SUB: WEB TECNOLOGY

Lecturer: Durganand Panjiyar

Qualification: Master of Computer Science

LINCOLN UNIVERSITY COLLEGE

LECTURE -1:

WWW: refers to the World Wide Web. The World Wide Web consists of all the public Web sites connected to the Internet worldwide, including the client devices that access Web content. The WWW is just one of many applications of the Internet and computer networks.

The World Web is based on these technologies:

1. HTML - Hypertext Markup Language.
2. HTTP - Hypertext Transfer Protocol.
3. Web servers and Web browsers.

Tim Berners-Lee led the development of the original World Wide Web in the late 1980s and early 1990s. He helped to build prototypes of the above Web technologies. Web sites and Web browsing exploded in popularity during the mid-1990s.

HTTP - the Hypertext Transfer Protocol - provides a standard for Web browsers and servers to communicate. The definition of HTTP is a technical specification of a network protocol that software must implement.

HTTP is an application layer network protocol built on top of TCP. HTTP clients and servers communicate via HTTP request and response messages. The three main HTTP message types are GET, POST, and HEAD.

HTTP utilizes TCP port 80 by default, though other ports such as 8080 can alternatively be used.

The current version of HTTP in widespread use - HTTP version 1.1 - was developed to address some of the performance limitations of the original version - HTTP 1.0. HTTP1.1

URL: Uniform Resource Locator or URL. The Internet address is a form of URI and standardized naming convention for addressing documents accessible over the Internet.

http://	www.	ltsdevelop.com	website	index.html
Protocol	Subdomain	Domain and domain suffix	Directories	Web page

INTERNET BROWSER: A web browser is a software application which enables a user to display and interact with text, images, videos, music, and other information that could be on a website. Text and images on a web page can contain hyperlinks to other web pages at the same and different websites. Web browsers allow a user to make quickly and easily access information provided on many web pages at many websites by traversing these links. Web browsers format HTML information for display so the appearance of a web pages many differs between browsers.

WEB TECHNOLOGY:

Web technology is the development of the mechanism that allows two of more computer devices to communicate over a network. For instance, in a typical office setting, a number of computers plus additional devices such as printers may be interconnected via a network, allowing for quick and convenient transmission of information. The processes involved in web technology are complex and diverse, that is why major businesses employ whole departments to deal with the issue. Web technology has revolutionized communication methods and has made operations far more efficient.

ADVANTAGES OF WEB TECHNOLOGY: The main advantage of web technology is that it offers convenience and a high speed of communication in the computer world. Whether in the office or the home, processes using a computer are more swift and straightforward with the use of a network. Web technology allows messages to be sent around a system, whereas before it may have been necessary to employ a runner or leave your workspace to communicate a message. It is clear to see how web technology reduces costs and makes a company more efficient, raising business potential.

DISADVANTAGES OF WEB TECHNOLOGY: Matters involving web technology can be very complicated, and it would be difficult for someone without relevant experience to sort a network problem out. This means it is necessary to employ someone with the specific skills to solve network issues, which costs money. Additionally, the existence of a network provides the opportunity for an attack on the computer system. Weaknesses in a network could be exploited; important information could be stolen or destroyed and malware could infect the various network systems. For this reason, network security is another issue that must be considered when using web technology.

MARKUP LANGUAGES:

Markup is used to in text and word processing documents to describe how a document should look when displayed and printed. The Internet uses markup to define how Web pages should look when displayed in a browser or to define the data contained within a Web document.

There are many different types of markup languages. For example: Rich Text Formatting (RTF) is a markup language that word processors use. This section describes the most common markup languages that are used on the Internet.

HTML:

HTML stands for Hypertext Markup Language. HTML is the primary markup language that is used for Web pages. HTML tells the browser what to display on a page. For example, it specifies text, images, and other objects and can also specify the appearance of text, such as bold or italic text. HTML was created by BERNERS who was the inventor of the WWW.

