Module 1

Fundamentals of Web, XHTML

1.1 Internet

1.2 WWW

1.3 Web Browsers

1.4 Web Servers

1.5 URLs

1.6 MIME

1.7 HTTP

1.8 Security

1.9 The Web Programmers Toolbox

1.10 XHTML: Basic syntax

1.11 Standard XHTML document structure

1.12 Basic text markup

1.13 Images

1.14 Hypertext Links

1.15 Lists

1.16 Tables

1.17 Forms

1.1 Internet

The **Internet** is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks of local to global scope that are linked by a broad array of electronic and optical networking technologies. The Internet carries a vast array of information resources and services, most notably the inter-linked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail.

Origins:

- **1960s**

- U.S. Department of Defence (DoD) became interested in developing a new large-scale computer network
- The purposes of this network were communications, program sharing, and remote computer access for researchers working on defence-related contracts.
- The DoD's Advanced Research Projects Agency (ARPA) funded the construction of the first such network. Hence it was named as ARPAnet.
- The primary early use of ARPAnet was simple text-based communications through e-mail.

- **Late 1970s and early 1980s**

- BITNET, which is an acronym for ***B**ecause **I**t's **T**ime **NET**work*, began at the City University of New York. It was built initially to provide electronic mail and file transfers.
- CSNET is an acronym for ***C**omputer **S**cience **NET**work*. Its initial purpose was to provide electronic mail.

- **1990s**

- NSFnet which was created in 1986 replaced ARPAnet by 1990.
- It was sponsored by the National Science Foundation (NSF).
- 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 became known as the ***Internet.***

What Is the Internet?

- The Internet is a huge collection of computers connected in a communications network.
- The Transmission Control Protocol/Internet Protocol (TCP/IP) became the standard for computer network connections in 1982.
- Rather than connecting every computer on the Internet directly to every other computer on the Internet, normally the individual computers in an organization are connected to each other in a local network. One node on this local network is physically connected to the Internet.
- So, the Internet is actually a ***network of networks***, rather than a network of computers.
- Obviously, all devices connected to the Internet must be uniquely identifiable.

Internet Protocol Addresses

- The Internet Protocol (IP) address of a machine connected to the Internet is a unique 32-bit number **Internet Protocol Addresses**
- The Internet Protocol (IP) address of a machine connected to the Internet is a unique 32-bit number.
- IP addresses usually are written (and thought of) as four 8-bit numbers, separated by periods.
- The four parts are separately used by Internet-routing computers to decide where a message must go next to get to its destination.
- Although people nearly always type domain names into their browsers, the IP works just as well.
- For example, the IP for United Airlines (www.ual.com) is 209.87.113.93. So, if a browser is pointed at http://209.87.113.93, it will be connected to the United Airlines Web site.

Domain Names

The IP addresses are numbers. Hence, it would be difficult for the users to remember IP address. To solve this problem, text based names were introduced. These are technically known as ***domain name system (DNS).***

These names begin with the names of the host machine, followed by progressively larger enclosing collection of machines, called domains. There may be two, three or more domain names. DNS is of the form hostname.domainName.domainName . Example: vtu.ac.in.The steps for conversion from DNS to IP:

- The DNS has to be converted to IP address before destination is reached.
- This conversion is needed because computer understands only numbers.
- The conversion is done with the help of name server.
- As soon as domain name is provided, it will be sent across the internet to contact name servers.
- This name server is responsible for converting domain name to IP
- If one of the name servers is not able to convert DNS to IP, it contacts other name server.
- This process continues until IP address is generated.
- Once the IP address is generated, the host can be accessed.
- The hostname and all domain names form what is known as FULLY QUALIFIED DOMAIN NAME.

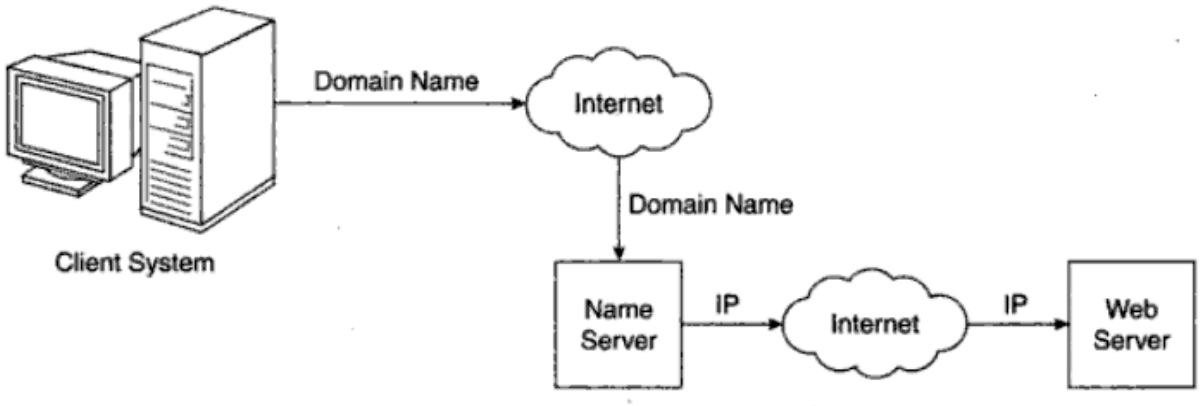


Figure 1.1 Domain name conversion

1.2 WWW

Origins

- Tim Berners Lee and his group proposed a new protocol for the Internet whose intention was to allow scientists around the world to use the Internet to exchange documents describing their work.
- The proposed new system was designed to allow a user anywhere on the Internet to search for and retrieve documents from the databases on any number of different document-serving computers.
- The system used *hypertext*, which is text with embedded links to text in other documents to allow non-sequential browsing of textual material.
- The units of web are referred as pages, documents and resources.
- Web is merely a vast collection of documents, some of which are connected by links.
- These documents can be accessed by web browsers and are provided by web servers.

Web or Internet?

It is important to understand that the Internet and the Web is not the same thing.

- The **Internet** is a collection of computers and other devices connected by equipment that allows them to communicate with each other.
- The **Web** is a collection of software and protocols that has been installed on most, if not all, of the computers on the Internet.

1.3 Web Browsers

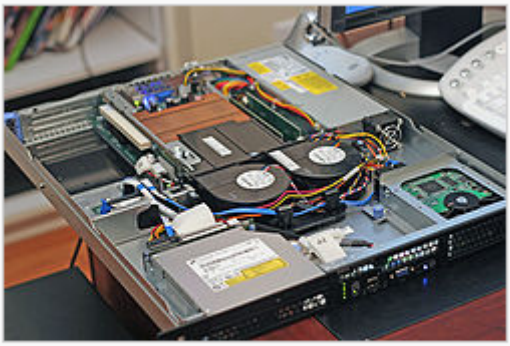
A **web browser** is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI) and may be a web page, image, video, or other piece of content.[1] Hyperlinks present in resources enable users to easily navigate their browsers to related resources.

Although browsers are primarily intended to access the World Wide Web, they can also be used to access information provided by Web servers in private networks or files in file systems. Some browsers can be also used to save information resources to file systems.

Internet Explorer, the most used web browser

1.4 Web Servers

A **web server** is a computer program that delivers (serves) content, such as web pages, using the Hypertext Transfer Protocol (HTTP), over the World Wide Web. The term web server can also refer to the computer or virtual machine running the program. In large commercial deployments, a server computer running a web server can be rack-mounted with other servers to operate a web farm. Example: Apache



The inside and front of a Dell PowerEdge Web server

Web server operations:

- All the communications between a web client and a web server use the HTTP
- When a web server begins execution, it informs the OS under which it is running & it runs as a background process

- A web client or browser, opens a network connection to a web server, sends information requests and possibly data to the server, receives information from the server and closes the connection.
- The primary task of web server is to monitor a communication port on host machine, accept HTTP commands through that port and perform the operations specified by the commands.
- When the URL is received, it is translated into either a filename or a program name.

General characteristics of web server:

- The file structure of a web server has two separate directories
- The root of one of these is called document root which stores web documents
- The root of the other directory is called the server root which stores server and its support softwares
- The files stored directly in the document root are those available to clients through top level URLs
- The secondary areas from which documents can be served are called virtual document trees.
- Many servers can support more than one site on a computer, potentially reducing the cost of each site and making their maintenance more convenient. Such secondary hosts are called virtual hosts.
- Some servers can serve documents that are in the document root of other machines on the web; in this case they are called as proxy servers.

Apache

- Apache is the most widely used Web server.
- The primary reasons are as follows: Apache is an excellent server because it is both fast and reliable.
- Furthermore, it is open-source software, which means that it is free and is managed by a large team of volunteers, a process that efficiently and effectively maintains the system.

- Finally, it is one of the best available servers for Unix-based systems, which are the most popular for Web servers.
- Apache is capable of providing a long list of services beyond the basic process of serving documents to clients.
- When Apache begins execution, it reads its configuration information from a file and sets its parameters to operate accordingly.

1.5 URLs

Uniform Resource Locators (URLs) are used to identify different kinds of resources on Internet.

