- A border attribute in the `<table>` tag specifies a border between the cells
- If border is set to "border", the browser's default width border is used
- The border attribute can be set to a number, which will be the border width
- Without the border attribute, the table will have no lines!
- Tables are given titles with the `<caption>` tag, which can immediately follow `<table>`
- Each row of a table is specified as the content of a `<tr>` tag
 - The row headings are specified as the content of a `<th>` tag
 - The contents of a data cell is specified as the content of a `<td>` tag

```
<table border = "border">
  <caption> Fruit Juice Drinks </caption>
  <tr>
    <th> </th>
    <th> Apple </th>
    <th> Orange </th>
    <th> Screwdriver </th>
  </tr>
  <tr>
    <th> Breakfast </th>
    <td> 0 </td>
```

```

<td> 1 </td>
<td> 0 </td>
</tr>
<tr>
  <th> Lunch </th>
  <td> 1 </td>
  <td> 0 </td>
  <td> 0 </td>
</tr>
</table>

```

	Apple	Orange	Screwdriver
Breakfast	0	1	0
Lunch	1	0	0
Dinner	0	0	1

- A table can have two levels of column labels
 - If so, the colspan attribute must be set in the <th> tag to specify that the label must span some number of columns

```

<tr>
  <th colspan = "3"> Fruit Juice Drinks </th>
</tr>
<tr>
  <th> Orange </th>
  <th> Apple </th>
  <th> Screwdriver </th>
</tr>

```

Fruit Juice Drinks		
Orange	Apple	Screwdriver

- If the rows have labels and there is a spanning column label, the upper left corner must be made larger, using rowspan

```

<table border = "border">
  <tr>

```

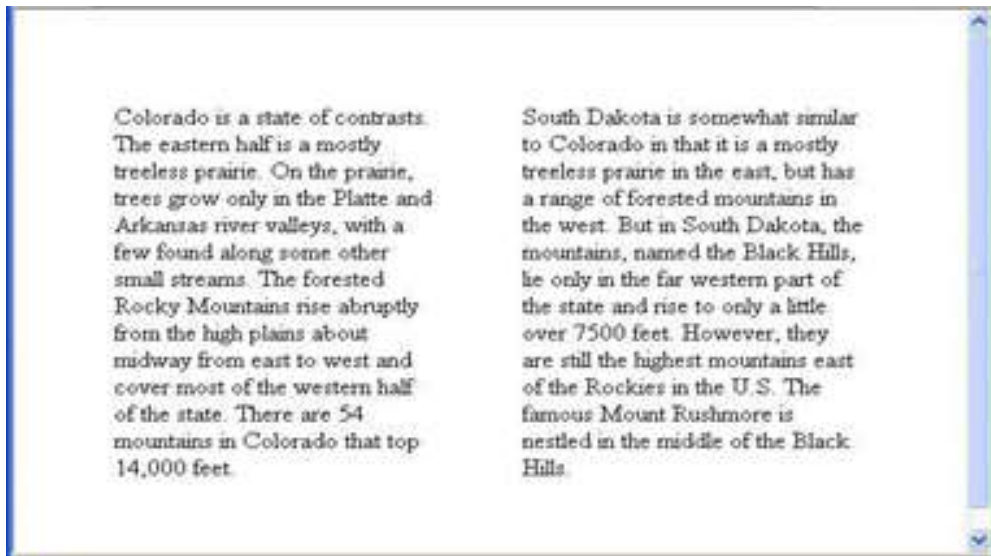
```
<td rowspan = "2"> </td>
<th colspan = "3"> Fruit Juice Drinks
</th>
</tr>
<tr>
<th> Apple </th>
<th> Orange </th>
<th> Screwdriver </th>
</tr>
...
</table>
```

- The align attribute controls the horizontal placement of the contents in a table cell
 - Values are left, right, and center (default)
 - align is an attribute of <tr>, <th>, and <td> elements
- The valign attribute controls the vertical placement of the contents of a table cell
 - Values are top, bottom, and center (default)
 - valign is an attribute of <th> and <td> elements

