BACHELOR OF COMPUTER APPLICATIONS

SEMESTER *5*

DCA3101

WEB DESIGN

Unit 4

HTML – URI, LIST, Hyper Links

Table of Contents

1. INTRODUCTION

In a previous unit, you learnt about internet services, domain names and IP addressing. With the help of domain names, you can map the computer names and IP address which will greatly improve the usability of the internet and give you quick access to web pages. In this unit, you will learn how to develop web pages using HTML. A lot of information flows from one computer to another over the web, but how this information is displayed on your computer? How does your computer know how to show the information?

Obviously the information should be accompanied with some "commands" and "instructions" that would tell your computer or more specifically Internet Explorer, Netscape, Opera, Hot Java or any one of the other web browsers, the structure of the information and how to format this information, HTML is one of the most popular used languages for information being exchanged between computers. In this unit you shall study about the various HTML tags and how to create a web page using these tags. You shall also study how to design a hyper linking pages using HTML.

1.1 Objectives:

After studying this unit, you should be able to:

- *Define markup language and evolution of HTML*
- *Structure the web page using HTML*
- *Explain hypertext and navigations*
- *Describe list form*

1.2 What is Markup Language?

A markup language is a language that annotates text so that the computer can manipulate the text. Most markup languages are human readable because the annotations are written in a way to distinguish them the text.

Markup languages are designed for the processing, definition and presentation of text. The language specifies code for formatting both the layout and style, within a text file, the code

used to specify the formatting is called **tags**. In recent year's number of markup languages have been developed, some examples of widely known markup languages are:

- Standard Generalized Markup Language (SGML)
- Hypertext Markup Language (HTML)
- Extensible Markup Language (XML)
- **Standard Generalized Markup Language (SGML)**

  SGML was developed and standardized by ISO (International Organization for Standards) in 1986. SGML is a language for recording and storing document information. In general you use SGML to write rules that a group of related documents should follow when they store your desire added value. The general process of figuring out the rules is called modeling, and when you're done modeling and expressing the model in SGML form, your set of rules serve as the "language" that these documents use to "speak" computer. In SGML terminology, such a group of documents is described by the term *document type*, and rule set for a document type is called a *document type definition*, or DTD.

- **Hypertext Markup Language (HTML)**

  HTML stands for Hyper Text Markup Language.

  Hyper: HTML allows non-linear access to information. Anyone can go anywhere on the World Wide Web, whenever he/she wants to.

  Text: The HTML code is simple text.

  Markup: To indicate to the browser, the structure and format of the information, tags are used to mark the information, i.e. the data is marked.

  Language: It is a language.

  HTML has changed the way we access information. If you want to read a book, you read it from the start to the end. It is a linear process. You cannot skip and move around in the text for the information you want very easily.

Moreover you access the information from one source at a time. HTML has changed all that. It allows one to jump to different parts of the information source, move from one source to another, with just the click of a button. So the user can access the information in the manner he/she wants to.

- **Extensible Markup Language (XML)**

  Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. Both XML and HTML contains markup symbols describe the contents of page or file. XML is "extensible" because, unlike HTML, the markup symbols are unlimited and self-defining.

  HTML, however, describes the content of web page (mainly text and graphical images) only in terms of how it is to be displayed and interacted, XML describes the content in terms of what date is being described.

1.3 History of HTML

HTML was originally developed by Tim Berners-Lee while at CERN, and popularized by the Mosaic browser developed at NCSA. During the course of the 1990s it has blossomed with the explosive growth of the Web. During this time, HTML has been extended in a number of ways. The Web depends on Web page authors and vendors sharing the same conventions for HTML. This has motivated joint work on specifications for HTML. It is a common way of representing hypertext documents on the web. The first public release was in 1990, named as HTML 0.0, then followed by the next version HTML 1.0 in 1993, HTML 2.0 in 1994. There are other versions present see below table.