- If the web browser wants some document from web server, just giving domain name is not sufficient because domain name can only be used for locating the server.
- It does not have information about which document client needs. Therefore, URL should be provided.
- The general format of URL is: ***scheme: object-address***
- Example: http: www.vtu.ac.in/results.php
- The scheme indicates protocols being used. (http, ftp, telnet...)
- In case of http, the full form of the object address of a URL is as follows:

//fully-qualified-domain-name/path-to-document

- URLs can never have embedded spaces
- It cannot use special characters like semicolons, ampersands and colons
- The path to the document for http protocol is a sequence of directory names and a filename, all separated by whatever special character the OS uses. (forward or backward slashes)
- The path in a URL can differ from a path to a file because a URL need not include all directories on the path
- A path that includes all directories along the way is called a **c*omplete path.***
- Example: **http://www.vtu.ac.in/index.html**
- In most cases, the path to the document is relative to some base path that is specified in the configuration files of the server. Such paths are called ***partial paths***.

- Example**: http://www.vtu.ac.in/**

1.6 MIME

MIME stands for Multipurpose Internet Mail Extension.

- The server system apart from sending the requested document, it will also send MIME information.
- The MIME information is used by web browser for rendering the document properly.
- The format of MIME is: type/subtype
- Example: text/html , text/doc , image/jpeg , video/mpeg
- When the type is either text or image, the browser renders the document without any problem
- However, if the type is video or audio, it cannot render the document
- It has to take the help of other software like media player, win amp etc.,
- These softwares are called as helper applications or plugins
- These non-textual information are known as HYPER MEDIA
- Experimental document types are used when user wants to create a customized information & make it available in the internet
- The format of experimental document type is: type/x-subtype
- Example: database/x-xbase, video/x-msvideo
- Along with creating customized information, the user should also create helper applications.
- This helper application will be used for rendering the document by browser.
- The list of MIME specifications is stored in configuration file of web server.

1.7 HTTP

Request Phase:

The general form of an HTTP request is as follows:

1. HTTP method Domain part of the URL HTTP version

2. Header field.

3. Blank line

4. Message body

The following is an example of the first line of an HTTP request: GET /storefront.html HTTP/1.1.

Table 1.1 HTTP request methods

Method	Description
GET	Returns the contents of the specified document
HEAD	Returns the header information for the specified document
POST	Executes the specified document, using the enclosed data
PUT	Replaces the specified document with the enclosed data
DELETE	Deletes the specified document

The format of a header field is the field name followed by a colon and the value of the field. There are four categories of header fields:

1. ***General***: For general information, such as the date

2. ***Request***: Included in request headers

3. ***Response***: For response headers

4. ***Entity***: Used in both request and response headers

A wildcard character, the asterisk (*), can be used to specify that part of a MIME type can be anything.

The Response Phase:

The general form of an HTTP response is as follows:

1. Status line

2. Response header fields

3. Blank line

4. Response body

The status line includes the HTTP version used, a three-digit status code for the response, and a short textual explanation of the status code. For example, most responses begin with the following:

HTTP/1.1 200 OK

The status codes begin with 1, 2, 3, 4, or 5. The general meanings of the five categories specified by these first digits are shown in Table 1.2.

Table 1.2 First digits of HTTP status codes

First Digit	Category
1	Informational
2	Success
3	Redirection
4	Client error
5	Server error

1.8 Security

Security is one of the major concerns in the Internet. The server system can be accessed easily with basic hardware support, internet connection & web browser. The client can retrieve very important information from the server. Similarly, the server system can introduce virus on the client system. These viruses can destroy the hardware and software in client. While programming the web, following requirements should be considered:

Privacy: it means message should be readable only to communicating parties and not to intruder.

Integrity: it means message should not be modified during transmission.

Authentication: it means communicating parties must be able to know each other's identity

Non-repudiation: it means that it should be possible to prove that message was sent and received properly.

Security can be provided using cryptographic algorithm. Ex: private key, public key Protection against viruses and worms is provided by antivirus software, which must be updated

frequently so that it can detect and protect against the continuous stream of new viruses and worms.

1.9 XHTML: Basic syntax

XHTML has strict syntactic rules that impose a consistent structure on all XHTML documents. Another significant reason for using XHTML is that when you create an XHTML document, its syntactic correctness can be checked, either by an XML browser or by a validation tool .Writing valid HTML (or XHTML) is not a terribly difficult task once you know what the rules are, although the rules are slightly more stringent in XHTML than in HTML.

The list below provides a quick reference to the rules that will ensure your markup is well-formed and valid.

The Document Tree

A web page is, at its heart, little more than a collection of HTML elements—the defining structures that signify a paragraph, a table, a table cell, a quote, and so on. The element is created by writing an opening tag, and completed by writing a closing tag. In the case of a paragraph, you'd create a `p` element by typing `<p>Content goes here</p>`.The elements in a web page are contained in a tree structure in which `html` is the root element that splits into the `head` and `body` elements (as explained in Basic Structure of a Web Page).

An element may contain other nested elements (although this very much depends on what the parent element is; for example, a `p` element can contain `span`, `em`, or `strong` elements, among others). Where this occurs, the opening and closing tags must be symmetrical. If an opening paragraph tag is followed by the opening `em` element, the closing tags must appear in the reverse order, like so: `<p>Content goes here, <em>and some of it needs emphasis</em> too</p>`. If you were to type `<p>Content goes here, <em>and some of it needs emphasis too</p></em>`, you'd have created invalid markup.

Case Sensitivity

In HTML, tag names are case insensitive, but in XHTML they're case sensitive. As such, in HTML, you can write the markup in lowercase, mixed case, or uppercase letters. So `this is a paragraph` , as is <P>`this example`</P>, and even <P>`this markup would be valid` . In XHTML, however, you must use lowercase for markup: `<p>This is a valid paragraph in XHTML</p>`.

Opening and Closing Tags

In HTML, it's possible to omit some closing tags (check each element's reference to see whether an HTML closing tag is required), so this is valid markup: `<p>This is my first paragraph.<p>This is my second paragraph.<p>And here's the last one.`.

In XHTML, all elements must be closed. Hence the paragraph example above would need to be changed to: `<p>This is my first paragraph.</p>.<p>This is my second paragraph.</p>.<p>And here's the last one.</p>`.As well as letting you omit some closing tags, HTML allows you to omit start tags—but only on the `html`, `head`, `body`, and `tbody` elements. This is not a recommended practice, but is technically possible.

For empty elements such as `img`, XHTML (that is not served with the application/xhtml+xml) requires us to use the XML empty element syntax: `<elementname attribute="attributevalue"/>`.If serving the document as application/xhtml+xml, it's also valid to close empty elements using a start and end tag, for example the `img` element, as `<img></img>`

Readability Considerations

A browser doesn't care whether you use a single space to separate attributes, ten spaces, or even complete line breaks; it doesn't matter, as long as some space is present. As such, all of the examples below are perfectly acceptable (although the more spaces you include, the larger your web page's file size will be—each occurrence of whitespace takes up additional bytes—so the first example is still the most preferable):

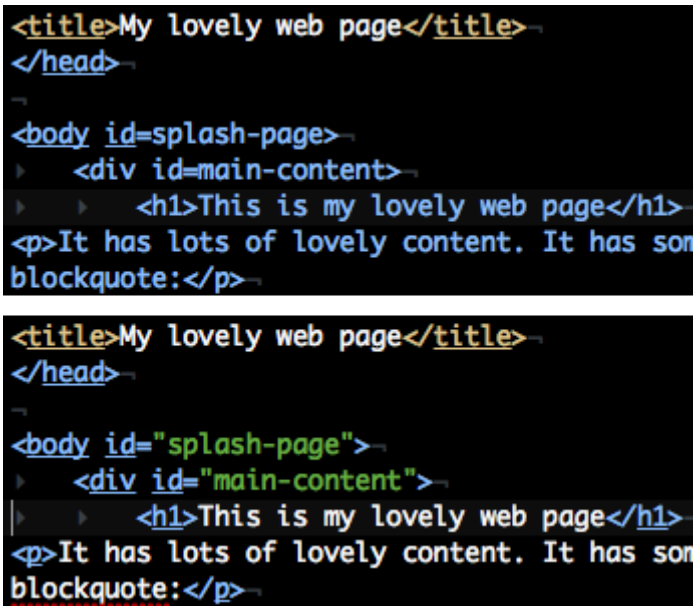
```
<img src="/images/burj.jpg" alt="Burj Al Arab, iconic hotel in
Dubai" class="gallery"/>
<img
    src="/images/burj.jpg"
    alt="Burj Al Arab, iconic hotel in Dubai"
    class="gallery"
/>
<img
    src="/images/burj.jpg"
    alt="Burj Al Arab, iconic hotel in Dubai"
   class="gallery"/>
```

In XHTML all attribute values must be quoted, so you'll need to write `class="gallery"` rather than `class=gallery`. It's valid to omit the quotes from your HTML, though it may make reading the markup more difficult for developers revisiting old markup (although this really depends on the developer—it's a subjective thing). It's simply easier always to add quotes, rather than to have to remember in which scenarios attribute values require quotes in HTML, as the following piece of HTML demonstrates:

```
<a href="http://example.org"> needs to be quoted because it contains a /
<a href=index.html> acceptable without quotes in HTML
```

Another reason why it's a good idea always to quote your attributes, even if you're using HTML 4.01, is that your HTML editor may be able to provide syntax coloring that makes the code even easier to scan through. Without the quotes, the software may not be able to identify the difference between elements, attributes, and attribute values. This fact is illustrated in Figure , which shows a comparison between quoted and unquoted syntax coloring in the Mac text editor TextMate.