→ SHOW cell_align.html and display it

- The cellspacing attribute of <table> is used to specify the distance between cells in a table
- The cellpadding attribute of <table> is used to specify the spacing between the content of a cell and the inner walls of the cell

```
<table cellspacing = "50">
<tr>
<td> Colorado is a state of ...
</td>
<td> South Dakota is somewhat...
</td>
</tr>
</table>
```



- *Table Sections*
 - Header, body, and footer, which are the elements: thead, tbody, and tfoot
 - If a document has multiple tbody elements, they are separated by thicker horizontal lines

2.3 Forms

- A form is the usual way information is gotten from a browser to a server
- HTML has tags to create a collection of objects that implement this information gathering
 - The objects are called *widgets* (e.g., radio buttons and checkboxes)
- When the Submit button of a form is clicked, the form's values are sent to the server
- All of the widgets, or components of a form are defined in the content of a <form> tag
 - The only required attribute of <form> is action, which specifies the URL of the application that is to be called when the Submit button is clicked

action =

"http://www.cs.ucp.edu/cgi-bin/survey.pl"

- If the form has no action, the value of action is the empty string
- The method attribute of <form> specifies one of the two possible techniques of transferring the form data to the server, get and post
 - get and post are discussed in Chapter 10
- *Widgets*
 - Many are created with the <input> tag
 - The type attribute of <input> specifies the kind of widget being created
 - Text

- Creates a horizontal box for text input
- Default size is 20; it can be changed with the size attribute
- If more characters are entered than will fit, the box is scrolled (shifted) left
- If you don't want to allow the user to type more characters than will fit, set maxlength, which causes excess input to be ignored

```
<input type = "text" name = "Phone"
size = "12" >
```

2. Checkboxes - to collect multiple choice input

- Every checkbox requires a value attribute, which is the widget's value in the form data when the checkbox is 'checked'
 - A checkbox that is not 'checked' contributes no value to the form data
- By default, no checkbox is initially 'checked'
- To initialize a checkbox to 'checked', the checked attribute must be set to "checked"
- *Widgets* (continued)

Grocery Checklist

```
<form action = "">
<p>
<input type = "checkbox" name ="groceries"
value = "milk" checked = "checked">
Milk
<input type = "checkbox" name ="groceries"
value = "bread">
Bread
<input type = "checkbox" name = "groceries"
value= "eggs">
Eggs
</p>
</form>
```



3. Radio Buttons - collections of checkboxes in which only one button can be 'checked' at a time

- Every button in a radio button group **MUST** have the same name

- *Widgets* (continued)

3. Radio Buttons (continued)

- If no button in a radio button group is ‘pressed’, the browser often ‘presses’ the first one

Age Category

```
<form action = "">
<p>
<input type = "radio" name = "age"
value = "under20" checked = "checked"> 0-19
<input type = "radio" name = "age"
value = "20-35"> 20-35
<input type = "radio" name = "age"
value = "36-50"> 36-50
<input type = "radio" name = "age"
value = "over50"> Over 50
</p>
</form>
```



4. Menus - created with <select> tags

- There are two kinds of menus, those that behave like checkboxes and those that behave like radio buttons (the default)
 - Menus that behave like checkboxes are specified by including the multiple attribute, which must be set to "multiple"
- The name attribute of <select> is required
- The size attribute of <select> can be included to specify the number of menu items to be displayed (the default is 1)
 - If size is set to > 1 or if multiple is specified, the menu is displayed as a pop-up menu

Menus (continued)

- Each item of a menu is specified with an <option> tag, whose pure text content (no tags) is the value of the item
- An <option> tag can include the selected attribute, which when assigned "selected" specifies that the item is preselected