HTML CODES IN EDITOR:

In the HTML editor, we can edit the HTML codes on the following method:

```
<html>
<head>
<title> advance web</title>
</head>

<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>
<img src=”path”>

</body>
</html>
```

HTML is the markup language that is used to write web pages. It simply describes a web page's content and its structure.

HTML tags: HTML uses tags which are written in pairs of angle brackets, like this <tag>.

HTML Example:

 this is a paragraph. Everything between these two tags is part of this paragraph.

Standalone HTML: The main basic standalone HTML are:

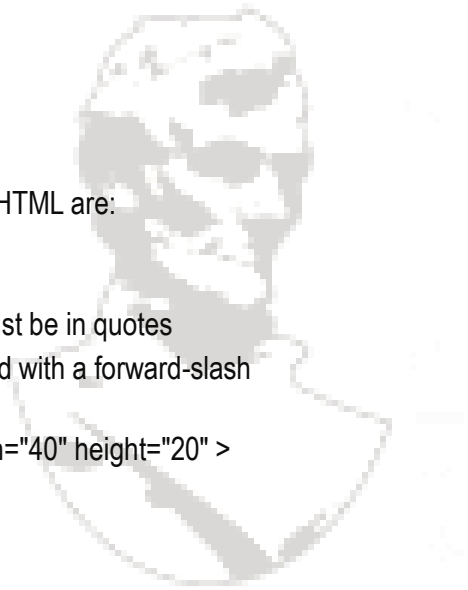
: Line-break.

<hr>: A horizontal rule.

<img>: An image.

The main things you know about HTML are:

1. Tags must be lowercase
2. All parameters in tags must be in quotes
3. Self-closing tags must end with a forward-slash



<img src="background.jpeg" width="40" height="20" >

FORMATTING IN HTML:

<h1> … </h1>, <h2> … </h2>, <h3> … </h3>

Headings are used for marking section headings in your web pages.

 …

Here is where the paragraph starts" and "Here is the end of the paragraph.

 Inserts a new line.

<Hr>

Horizontal rule

 …

This is one of the most common tags. Basically, a div is a general-purpose box that takes up the full width of the space it's in.

...

This is another general-purpose container. Whereas a div is a blocky full-width box, this quietly wraps round anything it contains.

...

A section of text that you want to be stressed. Normally rendered as bold in web browsers.

 ...

Emphasized text – normally rendered in italics in web browsers.

PRACTICAL SESSION-1:

```
<html>
<head>
<title>MySite< /title>
</head>
<body bgcolor="yellow" TEXT="red">
<h1>
<center>
WELCOME TO MY WEBSITE
</center>
<hr color="red" size="2%" width="50%">
<A HREF="CONTACT.HTML">CONTACT< /A>
<hr color="red" size="2%" width="100%">
<img src="image/2.jpg" width="30%" height="40%">
<p align="center">
<img src="image/1.jpg" width="10%" height="15%" align="left">
<img src="image/4.jpg" width="10%" height="15%" align="right">
<font color="green" size="4">
```

```
<b>On the Insert tab, the galleries include items< /b>
<u>that are designed to coordinate with the overall < /u>
<i>look of your document. You can < /i>use these galleries
<hr color="red" size="2%" width="50%">
look for the selected text from the Quick Styles gallery
on the Home tab.< /font>
</p>
</body>
</html>
```

NOTE:

1. The **DOCTYPE** declaration defines the document type
2. The text between **<html>** and **</html>** describes the web document
3. The text between **<body>** and **</body>** describes the visible page content
4. The text between **<h1>** and **</h1>** describes a heading
5. The text between ** ** and ** ** describes paragraph