Figure: ***TextMate's syntax coloring taking effect to display quoted attributes***



Commenting Markup

You may add comments in your HTML, perhaps to make it clear where sections start or end, or to provide a note to remind yourself why you approached the creation of a page in a certain way. What you use comments for isn't important, but the way that you craft a comment is important. The HTML comment looks like this: `<!-- this is a comment -->`. It's derived from SGML, which starts with an `<!` and ends with an `>`; the actual comment is, in effect, inside the opening `--` and the closing `--` parts. These hyphens tell the browser when to start ignoring text content, and when to start paying attention again. The fact that the double hyphen `--` characters signify the beginning and end of the comment means that you should not use double hyphens anywhere inside a comment, even if you believe that your usage of these characters conforms to SGML rules. Single hyphens are allowed, however.

The markup below shows examples of good and bad HTML comments—see the remark associated with each example for more information:

```
<p>Take the next right.<!-- Look out for the
   signpost for 'Castle' --></p> a valid comment
```

Take the next right.<!-- Look out for -- Castle -->
not a valid comment; the double dashes in the middle could be
misinterpreted as the end of the comment

Take the next right.<!-- Look out for -- -- Castle -->
a valid comment; 'Look out for' is one comment, 'Castle' is another

Take the next right.
<!---------------------------------
This is just asking for trouble. Too
many hyphens! -->
a valid comment; don't use hyphens or <> characters to format comment text

<p <!-- class="lively" -->>Wowzers!
It's not possible to comment out attributes inside an HTML element

1.11 Standard XHTML document structure

An XHTML document consists of three main parts:

- DOCTYPE
- Head
- Body

The basic document structure is:

```
<!DOCTYPE ...>

<html  ... >
<head> ... </head>
```

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

The <head> area contains information about the document, such as ownership, copyright, and keywords; and the <body> area contains the content of the document to be displayed.

Listing 1 shows you how this structure might be used in practice:

Listing 1. An XHTML example

```
1.  <?xml version="1.0"?>
2.  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
    Transitional//EN"  "DTD/xhtml1-transitional.dtd">
3.  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
                lang="en">
4.  <head>
    <title>My XHTML Sample Page</title>

    </head>
5.  <body bgcolor="white">
    <center><h1>Welcome to XHTML !</h1></center>
    </body>
6.  </html>
```

Line 1: Since XHTML is HTML expressed in an XML document, it must include the initial XML declaration <?xml version="1.0"?> at the top of the document.

Line 2: XHTML documents must be identified by one of three standard sets of rules. These rules are stored in a separate document called a Document Type Declaration (DTD), and are utilized to

validate the accuracy of the XHTML document structure. The purpose of a DTD is to describe, in precise terms, the language and syntax allowed in XHTML.

Line 3: The second tag in an XHTML document must include the opening <html> tag with the XML namespace identified by the xmlns=http://www.w3.org/1999/xhtml attribute. The XML namespace identifies the range of tags used by the XHTML document. It is used to ensure that names used by one DTD don't conflict with user-defined tags or tags defined in other DTDs.

Line 4: XHTML documents must include a full header area. This area contains the opening <head> tag and the title tags (<title></title>), and is then completed with the closing </head> tag.

Line 5: XHTML documents must include opening and closing <body></body> tags. Within these tags you can place your traditional HTML coding tags. To be XHTML conformant, the coding of these tags must be well-formed.

Line 6: Finally, the XHTML document is completed with the closing </html> tag.

1.12 Basic text markup

A **markup language** is a modern system for annotating a text in a way that is syntactically distinguishable from that text. The idea and terminology evolved from the "marking up" of manuscripts, i.e. the revision instructions by editors, traditionally written with a blue pencil on authors' manuscripts. Examples are typesetting instructions such as those found in troff and LaTeX, and structural markers such as XML tags. Markup is typically omitted from the version of the text which is displayed for end-user consumption. Some markup languages, like HTML have presentation semantics, meaning their specification prescribes how the structured data is to be presented, but other markup languages, like XML, have no predefined semantics.

A well-known example of a markup language in widespread use today is HyperText Markup Language (HTML), one of the document formats of the World Wide Web. HTML is mostly an instance of SGML (though, strictly, it does not comply with all the rules of SGML) and follows many of the markup conventions used in the publishing industry in the communication of printed work between authors, editors, and printers.

1.13 Images

Images can be included to enhance appearance of the document. Most common methods of representing the images are GIF (graphical interchange format and JPEG (joint photographic experts group) format. The former is 8bit color representation whereas the latter is 24 bit color representation. The image tag specifies an image that appears in a document. It has attributes like src which specifies the source of the image.

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> Images</title></head>
<body>
<h1> Twinkle twinkle
<h2>little star
<h3> how I wonder
<h4>what you are ???
up above the world so high
like a diamond in the sky.
<img src="hills.jpg" alt ="my picture" height=50 width=50 />
</body>
</HTML>
```

XHTML validation

1.14 Hypertext link

A hypertext link in a XHTML document acts as a pointer to some resource. It could be an XHTML document anywhere in the web or another place in the document currently displayed. Links that point to another document specifies the address of the document. It could be a filename, complete url, directory path and a filename. Links are specified in an attribute of an anchor tag <a> which is an inline tag. The anchor tag is the source of an link whereas the document is the target of the link. Links facilitate reader to click on links to learn more about a particular subtopic of interest and also return back to the location of the link

If target is in the same document as the link it is specified in the href attribute value by preceding the id with a pound sign(#)

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> Images</title></head>
<body>
<h1> Twinkle twinkle
<h2>little star
<h3> how I wonder
<h4>what you are ??? up above the world so high like a diamond in the sky.
<p>
<a href = "C:\Documents and Settings\Administrator\My Documents\XHTML programs\1my.html">The blue hill image document
</a><p>
</body>
</HTML>
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> Images</title></head>
<body>
<h1 id = "twinkle" > Twinkle twinkle </h1>
<h2>little star</h2>
<h3> how I wonder </h3>
<h4 >what you are ???up above the world so high like a diamond in the sky.</h4>
<a href = "#twinkle">which poem
</a>
</body>
</HTML>
```

Twinkle twinkle

little star

how I wonder

what you are ??? up above the world so high like a diamond in the sky.

Which poem

1.15 Lists

XHTML provides simple and effective ways to specify both ordered and unordered lists <ul> are tags for unordered and ordered lists. Each item in a list is specified with an <li> tag. Any tags can appear in a list item including nested lists.

Definition lists

They are used to specify list of terms and their definitions such as glossaries. They are given by <dl> tag which is a block tag. The definitions are specified as the content of <dd> tag and the definition list is given as the content of a <dt> tag

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> ordered lists</title></head>
<body>
<h3 > Lists of poems</h1>
<ul>
<li> Twinkle twinkle</li>
<li> Baa Baa black sheep</li>
<li> pussy cat </li>
<li> Humpty dumpty</li>
</ul>
</body>
</html>
<body>
<h3 > Lists of poems</h1>
<ol>
```

 Twinkle twinkle

 Baa Baa black sheep

 pussy cat

 Humpty dumpty

</body>

Lists of poems

1. Twinkle twinkle
2. Baa Baa black sheep
3. pussy cat
4. Humpty dumpty

Lists of poems

- Twinkle twinkle
- Baa Baa black sheep
- pussy cat
- Humpty dumpty

head><title> ordered lists</title></head>

<body>

 Lists of poems

 Twinkle twinkle

 Baa Baa black sheep

 pussy cat

 Humpty dumpty

 Lists of stories

```
<ol>
<li> Thirsty crow</li>
<li> Lion and the mouse</li>
<li> pussy cat </li>
<li> Midas touch</li>
</ol>
</li>
</ol>
```

1. Lists of poems
    1. Twinkle twinkle
    2. Baa Baa black sheep
    3. pussy cat
    4. Humpty dumpty
2. Lists of stories
    1. Thirsty crow
    2. Lion and the mouse
    3. pussy cat
    4. Midas touch

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title> ordered lists</title></head>
<body>
<h3 > Lists of poems </h3>
<dl>
<dt> 111 </dt>
<dd> Twinkle twinkle</dd>
<dt> 222 </dt>
<dd> Pily poly</dd>
<dt> 333 </dt>
<dd> Rudolf the reindeer</dd>
</dl>
</body>
</HTML>
```

Lists of poems

111
Twinkle twinkle
222
Pily poly
333
Rudolf the reindeer

1.16 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 which is a block tag. A border attribute in the <table> tag specifies a border between the cells. Rule specifies the lines that separate 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 in pixels(0 is no border no rules). 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 are specified as the content of a <td> tag. The empty cell is specified with a table header tag that includes no content <th> </th>.

```
<body>
<table border = "border">
<caption> fruit juice drinks </caption>
<tr>
<th> </th>
<th> Apple </th>
<th> Mango </th>
<th> Strawberry </th>
</tr>
<tr> <th> Breakfast </th>
<th> 0</th>
```

<th> 1 </th>

<th> 0</th>

</tr>

<tr> <th> lunch </th>

<th> 1</th>

<th> 1 </th>

<th>1 </th>

</tr>

<tr> <th> dinner </th>

<th> 0 </th>

<th> 1 </th>

<th> 0 </th></tr><table></body></HTML>

Colspan Rowspan attributes

A table can have two levels column labels and also row 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>

<caption> fruit juice drinks </caption>

<tr>

<td rowspan="2"></td>

<th colspan="3">Juices chart</th>

</tr>

<tr>

<th> </th>

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>
```

fruit juice drinks

	Apple	Mango	Strawberry
Breakfast	0	1	0
lunch	1	1	1
dinner	0	1	0

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. 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.