Grocery Menu - milk, bread, eggs, cheese

```
<form action = "">
<p>
```

```

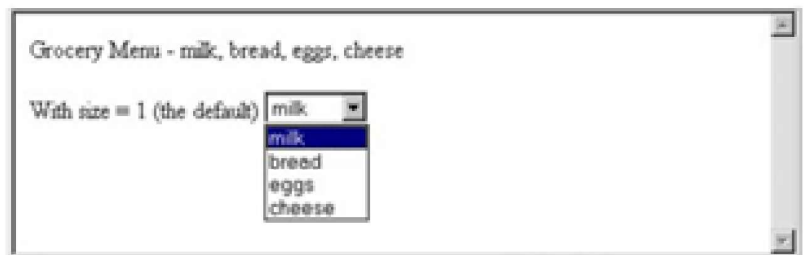
With size = 1 (the default)
<select name = "groceries">
  <option> milk </option>
  <option> bread </option>
  <option> eggs </option>
  <option> cheese </option>
</select>
</p>
</form>

```

- **Widgets (continued)**



■ **After clicking the menu:**



■ **After changing size to 2:**



- **Widgets (continued)**

5. Text areas - created with <textarea>

- ❑ Usually include the rows and cols attributes to specify the size of the text area
- ❑ Default text can be included as the content of <textarea>
- ❑ Scrolling is implicit if the area is overfilled

Please provide your employment aspirations

```
<form action = "">
```

```

<p>
<textarea name = "aspirations" rows = "3"
      cols = "40">
(Be brief and concise)
</textarea>
</p>
</form>

```



■ *Widgets* (continued)

6. Reset and Submit buttons

- Both are created with `<input>`

```
<input type = "reset" value = "Reset Form">
```

```
<input type = "submit" value = "Submit Form">
```

■ Submit has two actions:

1. Encode the data of the form
2. Request that the server execute the server-resident program specified as the value of the action attribute of `<form>`
3. A Submit button is required in every form

--> SHOW popcorn.html and display it

2.4 Frames

- Frames are rectangular sections of the display window, each of which can display a different document
- Because frames are no longer part of XHTML, you cannot validate a document that includes frames
- The `<frameset>` tag specifies the number of frames and their layout in the window
 - `<frameset>` takes the place of `<body>`
 - Cannot have both!
 - `<frameset>` must have either a rows attribute or a cols attribute, or both (usually the case)
 - Default is 1
 - The possible values for rows and cols are numbers, percentages, and asterisks
 - A number value specifies the row height in pixels - Not terribly useful!

- A percentage specifies the percentage of total window height for the row - Very useful!
 - ❑ An asterisk after some other specification gives the remainder of the height of the window
 - ❑ Examples:

```
<frameset rows = "150, 200, 300">
```

```
<frameset rows = "25%, 50%, 25%">
```

```
<frameset rows = "50%, 20%, *" >
```

```
<frameset rows = "50%, 25%, 25%"
  cols = "40%, *">
```

- The <frame> tag specifies the content of a frame
- The first <frame> tag in a <frameset> specifies the content of the first frame, etc.
 - ❑ Row-major order is used
 - ❑ Frame content is specified with the src attribute
 - ❑ Without a src attribute, the frame will be empty (such a frame CANNOT be filled later)
- If <frameset> has fewer <frame> tags than frames, the extra frames are empty
- Scrollbars are implicitly included if needed (they are needed if the specified document will not fit)
- If a name attribute is included, the content of the frame can be changed later (by selection of a link in some other frame)

→SHOW frames.html

- Note: the Frameset standard must be specified in the DOCTYPE declaration

Eg:

```
<!-- contents.html
```

```
  The contents of the first frame of
  frames.html, which is the table of
  contents for the second frame
```

```
-->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
  <head> <title> Table of Contents Frame
```

```
    </title>
```

```
  </head>
```

```
  <body>
```

```
    <h4> Fruits </h4>
```

```
    <ul>
```

```
      <li> <a href = "apples.html"
```

```
        target = "descriptions">
        apples </a>
<li> <a href = "bananas.html"
        target = "descriptions">
        bananas </a>
<li> <a href = "oranges.html"
        target = "descriptions">
        oranges </a>
</ul>
</body>
</html>
```