HTML PAGE STRUCTURE

This is the ways, that HTML visualization page structure:

```
<html>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

LECTURE -2:

HTML ATTRIBUTE: HTML attributes are generally classed as required attributes, optional attributes, standard attributes, and event attributes. Usually the required and optional attributes modify specific HTML elements, while the standard attributes can be applied to most HTML.

REQUIRED AND OPTIONAL HTML ATTRIBUTES:

Used by one tag:

1. <applet>: code, object
2. <area>: nohref
3. <body>: alink, background, link, text, vlink
4. <dir>: dir
5. <form>: accept-charset, action, enctype, method
6. <frame>: noresize
7. <head>: profile
8. <hr>: noshade
9. <html>: xmlns
10. <img>: ismap
11. <input>: checked, maxlength
12. <label>: for
13. <meta>: content, http-equiv, scheme
14. <object>: classid, codetag, data, declare, standby
15. : start
16. <option>: selected
17. <param>: valuetype
18. <script>: defer, xml:space
19. <select>: multiple
20. <table>: cellpadding, cellspacing, frame, rules, summary
21. <td>: headers

HML FORM

The HTML of a form consists of a form tag enclosing tags for various form controls, including text entry fields, various buttons, a file selection control, and menus. The user types in the text fields, clicks on the

Buttons and selects items from the menus. Finally, the user submits the form, usually by clicking on a special submit button.

The exact action that follows from submission is specified by the form tag. Generally, the user's browser gathers the settings from all the controls in the form, and sends the collected information to the site specified by the form tag, then waits for the server to respond. The response is in the form of a new HTML document, which the browser then displays to the user.

A single web page may contain multiple forms. However, the information that is submitted to the server comes from the form whose "submit" button was clicked.

PARTS OF THE FORM

Text area		<text area rows="2" cols="20">Multiple lines </text area>
Text field	hello	<input type="text">
Password field		<input type="password">
Button		<input type="button" value="Push Me">
Radio Button		<input type="radio">
Check box		<input type="checkbox">
Submit button	OK	<input type="submit" value="OK">
File Selection		<input type="file" value="Browse...">
Menu	item c	<select> <option>item a</option> <option>item b</option> <option selected="selected">item c</option> </select>
Label	a Label	<label>a Label</label>

LAYOUT WITH CSS:

By default HTML <DIV> sections will fill 100% of the web page width and stack up one on top of the other. It is highly unlikely that this is how we want our page elements to be arranged. We want some sections arranged side by side and using various widths. For example we might want to have a main content area with a menu down the left-hand side. CSS makes this a simple task. The first step is to understand the float property.

CSS FLOAT:

The float property tells HTML elements how to arrange themselves in relation to the DIVS around them. We can specify one of two values for the float property - left or right. If we tell a to float to the left, then it will shift itself as far left on the line as it can go before bumping into another section
Let's take a look at a working example:

```
<div id="header">Section header</div>
<div id="menu">Menu here</div>
<div id="content"> content here</div>
<div id="footer"> footer</div>
```

```
#menu {
        Float: left;
}
#content {
        Float: left;
}
```

DYNAMIC LAYOUT

Dynamic Layout is a simple JavaScript library that allows you to easily adjust page layout based on the current browser width. The script works by modifying the class property on the body element, adding a new class that will look something like bw-1000, where "1000" is one of the numbers in a predefined list of possible browser widths.

By default, the script uses three window sizes: 800px, 1000px, and 1200px. Dynamic layout chooses the largest possible size that fits within the window. If the window width is smaller than the smallest specified width, the bw-mini value will be used as the class name.

USAGE

```
<script src="http://static.fortes.com/projects/dynamiclayout/dynamiclayout-1.0.min.js"
type="text/javascript"></script>
```

Make sure you place this line of code right after the <body>tag.

You can tell the script to use custom screen widths by setting the sizes query string property to a comma delimited list of numbers. For example, to set the window sizes to 600, 800, and 1200 you would use the following code:

```
<script src=http://static.fortes.com/projects/dynamiclayout/dynamiclayout-1.0.min.js?sizes=600,800,1200
type="text/javascript"></script>
```

CSS rules that depend upon the screen size. To do so, make sure you use CSS selectors based on the browser width example:

```
#content
{Padding: .5em;
}
.bw-800 #content
{
  Margin: 0 auto;
  Width: 600px;
}
.bw-1000 #content
{
  Margin: 0 auto;
  Width: 800px;
  Line-height: 2 ;}
```

BASIC STRUCTURE OF A WEB PAGE

The Doc type

The first item to appear in the source code of a web page is the doc type declaration. This provides the web browser with information about the type of markup language in which the page is written. It may or may not affect the way the browser renders the content. It may look a little scary at first glance, but the good news is that most WYSIWYG web editors will create the doc type for you automatically after you have selected from a dialog the type of document you are creating.

The doc type looks like this:

```
<Html>
  <Head>
    <Title>Page title</title>
  </head>
  <Body>
  </body>
</html>
```

THE DOCUMENT TREE

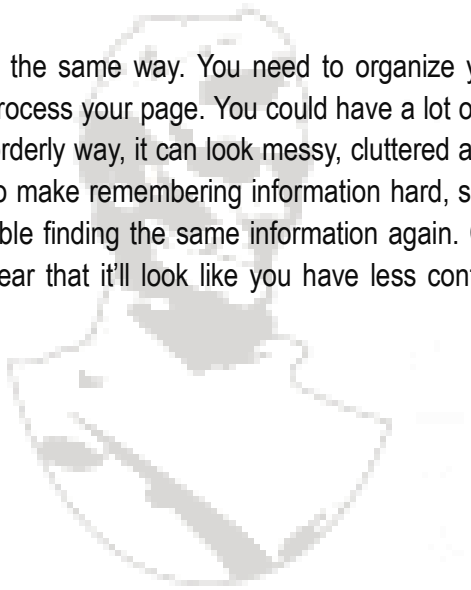
A web page could be considered as a document tree that can contain any number of branches. There are rules as to what items each branch can contain. To understand the concept of a document tree, it is useful to consider a simple web page with typical content features alongside its tree view.

ABOVE ONE IS DOCUMENT TREE TAKEN PRINT SCREEN.

CHUNKING CONTENT IN TO CLEAR, SCAN ABLE WEB PAGES

Every website has content. But not every website has content that's easy to process and understand. That is because not every website uses content chunking to its advantage. Content chunking is a technique of combining pieces of content into sizable chunks, so that it's easy and efficient for users to consume. If your website doesn't use content chunking, you will make users work harder than they need to consume your content. This can cause users to miss important information, struggle to find specific information and eventually leave your site because of a poor reading experience.

CREATING CONTENT CHUNKS



The content on a website works the same way. You need to organize your content into chunks, so that users can visually and mentally process your page. You could have a lot of content you want to show. But if it's not organized in a clear and orderly way, it can look messy, cluttered and make it hard for users to scan and read your content. It can also make remembering information hard, so that when users leave your site and come back, they'll have trouble finding the same information again. Content chunking can make your web pages look so clean and clear that it'll look like you have less content, when you actually have the same amount.

LECTURE -3

INTRODUCTION TO INTERFACE DESIGN:

Interface design is intended to give an overview of the concepts that will be covered in the larger course. In developing interactive systems, a designer is tasked with helping to define an experience. Before beginning, there are important questions to explore: Who are the target users? What needs do they have? How does one encourage exploration, learning, and accessibility for users through interfaces? Are there guidelines that should be followed? What experiences have target audiences had before? This course explores these issues in relation to different expressions of interface design. It also devotes a good deal of time to exploring usability principles and concepts on which the class can base its expressions.

CSS SELECTORS AND DESIGN:

In CSS, pattern matching rules determine which style rules apply to elements in the document these patterns, called selectors. It may range from simple element names to rich contextual patterns. The elements of the document tree that match a selector are called *subjects* of the selector. A selector consisting of a single simple selector matches any element satisfying its requirements.

ATTRIBUTES:

All HTML elements can have associated properties, called attributes. These attributes generally have values. Any number of attribute and value pairs can be used in an element's tag - as long as they are separated by spaces. They may appear in any order.

ATTRIBUTE SELECTORS: Attribute selectors are used to select elements based on their attributes and attribute value. For example, we may want to select any image on an HTML page that is called "st.jpg". This could be done with the rule below that will only target images with the chosen name:

UNIVERSAL SELECTOR: The universal selector written with "*" matches the name of any element type. It matches any single element in the document tree.

1. *.warning and .warning are equivalent.
2. *#m and #m are equivalent.

TYPE SELECTORS: A type selector matches the name of a document language element type. A type selector matches every instance of the element type in the document tree.

Ex: h1 {font-family: sans-serif}

CHILD SELECTORS: A child selector matches when an element is the child of some element. A child selector is made up of two and more selectors separated by ">".

The following rule sets the style of all P elements that are children of BODY:

```
body > P { line-height: 1.3 }
```

The following example combines descendant selectors and child selectors:

```
div ol>li p
```

It matches a P element that is a descendant of an LI; the LI element must be the child of an OL element; the OL element must be a descendant of a DIV.

ADJACENT SIBLING SELECTORS: Adjacent sibling selectors have the following syntax: E1 + E2, where E2 is the subject of the selector. The selector matches if E1 and E2 share the same parent in the document tree and E1 immediately precedes E2, ignoring non-element nodes.

Thus, the following rule states that when a P element immediately follows elements, it should not be indented: Ex:

```
math + p { text-indent: 0 }
h1 + h2 {margin-top: -5mm}
h1.opener + h2 {margin-top: -5mm}
```

CLASS SELECTORS: Working with HTML, We may use the period (.) notation as an alternative to the ~= notation when representing the class attribute.

Thus, for HTML, div.value and div[class~=value] have the same meaning. The attribute value must immediately follow the "period" (.). UAs may apply selectors using the period (.) notation in XML documents if the UA has namespace specific knowledge that allows it to determine which attribute is the "class" attribute for the respective namespace. One such example of namespace specific knowledge is the prose in the specification for a particular namespace.

```
*.pastoral {color: green};
```

Or

```
.pastoral {color: green}
```

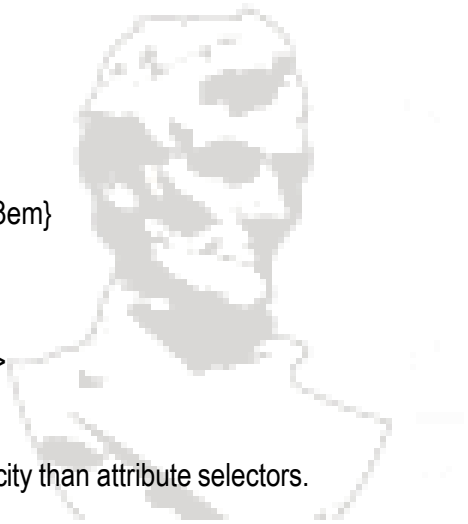
ID SELECTORS: The document languages may contain attributes that are declared to be of type ID. What makes attributes of type ID special is that no two such attributes can have the same value, whatever the document language, an ID attribute can be used to uniquely identify its element.

In HTML all ID attributes are named "id"; XML applications may name ID attributes differently, but the same restriction applies.

The ID attribute of a document language allows authors to assign an identifier to one element instance in the document tree. CSS ID selectors match an element instance based on its identifier. A CSS ID selector contains a "#" immediately followed by the ID value, which must be an identifier.

Like:

```
<HEAD>
  <TITLE>Match P</TITLE>
  <STYLE type="text/css">
    *#z98y {letter-spacing: 0.3em}
  </STYLE>
</HEAD>
<BODY>
   <P id=z98y>Wide text</P>
</BODY>
```



ID selectors have a higher specificity than attribute selectors.

The link pseudo-classes: link and: visited: User agents commonly display unvisited links differently from previously visited ones. CSS provides the pseudo-classes ': link' and ': visited' to distinguish them:

- The: link pseudo-class applies for links that have not yet been visited.
- The: visited pseudo-class applies once the link has been visited by the user.

It may return a visited link to the (unvisited) ': link' state at some point.

The two states are mutually exclusive.

The document language determines which elements are hyperlink source anchors. For example, in HTML4, the link pseudo-classes apply to elements with a "href" attribute. Thus, the following two CSS declarations have similar effect:

```
a: link { color: red }
: link {color: red }
```

<A class="external" href="http://out.side/">external link</A>
a: visited {color: blue} will cause it to be blue.

The dynamic pseudo-classes: hover, active, and: focus

WEB NAVIGATION:

Web navigation is a process of navigating information resources in the World Wide Web. The user interface that is used to do so is called a web browser. A central theme in web design is the development of a web navigation interface that maximizes usability. A website is overall navigational scheme includes several navigational pieces such as global, local, supplemental, and contextual navigation. All of these are vital aspects of the broad topic of web navigation. Hierarchal navigation systems are vital as well since it is the primary navigation system. It allows for the user to navigate within the site using levels alone, which is often seen as restricting and requires additional navigation systems to better structure the website. The global navigation of a website is as another segment of web navigation. All these navigational pieces fall under the categories of various types of web navigation.

There are many different types of website navigation:

HIERARCHICAL WEBSITE NAVIGATION:

The structure of the website navigation is built from general to specific. This provides a clear, simple path to all the web pages from anywhere on the website.

GLOBAL WEBSITE NAVIGATION

Global website navigation shows the top level sections and pages of the website. It is available on each page and lists the main content pages of the website.

LOCAL WEBSITE NAVIGATION

Local navigation would the links with the text of your web pages, linking to other pages within the website.

STYLE OF WEB NAVIGATION:

The availability of different navigational styles allows for the information in the website to be delivered easily and directly. This also differentiates between categories and the sites themselves to indicate what the vital information is and to enable the user's access to more information and facts discussed within the website

Different styles of web navigation are present

TEXT LINKS

Link title is the visible and clickable text in a hyperlink.

NAVIGATION BAR

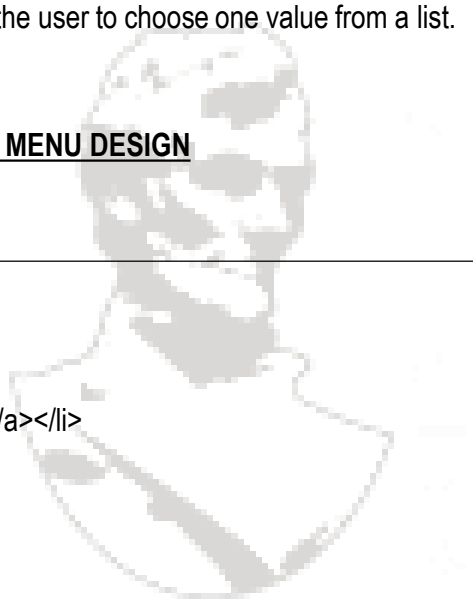
A navigation bar is a section of a website that online page intended to aide visitors in travelling through the online document.

SITEMAP

A site map is a list of pages of a web site accessible to users. It can be either a document in any form used as a planning tool for Web design that lists the pages on a site, typically organized in hierarchical fashion.

DROPDOWN MENU

In computing with graphical user interfaces, a dropdown menu list is a user interface control GUI element, similar to a list box, which allows the user to choose one value from a list.



PRACTICAL OF DROUP DOWN MENU DESIGN

Menu.html

```
<div id ="navmenu">
<ul>
<li><a href="home.php">HOME</a></li>
<li><a href="#">SERVICES</a>
<ul>
<li><a href="register.php">NEW REGISTER</a></li>
<li><a href="service.php">ONLINE HELP</a></li>
<li><a href="predict.php">ISSUES REPORT</a></li>
</ul>
</li>
<li><a href="contact.php">CONTACT</a></li>
</ul>
</div>
```

```
border-width:1px;
background:red;}

#navMenu ul ul

{position:absolute;
visibility:hidden;

top:30px;}

#navMenu ul li:hover ul{

visibility:visible;

z-index: 99999!important;}
```