```
<head><title> simple table</title></head>
<body>
<table border = "border">
```

```
<caption> align</caption>
<tr align ="center">
<th> </th>
<th> Apple </th>
<th> Mango </th>
<th> Strawberry </th>
</tr>
<tr>
<th> align </th>
<td align="left">left</td>
<td align="center">center</td>
<td align="right">right</td>
</tr>
<tr>
<th> valign<br /> <br /></th>
<td >default</td>
<td valign="top">top</td>
<td valign="bottom">bottom</td>
</tr
<table>
</body>
</HTML>
```

align

	Apple	Mango	Strawberry
align	left	center	right
valign	default	top	bottom

1.17 Forms

A form is the usual way information is gotten from a Web 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). All control tags are inline tags. These controls gather information used from user in the form of either text or button selections. Each control has a value given through the user input. Collectively values of all these controls in a form are called the form data. When the Submit button of a form is clicked, the form's values are sent to the Web server for processing.

The <form> tag

All of the widgets, or components of a form are defined in the content of a <form> tag which is a block tag. This tag can have many attributes of which the most required attribute is the action. The action attribute specifies the URL of the application on the web server that is to be called when the user clicks the Submit button.

Eg: 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 techniques, get or post, used to pass the form data to the server. get is the default, so if no method attribute is given in the <form> tag, get will be used. The alternative technique is post. With these techniques the form data is encoded into text string on click of submit button Widgets or Controls Many commonly used controls are created with the <input> tag which specifies the kind of control. It is used for the text, passwords, checkboxes, radio buttons and the action buttons Reset and Submit.

The type attribute of <input> specifies the kind of widget being created. Except Reset and Submit all other controls have a name attribute other than type attribute.

Text: Creates a horizontal box for text input

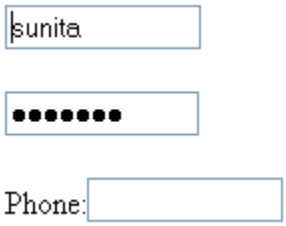
Default size is 20; it can be changed with the size Attribute. If more characters are entered than will fit, the box is scrolled (shifted) left. If you don't want to allow the user to type more characters than will fit, set max length, which causes excess input to be ignored

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

If the contents of the textbox should not be displayed when user types it than a password control should be used. Labeling a text box can be done by adding label control to the text box. Both the controls can be encapsulated. This has several advantages. The text content of the label will indicate content of text box and when a label is selected the cursor is implicitly moved to the control.

<form action = "">

```
<p>
<input type = "text" name = "Phone" size = "12" /></p>
<p>
<input type = "password" name = "myPassword" size = "12" maxlength="12"/ >
</p>
<p>
<label>Phone:<input type = "text" name = "Phone" size = "12" /><label>
</p>
```



Checkboxes

Checkboxes 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".

Radio Buttons

Radio buttons are 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. If no button in a radio button group is 'pressed', the browser often 'presses' the first one. Checkboxes and radio buttons are both multiple choice input from the user

```
<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>

<p> <input type = "radio" name = "age" value = "under20" checked = "checked">

0-19

<input type = "radio" name = "age" value = "20-35">

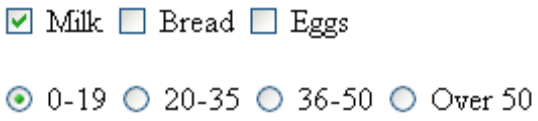
20-35

<input type = "radio" name = "age" value = "36-50">

36-50

<input type = "radio" name = "age" value = "over50">

Over 50



Menus

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. 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 beset 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

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.

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>.

A Submit button is required in every form

```
<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>
<p>
<textarea name = "aspirations" rows = "3”
cols = "40">
(Be brief in expressing your views)
</textarea>
</p>
<p>
<input type = "Submit” value = "Submit Form" />
<input type = "reset" value = "Reset Form" />
</p>
</form>
```

Module 2: CSS, Javascript

2.1 CSS: Introduction

2.2 Levels of style sheets

2.3 Style specification formats

2.4 Selector forms

2.5 Property value forms

2.6 Font properties

2.7 List properties

2.8 Color

2.9 Alignment of text

2.10 The Box model

2.11 Background images

2.12 Conflict resolution

2.13 Overview of Javascript

2.14 Object orientation and Javascript

2.15 General syntactic characteristics

2.16 Primitives, operations, and expressions

2.1 CSS: Introduction

The CSS1 specification was developed in 1996 by W3C. CSS2 was released in 1998 which added many properties and values to CSS1. CSS3 has been under development since the late 1990s. CSSs provide the means to control and change presentation of HTML documents. CSS is not technically HTML, but can be embedded in HTML documents.

Cascading style sheets were introduced to provide a uniform and consistent way to specify presentation details in XHTML documents. Most of the style tags and attributes are those deprecated from HTML 4.0 in favor of style sheets. Idea of style sheets is not a new concept as it existed in desktop publishing systems and word processors. Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents. Style sheets can be defined at three levels to specify the style of a document .Hence called Cascading style sheets. Style is specified for a tag by the values of its properties.

For example:

<h2 { font-size: 22pt; } >

2.2 Levels of Style Sheets

There are three levels of style sheets, in order from lowest level to highest level, are inline, document level, and external. Inline style sheets are specified for a specific occurrence of a tag and apply only to the content of that tag. This application of style, which defeats the purpose of style sheets – that of imposing uniform style on the tags of at least one whole document. Another disadvantage of inline style sheets is that they result in style information, which is expressed in a expressed in a language distinct from XHTML markup, being embedded in various places in documents.

Document-level style specifications appear in the document head section and apply to the whole body of the document. External style sheets are not part of the documents to which they apply.

They are stored separately and are referenced in all documents that use them. They are written as text files with MIME type text/css.

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<title> Our first document </title>
<style>
h2 {font-size: 32pt; font-weight: bold;font-family: 'Times New Roman';}
h3,h4 { font-size: 18pt; font-family: 'Courier New'; font-style:italic;
font-weight:bold}
</style>
</head> <body>
```

<h1> Twinkle twinkle little star</h1>

<h2> how I wonder </h2>

<h3>what you are ???</h3>

<h4>up above the world so high</h4>

<h5> like a diamond</h5>

<h6> in the sky.</h6>

</body>

</html>

Twinkle twinkle little star

how I wonder

what you are ???

up above the world so high

like a diamond

in the *sky.*

Linking an External Style sheet

A <link> tag is used to specify that the browser is to fetch and use an external style sheet file through href. The href attribute of <link> is used to specify the URL of the style sheet document, as in the following example:

<link rel = "stylesheet" type = "text/css"

href = “http://www.wherever.org/termpaper.css"> </link>

This link must appear in the head of the document. External style sheets can be validated, with the service http://jigsaw.w3.org/css-validator/ validator-upload.html. External style sheets can be added using other alternate style specification known as file import

- @import url (filename);

Filename is not quoted. Import appears only at the beginning of the content of a style element. The file imported can contain both markup as well as style rules

2.3 Style Specification Formats

The format of a style specification depends on the level of the style sheet. Inline **s**tyle sheet appears as the value of the style attribute of the tag. The general form of which is as follows:

style = "property_1: value_1;

property_2: value_2;

…

property_n: value_n;"

Format for Document-level

Document style specifications appear as the content of a style element within the header of a document, although the format of the specification is quite different from that of inline style sheets. The <style> tag must include the type attribute, set to "text/css“ (as there are other style sheets in JavaScript).

The list of rules must be placed in a comment, because CSS is not XHTML. Style element must be placed within the header of a document. Comments in the rule list must have a different form use C comments (/*…*/). The general form of the content of a style element is as follows:

<style type = "text/css">

/*

rule list(/* styles for paragraphs and other tags*/)

*/</style>

Form of the rules:

Each style in a rule list has two parts: selector, which indicates the tag or tags affected by the rules. Each property/value pair has the form->property: value

Selector { property_1: value_1; property_2: value_2:… property_n: value_n;}

Pairs are separated by semicolons, just as in the value of a <style> tag.

2.4 Selector Forms

Selector can have variety of forms like:

1. Simple selector form
2. Class selector
3. Generic selector
4. Id selector
5. Universal selector
6. Pseudo classes

Simple selector form

Simple selector form is a list of style rules, as in the content of a <style> tag for document-level style sheets. The selector is a tag name or a list of tag names, separated by commas. Consider the following examples, in which the property is font-size and the property value is a number of points :

h1, h3 { font-size: 24pt ;}

h2 { font-size: 20pt ;}

Selectors can also specify that the style should apply only to elements in certain positions in the document .This is done by listing the element hierarchy in the selector.

• Contextual selectors: Selectors can also specify that the style should apply only to elements in certain positions in the document .

• In the eg selector applies its style to the content of emphasis elements that are descendants of bold elements in the body of the document. body b em {font-size: 24pt ;} Also called as descendant selectors. It will not apply to emphasis element not descendant of bold face element.

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 the tag's name with a period.

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,

Once upon a time there lived a king in the place called Ayodhya.

...

Once upon a time there lived a king in the place called Ayodhya.

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 without a tag name in its name.

For Example:

.really-big { … }

Use it as if it were a normal style class

<h1 class = "really-big"> This Tuesday is a holiday </h1>...

 …

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

<head>

<title> Absolute positioning </title>

<style type = "text/css">

.regtext {font-family: Times; font-size: 14pt; width: 600px}

.abstext {position: absolute; top: 25px; left: 50px; font-family: Times; font-size: 24pt; fontstyle: italic; letter-spacing: 1em; color: rgb(102,102,102); width: 500px}

</style>

</head>

<body>

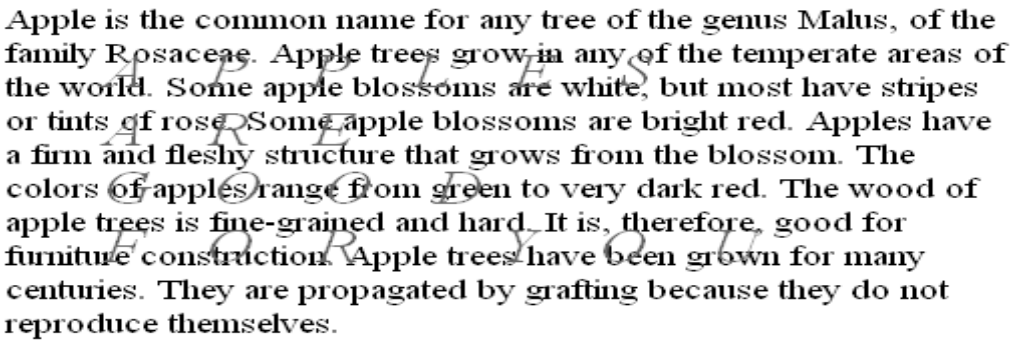
Apple is the common name for any tree of the genus Malus, of the family Rosaceae. Apple trees grow in any of the temperate areas of the world. Some apple blossoms are white, but most have stripes or tints of rose. Some apple blossoms are bright red. Apples have a firm and fleshy structure that grows from the blossom. The colors of apples range from green to very dark red. The wood of apple trees is fine-grained and hard. It is, therefore, good for furniture construction.

Apple trees have been grown for many centuries. They are propagated by grafting because they do not reproduce themselves.

 APPLES ARE GOOD FOR YOU

</body>

</html>



Id Selectors

An id selector allow the application of a style to one specific element. The general form of an id selector is as follows :

#specific-id {property-value list}

Example:

#section14 {font-size: 20} specifies a font size of 20 points to the element

<h2 id =“section14”> Alice in wonderland</h2>

Universal selector

The universal selector, denoted by an asterisk(*), which applies style to all elements in the document. For example:

{color: red;}

makes all elements in the document red.

Twinkle twinkle little star

how I wonder

what you are ???

up above the world so high

like a diamond

in the sky.

Pseudo Classes

Pseudo classes are styles that apply when something happens, rather than because the target element simply exists. Names of pseudo classes begin with colons hover classes apply when the mouse cursor is over the element focus classes apply when an element has focus i.e. the mouse cursor is over the element and the left mouse button is clicked. These two pseudo classes are supported by FX2 but IE7 supports only hover.