- Nested frames - to divide the screen in more interesting ways

→ SHOW nested_frames.html

2.5 Introduction

- The CSS1 specification was developed in 1996
- CSS2 was released in 1998
- CSS3 is on its way
- CSSs provide the means to control and change presentation of HTML documents
- CSS is not technically HTML, but can be embedded in HTML documents
- Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents
- Style is specified for a tag by the values of its properties

2.6 Levels of Style Sheets

- There are three levels of style sheets
 - Inline - specified for a specific occurrence of a tag and apply only to that tag
 - This is fine-grain style, which defeats the purpose of style sheets - uniform style
 - Document-level style sheets - apply to the whole document in which they appear
 - External style sheets - can be applied to any number of documents
- When more than one style sheet applies to a specific tag in a document, the lowest level style sheet has precedence
 - In a sense, the browser searches for a style property spec, starting with inline, until it finds one (or there isn't one)
- Inline style sheets appear in the tag itself

- Document-level style sheets appear in the head of the document
- External style sheets are in separate files, potentially on any server on the Internet
 - Written as text files with the MIME type text/css

2.7 Linking an External Stylesheet

- A <link> tag is used to specify that the browser is to fetch and use an external style sheet file

```
<link rel = "stylesheet" type = "text/css"
href = "http://www.wherever.org/termpaper.css">
</link>
```

- External style sheets can be validated

<http://jigsaw.w3.org/css-validator/validator-upload.html>

2.8 Style Specification Formats

- Format depends on the level of the style sheet
- Inline:
 - Style sheet appears as the value of the style attribute
 - General form:

```
style = "property_1: value_1;
property_2: value_2;
...
property_n: value_n"
```

2.9 Format for Document-level

- Style sheet appears as a list of rules that are the content of a <style> tag
- The <style> tag must include the type attribute, set to "text/css"
- The list of rules must be placed in an HTML comment, because it is **not** HTML
- Comments in the rule list must have a different form - use C comments (/*...*/)

2.10 General Form, Document Level

- General form:

```
<style type = "text/css">
<!--
rule list
-->
</style>
```

- Form of the rules:
 - selector {list of property/values}
 - Each property/value pair has the form:
 - property: value
 - Pairs are separated by semicolons, just as in the value of a <style> tag

General Form, External style sheets

- Form is a list of style rules, as in the content of a `<style>` tag for document-level style sheets

Selector Forms: Simple

- The selector is a tag name or a list of tag names, separated by commas
 - h1, h3
 - p
- *Contextual selectors*
 - ol ol li

Class Selectors

- Used to allow different occurrences of the same tag to use different style specifications
- A style class has a name, which is attached to a tag name
 - p.narrow {property/value list}
 - p.wide {property/value list}
- The class you want on a particular occurrence of a tag is specified with the class attribute of the tag
- For example,

```
<p class = "narrow">
```

```
...
```

```
</p>
```

```
...
```

```
<p class = "wide">
```

```
...
```

```
</p>
```

Generic Selectors

- A generic class can be defined if you want a style to apply to more than one kind of tag
- A generic class must be named, and the name must begin with a period

- Example,

```
.really-big { ... }
```

- Use it as if it were a normal style class

```
<h1 class = "really-big"> ... </h1>
```

```
...
```

```
<p class = "really-big"> ... </p>
```

id Selectors

- An id selector allow the application of a style to one specific element
- General form:

```
#specific-id {property-value list}
```

- Example:

```
#section14 {font-size: 20}
```

Pseudo Classes

- Pseudo classes are styles that apply when something happens, rather than because the target element simply exists
- Names begin with colons
- hover classes apply when the mouse cursor is over the element
- focus classes apply when an element has focus

Pseudo Class Example

```
<!-- pseudo.html -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Checkboxes </title>
    <style type = "text/css">
      input:hover {color: red;}
      input:focus {color: green;}
    </style>
  </head>
  <body>
    <form action = "">
      <p>
        Your name:
        <input type = "text" />
      </p>
    </form>
  </body>
</html>
```

Properties

- There are 60 different properties in 7 categories:
 - Fonts
 - Lists
 - Alignment of text
 - Margins
 - Colors
 - Backgrounds
 - Borders