```
<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>
```

Your name: sunita

Your name: sunita

Your name:

2.5 Property Values Forms

CSS1 includes 60 different properties in 7 categories(list can be found in W3C website) Fonts, Lists, Alignment of text, Margins, Colors, Backgrounds, Borders. Keywords property values are used when there are only a few possible values and they are predefined

Eg: small, large, medium.

Keyword values are not case sensitive, so Small, SmAlL, and SMALL are all the same as small. Number values can be integer or sequence of digits with decimal points and a + or – sign. Length value are specified as number values that are followed immediately by a two character abbreviation of a unit name. There can be no space between the number and the unit name. The possible unit names are px for pixels, in for inches, cm for centimeters, mm for millimeters, pt for points, pc for picas (12 points),em for value of current font size in pixels, ex for 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: eg: font size set to 85% means new font size will be 85% of the previous font size value.

URL values: URL property values use a form that is slightly different from references to URLs in links. The actual URL, which can be either absolute or relative, is placed in parentheses and preceded by url, as in the following:

url(protocol://server/pathname)

No space should be left between URL and the left parenthesis.

Colors : Color name rgb(n1, n2, n3). Hex form: #B0E0E6 stands for powder blue color. Property values are inherited by all nested tags, unless overridden.

2.6 Font properties

Font-family

The font-family property is used to specify a list of font name. The browser will use the first font in the list that it supports. For example, the following could be specified. font-family: Arial, Helvetica, Courier

Generic fonts: They can be specified as the font family value for example :serif, sans-serif, cursive, fantasy, and monospace (defined in CSS). Browser has a specific font defined for each generic name. If a font name that has more than one word, it should be single-quoted Eg: font-family: ’Times New Roman’

Font-size

Possible values: a length number or a name, such as smaller, xx-large, medium , large etc. Different browsers can use different relative value for the font-size.

Font-variant

The default value of the font-variant property is normal, which specifies the usual character font. This property can be set to small-caps to specify small capital characters.

Font-style

The font-style property is most commonly used to specify italic, as in the following example. Eg: font-style: italic

Font-weights

The font-weight property is used to specify the degree of boldness. For example: font-weight: bold

Font Shorthands

If more than one font property is to be specified than the values may be stated in a list as the value of the font property . The browser will determine from the form of the values which properties to assign. For example, consider the following specification : Eg: font: bold 24pt ’Times New Roman’ Palatino Helvetica The order which browser follows is last must be font name, second last font size and then the font style, font variant and font weight can be in any order but before the font size and names. Only the font size and the font family are required in the font value list. Below example displays the fonts.html

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

<head> <title> Font properties </title>

```
<style type = "text/css">
p.big {font-size: 14pt;
font-style: italic;
font-family: 'Times New Roman';
}
p.small {font: 10pt bold 'Courier New';}
h2 {font-family: 'Times New Roman';
font-size: 24pt; font-weight: bold}
h3 {font-family: 'Courier New'; font-size: 18pt}
</style>
</head>
<body>
<p class = "big">
Where there is will there is a way.
</p>
<p class = "small">
Practise makes a man perfect.
</p>
<h2> Chapter 1 Introduction </h2>
<h3> 1.1 The Basics of Web programming </h3>
<h4> Book by Robert Sebesta
</body>
</html>
```

Where there is will there is a way.

Practise makes a man perfect.

Chapter 1 Introduction

1.1 The Basics of Web programming

Book by Robert Sebesta

Text Decoration

The text-decoration property is used to specify some special features of the text. The available values are line-through, overline, underline, and none, which is the default. Many browsers underline links. Text decoration is not inherited Eg:line-through, overline, underline, none

Letter-spacing – value is any length property value controls amount of space between characters in text

• Eg: 3px

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Text decoration </title>
<style type = "text/css">
p.through {text-decoration: line-through}
p.over {text-decoration: overline}
p.under {text-decoration: underline}
</style> </head>
<body>
<p class = "through">
Twinkle twinkle little star how i wonder what you are!!!!! </p>
<p class= "over">
Twinkle twinkle little star how i wonder what you are!!!!! </p>
<p class = "under">
Twinkle twinkle little star how i wonder what you are!!!!! </p>
```

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

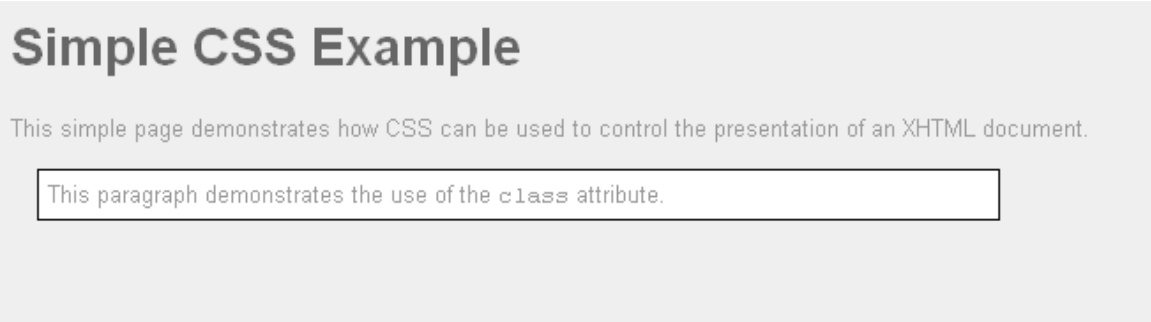
~~Twinkle twinkle little star how i wonder what you are!!!!!~~

Twinkle twinkle little star how i wonder what you are!!!!!

Twinkle twinkle little star how i wonder what you are!!!!!

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS Example</title>
<link rel="stylesheet" type="text/css" href="ch03_eg01.css" />
</head>
<body>
<h1>Simple CSS Example</h1>
<p>This simple page demonstrates how CSS can be used to control the presentation of an
XHTML document.</p>
<p class="important">This paragraph demonstrates the use of the <code>class</code>
attribute.</p>
</body>
</html>
/* CSS Document for ch03_eg01.html */
body {
font-family:arial, verdana, sans-serif;
background-color:#efefef;}
h1 {
color:#666666;
font-size:22pt;}
p {
color:#999999;
font-size:10pt;}
p.important {
```

```
border:solid black 1px;
background-color:#ffffff;
padding:5px;
margin:15px;
width:40em;}
```




```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS Example</title>
<link rel="stylesheet" type="text/css" href="ch03_eg02.css" /> </head> <body>
<p class="one">The first paragraph of text should be displayed in a sans-serif font.</p>
<p class="two">The second paragraph of text should be displayed in a serif font.</p>
<p class="three">The third paragraph of text should be displayed in a monospaced font.</p>
<p class="four">The fourth paragraph of text should be displayed in a cursive font.</p> <p
class="five">The fifth paragraph of text should be displayed in a fantasy font.</p> </body>
</html>
CSS Document for ch03_eg02.html */
p.one {font-family:arial, verdana, sans-serif;}
p.two {font-family:times, "times new roman", serif;}
p.three {font-family:courier, "courier new", monospace;}
p.four {font-family: Zapf-Chancery, Santivo, cursive;}
p.five {font-family:Cottonwood, Studz, fantasy;}
```

The first paragraph of text should be displayed in a sans-serif font.

The second paragraph of text should be displayed in a serif font.

The third paragraph of text should be displayed in a monospaced font.

The fourth paragraph of text should be displayed in a cursive font.

The fifth paragraph of text should be displayed in a fantasy font.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>CSS Example</title>
<link rel="stylesheet" type="text/css" href="ch03_eg08.css" /> </head>
<body>
<h1>Lengths</h1>
<p class="px">The length used here is 12 px</p> <p class="pt">The length used here is 12
pt</p>
<p class="pc">The length used here is 2 pc</p> <p class="in">The length used here is
0.5in</p>
<p class="cm">The length used here is 1cm</p> <p class="mm">The length used here is
12mm</p>
<p class="em">The length used here is 1.5em</p> <p class="ex">The length used here is
1.5ex</p>
</body> </html>
* CSS Document for ch03_eg08.html */
p {font-family:arial; font-size:12pt;}
/* lengths */
p.px {font-size:12px;}
p.pt {font-size:12pt;}
p.pc {font-size:2pc;}
p.in {font-size:0.5in;}
p.cm {font-size:1cm;}
p.mm {font-size:12mm;}
p.em {font-size:1.5em;}
```

p.ex {font-size:1.5ex;}

Lengths

The length used here is 12 px

The length used here is 12 pt

The length used here is 2 pc

The length used here is 0.5in

The length used here is 1cm

The length used here is 12mm

2.7 List properties

It is used to specify style of bullets or sequencing values in list items. The list-style-type of Unordered lists can be set to disc,circle,square or none. Bullet can be a disc (default), a square, or a circle. Set it on either the <ul> or <li> tag On <ul>, it applies to list items

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

<ul style = "list-style-type: square">

 Cessna Skyhawk

 Beechcraft Bonanza

 Piper Cherokee

On <li>, list-style-type applies to just that item

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

<li style = "list-style-type: disc">

Cessna Skyhawk

<li style = "list-style-type: square">

Beechcraft Bonanza

<li style = "list-style-type: circle">

Piper Cherokee

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

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

<html>

</head><body>

<h3> Name of subjects offered</h3>

<ul style = "list-style-type: square">

 web programming

 Data structures

 Compilers design

<h3> Name of subjects offered</h3>

<li style = "list-style-type: disc">

web programming

<li style = "list-style-type: square">

Data structures

<li style = "list-style-type: circle">

Compilers design

</body>

</html>

Name of subjects offered

- web programming
- Data structures
- Compilers design

Name of subjects offered

- web programming
- Data structures
- Compilers design

When ordered lists are nested, it is best to use different kinds of sequence values for the different levels of nesting. The list-style-type can be used to change the sequence values. Below table lists the different possibilities defined by CSS1. Property value Sequence type first four values
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
CSS2 has more, like lower-greek and hebrew

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Sequence types </title>
<style type = "text/css">
ol {list-style-type: upper-roman;}
ol ol {list-style-type: upper-alpha;}
ol ol ol {list-style-type: decimal;}
</style>
</head>body>
<h3> Aircraft Types </h3>
```

```
<ol>
<li> General Aviation (piston-driven engines)
<ol>
li> Single-Engine Aircraft
<ol>
<li> Tail wheel </li>
<li> Tricycle </li>
</ol>
</li>
<li> Dual-Engine Aircraft
<ol>
<li> Wing-mounted engines </li>
<li> Push-pull fuselage-mounted engines </li>
</ol>
</li>
</ol>
</li>
```

Aircraft Types

I. General Aviation (piston-driven engines)
    A. Single-Engine Aircraft
        1. Tail wheel
        2. Tricycle
    B. Dual-Engine Aircraft
        1. Wing-mounted engines
        2. Push-pull fuselage-mounted engines
II. Commercial Aviation (jet engines)
    A. Dual-Engine
        1. Wing-mounted engines
        2. Fuselage-mounted engines
    B. Tri-Engine
        1. Third engine in vertical stabilizer
        2. Third engine in fuselage

2.8 Colors

Colors are 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 larger set, the Web Palette 216 colors.Use hex color values of 00, 33, 66, 99, CC, and FF

2. Any one of 16 million different colors due to 24 bit color rep

3. 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

fuchia FF00FF aqua 00FFFF

Color properties

The color property specifies the foreground color of XHTML elements. For example, consider the following small table

<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.

2.9 Alignment of Text

The text-indent property allows indentation. Takes either a length or a % value. The textalign 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.

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

<head> <title> The float property </title>

<style type = "text/css">

img {float: right}

The Box Model

</style>

</head>

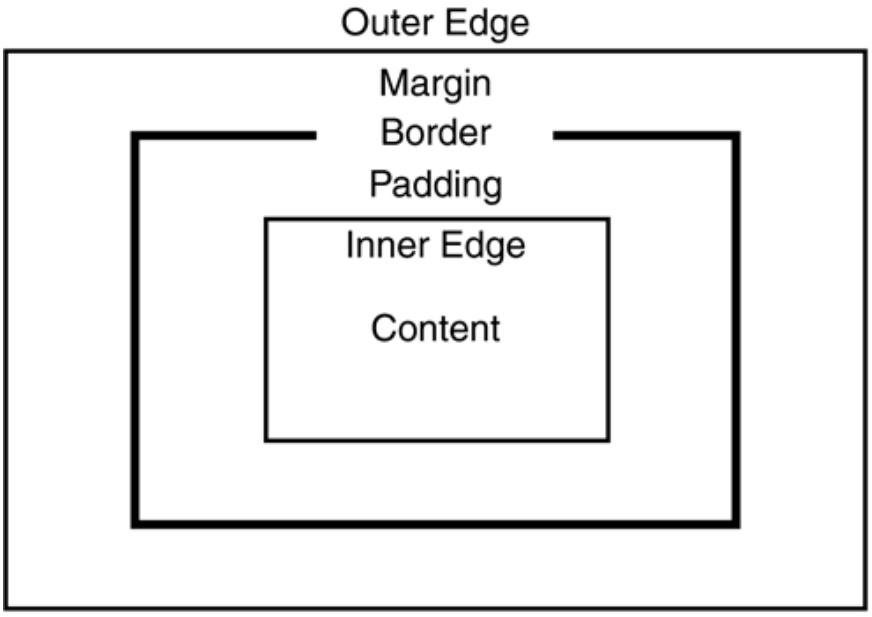
<body>

This is a picture of a Cessna 210. The 210 is the flagship single-engine Cessna aircraft. Although the 210 began as a four-place aircraft, it soon acquired a third row of seats, stretching it to a six-place plane. …. ……..

 </body>

</html>

2.10 The Box Model



Borders

Every element has a border-style property. It Controls whether the element has a border and if so, the style of the border. The styles of one of the four sides of an element can be set with 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) bordercolor – any color. Border color can be specified for any of the four borders (e.g., border-topcolor)

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Table borders </title>
<style type = "text/css">
table {border-top-width: medium;
border-bottom-width: thick;
border-top-color: red;
border-bottom-color: green;
border-top-style: dotted;
border-bottom-style: dashed;
```

```
}
p {border-style: dashed; border-width: thin;
border-color: green
} </style> </head>
<body> <table border = "5">
<caption> Diet chart </caption>
<tr>
<th> </th>
<th> Fruits </th>
<th> vegetables </th>
<th> Carbohydrates </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> <tr>
<th> Dinner </th>
<td> 0 </td>
<td> 0 </td>
<td> 1 </td>
</tr>
</table> <p>
If you strictly follow the chart you can easily lose weight.
</p> </body>
</html>
```

Diet chart

	Fruits	vegetables	Carbohydrates
Breakfast	0	1	0
Lunch	1	0	0
Dinner	0	0	1

If you strictly follow the chart you can easily lose weight.