Property Values

- Keywords - left, small, ...
 - Not case sensitive
- Length - numbers, maybe with decimal points

- Units:
 - px - pixels
in - inches
 - cm - centimeters
mm - millimeters
 - pt - points
pc - picas (12 points)
 - em - height of the letter 'm'
ex-height - height of the letter 'x'
 - No space is allowed between the number and the unit specification e.g.,
1.5 in is illegal!
- Percentage - just a number followed immediately by a percent sign
- URL values
 - url(protocol://server/pathname)
- Colors
 - Color name
 - rgb(n1, n2, n3)
 - Numbers can be decimal or percentages
 - Hex form: #XXXXXX
- Property values are inherited by all nested tags, unless overridden

Font Properties

- font-family
 - Value is a list of font names - browser uses the first in the list it has
 - font-family: Arial, Helvetica, Courier
 - Generic fonts: serif, sans-serif, cursive, fantasy, and monospace (defined in CSS)
 - Browser has a specific font for each
 - If a font name has more than one word, it should be single-quoted
 - font-size
 - Possible values: a length number or a name, such as smaller, xx-large, etc.
 - font-style
 - italic, oblique (useless), normal
 - font-weight - degrees of boldness
 - bolder, lighter, bold, normal
 - Could specify as a multiple of 100 (100 – 900)
 - font
 - For specifying a list of font properties
- font: bolder 14pt Arial Helvetica
- Order must be: style, weight, size, name(s)
 - > SHOW fonts.html and display

- > SHOW fonts2.html and display
- The text-decoration property
 - line-through, overline, underline, none
 - letter-spacing – value is any length property value

List properties

- list-style-type
- *Unordered lists*
 - Bullet can be a disc (default), a square, or a circle
 - Set it on either the or tag
 - On , it applies to list items

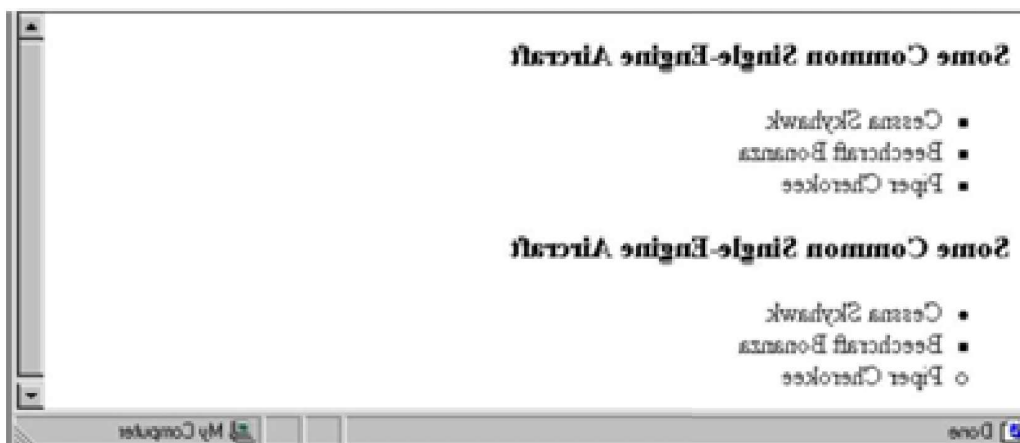
<h3> Some Common Single-Engine Aircraft </h3>

```
<ul style = "list-style-type: square">
  <li> Cessna Skyhawk </li>
  <li> Beechcraft Bonanza </li>
  <li> Piper Cherokee </li>
</ul>
```

- On , list-style-type applies to just that item

<h3> Some Common Single-Engine Aircraft </h3>

```
<ul>
  <li style = "list-style-type: disc">
    Cessna Skyhawk </li>
  <li style = "list-style-type: square">
    Beechcraft Bonanza </li>
  <li style = "list-style-type: circle">
    Piper Cherokee </li>
</ul>
```