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. bottom, left, or right

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

<head><title> Margins and Padding

</title><style type = "text/css">

p.one {margin: 0.2in;

padding: 0.2in;

background-color: #C0C0C0;

border-style: solid;

}

p.two {margin: 0.1in;

padding: 0.3in;

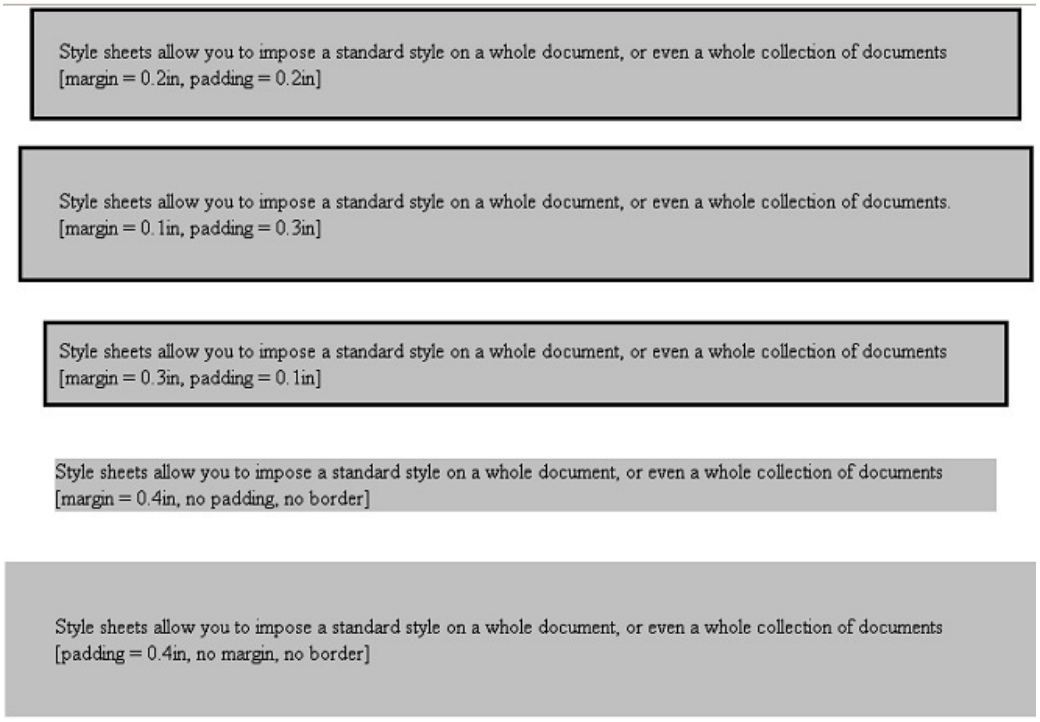
background-color: #C0C0C0;

border-style: solid;

}

p.three {margin: 0.3in; padding: 0.1in;

```
background-color: #C0C0C0;
border-style: solid; }
p.four {margin:0.4in;
background-color: #C0C0C0;}
p.five {padding: 0.4in;
background-color: #C0C0C0;
}
</style> </head>
</style> </head> <body>
<p> Here is the first line. </p>
<p class = "one">
Style sheets allow you to impose a standard style on a whole document, or even a whole
collection of documents <br /> [margin = 0.2in, padding = 0.2in]
</p>
<p class = "two">
Style sheets allow you to impose a standard style on a whole document, or even a whole
collection of documents. <br /> [margin = 0.1in,
padding = 0.3in]
</p>
<p class = "three">
Style sheets allow you to impose a standard style on a whole document, or even a whole
collection of documents <br /> [margin = 0.3in,
padding = 0.1in]
```



This is my last session.

2.11 Background Images

The background-image property is used to place an image in the background of an element Repetition can be controlled. Background image can be replicated to fill the area of the element. This is known as tiling 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.

```
<html xmlns = "http://www.w3.org/1999/xhtml">
<head> <title> Background images </title>
<style type = "text/css">
body {background-image: url(c172.gif);}
p {margin-left: 30px; margin-right: 30px;
margin-top: 50px; font-size: 14pt;}
</style>
</head>
<body> <p >
```

The Cessna 172 is the most common general aviation airplane in the world. It is an allmetal, single-engine position at sea level is 720 feet per minute.

</p> </body>

</html>

2.12 Conflict Resolution

When two or more rules apply to the same tag there are resolutions for deciding which rule applies. 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

h3{color:red;} body h3 {color: green;}

A tag may inherit a property and also be used as a selector. Style sheets can have different sources:

The browser itself may set some style

eg: In FX2 min font size can be set in Tools-Options-

Advanced window

The author of a document may specify styles. The user, through browser settings, may specify styles. Individual properties can be specified as important or normal.

Eg: p.special{font-style: italic !important; font-size :14}

This property is known as weight of a specification. Conflict resolution is a multistage sorting process. The first step in the process is to gather the style specifications from the three possible levels of style sheets. These specifications are sorted into order by the relative precedence of the style sheet levels. This is done according to the following rules, in which the first has the highest precedence. 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

Conflict resolution by Specificity (high to low)

1. id selectors
2. Class and pseudo-class selectors
3. Contextual selectors
4. General selectors

Position

Essentially, later has precedence over earlier. Most recently seen specification is the one which gets more precedence. Sorting process to resolve the style specification is known as cascade.

JAVASCRIPT

2.13 Overview of Javascript

JavaScript is a sequence of statements to be executed by the browser. It is most popular scripting language on the internet, and works in all major browsers, such as IE, FireFox, chrome, opera safari. Prequisite –HTML/XHTML

Origins

It is originally known as LiveScript, developed by Netscape. It become a joint venture of Netscape and Sun in 1995, and was renamed as JavaScript. It was standardized by the European computer Manufacturers Association as ECMA-262. ISO-16262. Current standard specifications can be found at

http://www.ecma-international.org/publications/standardsEcma-262.htm

Collections of JavaScript code scripts and not programs.

What is JavaScript?

1. JavaScript was designed to add interactivity to HTML pages.
2. JavaScript is a scripting language.
3. A scripting language is a lightweight programming language.
4. It is usually embedded directly into HTML pages.
5. JavaScript is an interpreted language (Scripts are executed without preliminary compilations)

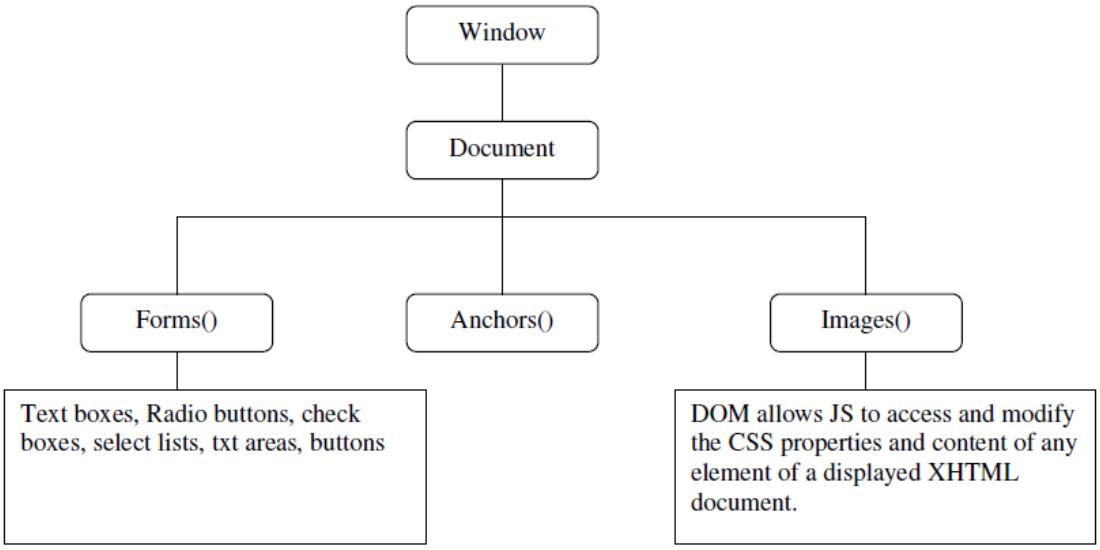
JavaScript can be divided into three parts.

1. The Core: It is a heart of the language, including its operators, expressions, statements and subprograms.

2. Client Side: It is a collection of objects that support control of a browser and interactions with users. Eg. With JavaScript an XHTML document can be made to be responsible to user inputs. Such as mouse clicks and keyboard use.

3. Server side: It is a collection of objects that make the language useful on a Web server. Eg. To support communication with a DBMS. Client side JavaScript is an XHTML embedded scripting language. We refer to every collection of JavaScript code as a script. An XHTML document can include any number of embedded scripts. The HTML Document Object Model(DOM) is the browsers view of an HTML page as an object hierarchy, starting with the browser window itself and moving deeper into the page, including of the elements on the page and their attribute.

Fig: The HTML DOM



The top level object is window. The document object is a child of window and all the objects that appear on the page are descendants of the document object. These objects can have children of their own. Eg. Form objects generally have several child objects , including textboxes, radio buttons and select menus.

JavaScript and java

Document Forms() Anchors() Images()

Window Text boxes, Radio buttons, check boxes, select lists, txt areas, buttons

DOM allows JS to access and modify the CSS properties and content of any element of a displayed XHTML document.

JavaScript and java is only related through syntax.

JavaScript support for OOP is different from that of Java.

JavaScript is dynamically typed.

Java is strongly typed language. Types are all known at compile time and operand types are checked for compatibility. But variables in JavaScript need not be declared and are dynamically typed, making compile time type checking impossible.

Objects in Java are static -> their collection of data number and methods is fixed at compile time.

JavaScript objects are dynamic: The number of data members and methods of an object can change during execution.

Uses of JavaScript

Goal of JavaScript is to provide programming capability at both server and the client ends of a Web connection. Client-side JavaScript is embedded in XHTML documents and is interpreted by the browser. This transfer of load from the often overloaded server to the normally under loaded client can obviously benefit all other clients. It cannot replace server side computations like file operations, database access, and networking.

JavaScript can be used as an alternative to Java applets. Java applets are downloaded separately from the XHTML documents that call them but JavaScript are integral part of XHTML document, so no secondary downloading is necessary. Java applets far better for graphics files scripts.

Interactions with users through form elements, such as buttons and menus, can be conveniently described in JavaScript. Because events such as button clicks and mouse movements are easily detected with JavaScript they can be used to trigger computations and provide feedback to the users.

Eg. When user moves the mouse cursor from a textbox, JavaScript can detect that movement and check the appropriateness of the text box's value. Even without forms, user interactions are both possible and simple to program. These interactions which take place in dialog windows include getting input from the user and allowing the user to make choices through buttons. It is also easy to generate new content in the browser display dynamically.

Event driven computation

Event driven computation means that the actions often are executed in response to actions often are executed in response to actions of the users of doc, actions like mouse clicks and form submissions. This type of computation supports user interactions through XHTML form elements on the client display. One of the common uses of JS is client end input data validation values entered by users will be checked before sending them to server for further processing. This becomes more efficient to perform input data checks and carry on this user dialog entirely on the client. This saves both server time and internet time.

Browsers and XHTML/JS documents.

It is an XHTML document does not include embedded scripts, the browser reads the lines of the document and renders its window according to the tags, attributes and content it finds when a JavaScript script is encountered in the doc, the browser uses its JS interpreter to execute the script. When the end of script reached, the browser goes back to reading the XHTML document and displaying its content.

JS scripts can appear in either part of an XHTML document, the head or the body, depending on the purpose of the script. Scripts that produce content only when requested or that react to user interactions are placed in the head of the document. -> Function definition and code associated with form elements such as buttons. Scripts that are to be interpreted just once, when the interpreter finds them are placed in the document body. Accordingly, the interpreter notes the existence of scripts that appear in the head of a document, but it does not interpret them while processing the head. Scripts that are found in the body of a document are interpreted as they are found.

2.14 Object orientation and Javascript

JavaScript is object based language. It doesn't have classes. Its objects serve both as objects and as models of objects. JavaScript does not support class based inheritance as is supported in OO language. CTT-Java. But it supports prototype based inheritance i.e a technique that can be used to simulate some of the aspects of inheritance. JavaScript does not support polymorphism. A polymorphic variable can reference related objects of different classes within the same class hierarchy. A method call through such a polymorphic variable can be dynamically bound to the method in the objects class.

JavaScript Objects

JavaScript objects are collection of prospectus, which corresponds to the members of classes in Java & C++. Each property is either a data property or a function or method property.

1. Data Properties

a. Primitive Values (Non object Types)

b. Reference to other objects

2. Method Properties –methods.

Primitives are non object types and are used as they can be implemented directly in hardware resulting in faster operations on their values. These are accessed directly-like scalar types in java & C++ called value types. All objects in a JavaScript programs are directly accessed through variables. Such a variable is like a reference in java. The properties of an object are referenced by attaching the name of the property to the variable that references the object. Eg. If myCar variable referencing an object that has the property engine, the engine property can be referenced with myCar.engine.

The root object in JavaScript is object. It is ancestor through prototype inheritance, of all objects. Object is most generic of all objects, having some methods but no data properties. All other objects are specializations of object, and all inherit its methods.

JavaScript object appears both internally and externally as a list of property/value pairs. Properties are names values are data values of functions. All functions are objects and are referenced through variables. The collection of properties of JavaScript is dynamic –Properties can be added or deleted at any time.

2.15 General syntactic Characteristics

1. JavaScript are embedded either directly or indirectly in XHTML documents.
2. Scripts can appear directly as the content of a <script> tag.
3. The type attribute of <script> must be set to “text/JavaScript”.
4. The JavaScript can be indirectly embedded in an XHTML document using the src attribute of a <script> tag, whose value is name of a file that contains the script.

Eg. <script type=”text/JavaScript” src=”tst_number.js”>
</script>

Closing tag is required even if script element has src attribute included.

The indirect method of embedding JavaScript in XHTML has advantages of

1) Hiding the script from the browser user.
2) It also avoids the problem of hiding scripts from older browsers.
3) It is good to separate the computation provided by JavaScript from the layout and presentation provided by XHTML and CSS respectively. But it is sometimes not convenient and cumbersome to place all JavaScript code in separate file JavaScript identifiers or names are similar to programming languages.

1. must begin with (-), or a letter. Subsequent characters may be letters, underscores or digits.

2. No length limitations for identifiers.

3. Case sensitive

4. No uppercase letters.

Reserved words are break delete function return typeof case do if switch var catch else in this void continue finally instanceof throw while default for new try with

JavaScript has large collection of predefined words

alert

open

java

self

Comments in JavaScript

// - Single line

/* */ -Multiple line

Two issues regarding embedding JavaScript in XHTML documents.

1) There are some browsers still in use that recognize the <script> tag but do not have JS interpreters. These browsers will ignore the contents of the script element and cause no problems.

2) There are still a few browsers in use that are so old they do not recognize <script> tag. These browsers will display the contents of the script elements as if it were just text. Therefore it has been customary to enclose the contents of all script elements in XHTML comments to avoid this problem. XHTML validator also has a problem with embedded JS. When embedded JS happens to include recognizable tags.

For eg
 in output of JS-they often cause validation errors.

Therefore we have to enclose embedded JS in XHTML comments. XHTML comment introduction (<! - -) works as a hiding prelude to JS code. Syntax for closing a comment that encloses JS code is different. It is usual XHTML comment closer but it must be on its own line and preceeded by two slashes.

Eg. <!—

-- JS ---

//-->

Many more problem are associated with putting embedded JavaScript in comments in XHTML document.

Solution : Put JavaScript scripts of significant style in separate files.

Use of ; in JS is unusual

When EOL coincides with end of statement, the interpreter effectively insects a semicolon there, but this leads to problems.

Eg. return x;

Interpreter puts; after return making x an illegal orphan.

Therefore put JS statements on its own line when possible and terminate each statement with a semicolon. If stmt does not fit in one line, break the stmt at a place that will ensure that the first line does not have the form of a complete statement.

<?xml version = “1.0 encoding = “utf-8”?>

<!DOCTYPE html PUBLIC “-//w3c//DTD XHTML 1.1//EN”

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

<! - -hello.html

8

A trivial hello world example of XHTML/JavaScript

- ->

<html xmlns = “http://www.w3.org/1999/xhtml”.>

<head>

<title> Hello World</title>

</head>

<body>

<script type = “text/javascript”>

<!- -

Document.write(“Hello, fellow Web programmers!”);

//- ->

</script>

</body>

</html>

2.16 Primitives, operations, and expressions

The primitive data types, operations and expressions of JavaScript.

Primitive Types:

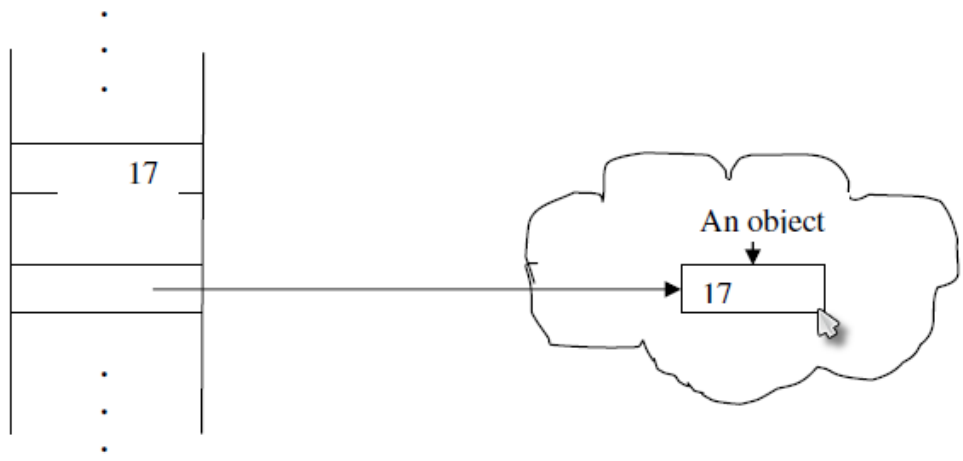
Pure primitive types: Number, String, Boolean, Undefined and null. JavaScript includes predefined objects that are closely related to the number, string and Boolean types named number, string and Boolean. These are wrapper objects. Each contains a property that stores a value of the corresponding primitive type. The purpose of the wrapper object is to provide properties and methods that are convenient for use with values of the primitive types.

In case of numbers: Properties are more useful.

In case of string: Methods are more useful.

Because JavaScript coerces values between the number type and number objects and between the string type and objects, the methods of number and string can be used on variables of the corresponding primitive types.

Fig:



Prim is a primitive variable with value 17 and obj is a number object whose property value is 17. Fig shows how they are stored.

Numeric and String literals:

All numeric literals are values of type number. The numeric values of JavaScript are represented internally in double precision floating point form, Numeric values in JavaScript are called

numbers because of single numeric data type. Literal numbers in a script can have forms of either integers or floating point values. Integer literals are strings of digits.

Floating point literals can have decimal points or exponents or both.

Legal numeric literals: 72, 7.2, .72, 72, 7E2, 7e2, .7e2, 7.e2, 7.2E-2.

Integers in Hexadecimal form 0x or 0X.String Literal: Sequence of 0 or more characters delimited by either single quotes or double quotes. They can include characters specified with escape sequences, such as \n and \t. If you want an actual single quote character in a string literal that is delimited by single quotes, embedded single quote must be preceded by a backslash.

'You\' re the most freckly person I\'ve ever met'

"D:\\bookfiles" ->Jo embed\

' ' or " " -> Null string