- Could use an image for the bullets in an unordered list

■ Example:

```
<li style = "list-style-image:
url(bird.jpg)">
```

- *On ordered lists* - list-style-type can be used to change the sequence values

Property value Sequence type First four

Decimal Arabic numerals 1, 2, 3, 4

upper-alpha Uc letters A, B, C, D

lower-alpha Lc letters a, b, c, d

upper-roman Uc Roman I, II, III, IV

lower-roman Lc Roman i, ii, iii, iv

→ SHOW sequence_types.html and display

- CSS2 has more, like lower-greek and hebrew

Colors

- *Color is a problem for the Web for two reasons:*

1. Monitors vary widely
2. Browsers vary widely

- There are three color collections

1. There is a set of 16 colors that are guaranteed to be displayable by all graphical browsers on all color monitors

black	000000	green	008000
silver	C0C0C0	lime	00FF00
gray	808080	olive	808000
white	FFFFFF	yellow	FFFF00
maroon	800000	navy	000080
red	FF0000	blue	0000FF
purple	800080	teal	008080
fuchsia	FF00FF	aqua	00FFFF

2. There is a much larger set, the Web Palette

- 216 colors
- Use hex color values of 00, 33, 66, 99, CC, and FF
- Inside back cover of this book has them!

3. Any one of 16 million different colors

- The color property specifies the foreground color of elements

```
<style type = "text/css">
th.red {color: red;}
th.orange {color: orange}
```



```
</style>
```

```
...
```

```
<table border = "5">
```

```
<tr>
```

```
<th class = "red"> Apple </th>
```

```
<th class = "orange"> Orange </th>
```

```
<th class = "orange"> Screwdriver </th>
```

```
</tr>
```

```
</table>
```

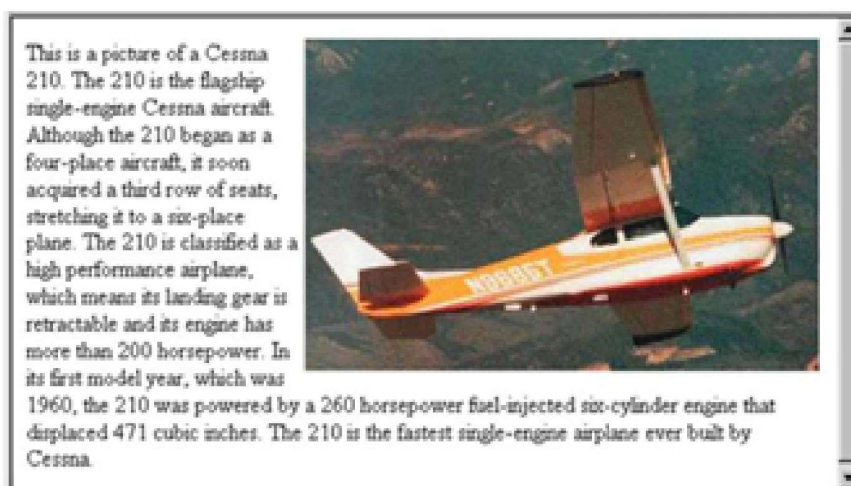
- The background-color property specifies the background color of elements
- SHOW back_color.html and display

Alignment of Text

- The text-indent property allows indentation
 - Takes either a length or a % value
- The text-align property has the possible values, left (the default), center, right, or justify
- Sometimes we want text to flow around another element - the float property
 - The float property has the possible values, left, right, and none (the default)
 - If we have an element we want on the right, with text flowing on its left, we use the default text-align value (left) for the text and the right value for float on the element we want on the right

```
<img src = "c210.jpg"
  style = "float: right" />
```

- Some text with the default alignment - left



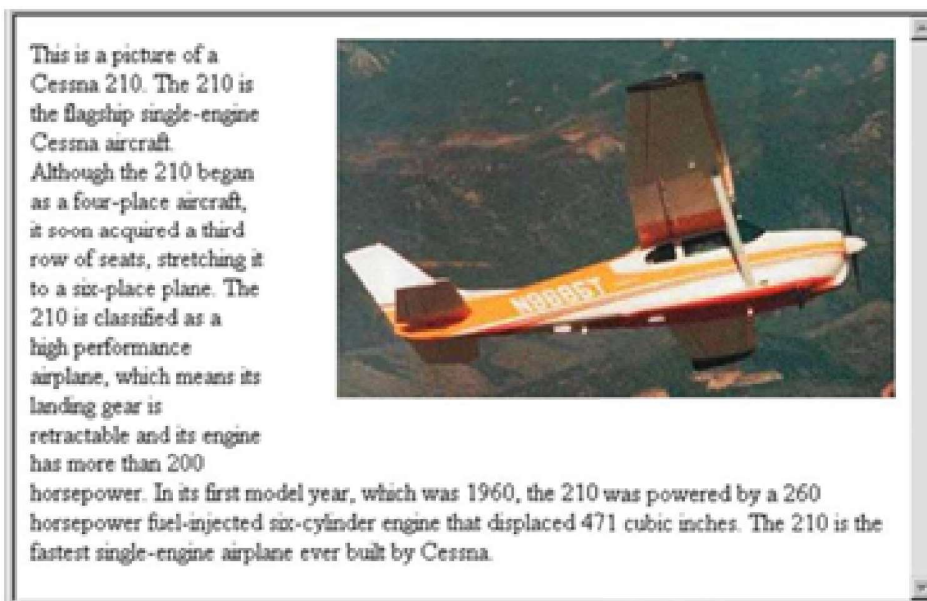
The Box Model

- Borders – every element has a border-style property
 - ❑ Controls whether the element has a border and if so, the style of the border
 - ❑ border-style values: none, dotted, dashed, and double
 - ❑ border-width – thin, medium (default), thick, or a length value in pixels
 - ❑ Border width can be specified for any of the four borders (e.g., border-top-width)
 - ❑ border-color – any color
 - ❑ Border color can be specified for any of the four borders (e.g., border-top-color)

→ SHOW borders.html and display

- Margin – the space between the border of an element and its neighbor element
- The margins around an element can be set with margin-left, etc. - just assign them a length value

```
<img src = "c210.jpg " style = "float: right;  
margin-left: 0.35in;  
margin-bottom: 0.35in" />
```



- Padding – the distance between the content of an element and its border
 - Controlled by padding, padding-left, etc.
- SHOW marpads.html and display

Background Images

- The background-image property
- SHOW back_image.html and display
- Repetition can be controlled
 - background-repeat property
 - Possible values: repeat (default), no-repeat, repeat-x, or repeat-y
 - background-position property
 - Possible values: top, center, bottom, left, or right

The and <div> tags

- One problem with the font properties is that they apply to whole elements, which are often too large
 - Solution: a new tag to define an element in the content of a larger element -
 - The default meaning of is to leave the content as it is

<p>

Now is the best time ever!

</p>

- Use to apply a document style sheet to its content

<style type = "text/css">?

```
bigred {font-size: 24pt;
font-family: Ariel; color: red}
```

</style>

<p>

Now is the

best time ever!

</p>



- The tag is similar to other HTML tags, they can be nested and
- they have id and class attributes
- Another tag that is useful for style specifications: <div>
 - Used to create document sections (or divisions) for which style can be specified
 - e.g., A section of five paragraphs for which you want some particular style

Conflict Resolution

- When two or more rules apply to the same tag there are rules for deciding which rule applies
- Document level
 - In-line style sheets have precedence over document style sheets
 - Document style sheets have precedence over external style sheets
- Within the same level there can be conflicts
 - A tag may be used twice as a selector
 - A tag may inherit a property and also be used as a selector
- Style sheets can have different sources
 - The author of a document may specify styles
 - The user, through browser settings, may specify styles
- Individual properties can be specified as important

Precedence Rules

- From highest to lowest
 1. Important declarations with user origin
 2. Important declarations with author origin
 3. Normal declarations with author origin
 4. Normal declarations with user origin
 5. Any declarations with browser (or other user agent) origin

Tie-Breakers

- Specificity
 1. id selectors
 2. Class and pseudo-class selectors
 3. Contextual selectors
 4. General selectors
- Position
 1. Essentially, later has precedence over earlier