Table 4.1 different versions of HTML

Version	Year
HTML 0.0	1991
HTML 1+	1993
HTML 2.0	1995
HTML 3.2	1997
HTML 4.0	1999
XHTML 1.0	2000
XHTML 5	2013

1.4 Introduction to URI (Uniform Resource Identifier)

Using the Web means having your browser act as a client program on your behalf. In order to fulfill your requests, your browser will contact a server, and ask for either some information or a service of some type.

Two types of URIs are there, Universal Resource Locator (URL) and the Universal Resource Name (URN).

- **URL (Uniform Resource Locator)**

  URLs provides a standard way to specify the exact location and name of just about any Internet resource. URLs are location dependent and contain four distinct parts: the protocol type, the machine name, the directory path and the file name. Different URL's are there, namely: file URLs, FTP URLs, Gopher URLs, News URLs, and HTTP URLs. URLs may be relative to a directory or offsets into a document. All URLs have the same general format:

  Scheme: object-address

  The scheme is often a communications protocol. Common schemes include http, ftp, gopher, telnet, file, mailto, and news. Different schemes use different object addresses that have different forms

  In general, most URLs have one of two common formats:

1. Scheme: //hostname/description
2. Scheme: description

Example 1: http://www.alan.com/afan

This example describes a particular web page on a particular computer. The URL begins with a name, indicating a specific type of resource.

Example 2: news.rec.human

This example describes a more general resource. The scheme is news, which indicates a Usenet discussion group.

- **URN (Uniform Resource Name)**

  A URN (Uniform Resource Name) is an internet resource with a name that, unlike URL, has persistent significance- that is, the owner of the URN can expect the someone else (or program) will always be able to find the resource.

  A URN is the historical name for a URL that uses the

  Urn: scheme

A URN looks something like URL. For example, below a hypothetical URN:

Urn: def//blue_laser

Where def:// might indicate an agency or an accessible directory of all dictionaries, glossaries, and encyclopedias on the internet and blue_laser is name of the term.

- In HTML, URI used for:
  1. Link to another document or resource (see Hyperlink concept in next sections)
  2. Link to an external style sheet or script (see Link and Script elements in next sections)
  3. Include an image, object, in page (see Image concepts in next unit)
  4. Submit for form (see Form concept)
  5. Create a frame document (see Frame Concept in next unit)

1.5 Fragment Identifiers

Some URIs refers to a location within a resource. This kind of URI ends with "#" followed by an anchor identifier (called the fragment identifier).

Example: http://somesite.com/html/top.html # section_2

In above example URI pointing to an anchor named Section_2. Here # section_2 is fragment identifier.

SELF-ASSESSMENT QUESTIONS - 1

1. SGML stands for _____ .
2. Both XML and HTML contains symbols describe the contents of page or file.
3. provides a standard way to specify the exact location and name of just about any Internet resource.

2. STRUCTURING WEB PAGE USING HTML

A markup language like HTML is simply a collection of codes/elements tags that indicate the structure and format of a web document.

How do you instruct the browser to display the page the way you want it to be displayed? Tags are the mechanisms for attaining this end. The various tags that you put in the html document will serve as commands to the browser, which will display the contents of the page accordingly.

All tags begin with a less than symbol < and end with a greater than symbol

>. Within these two symbols you enter your tag, your command to the browser.

If you have started a "command" called tagname with **<tagname>,** you close it with **</tagname>.** The ending tag is the same as the starting tag.

Each web page begins and ends with a special command: **<HTML>** and the ending **</HTML>.** These commands tell the browser that everything in- between is the web page.

Inside the web page, there are two basic parts: the **HEAD** and the **BODY**.

HEAD – The HEAD is used to give the title of the web page (visible in the Title Bar). It can also contain information about the web page, such as keywords, what the page is for, where people will go after this page, and so on. The HEAD tag can embed the following tags within it, **TITLE, BASE, ISINDEX, META, SCRIPT, STYLE, and LINK tags** in this unit, we will use it just for the TITLE of the web page.

BODY – The BODY is where the real web page is. That's the area which holds the information which we think of as the "web page." The BODY area of the web page begins and ends with the commands <BODY> and

</BODY>.

Here is how the web page looks with only the basic structure:

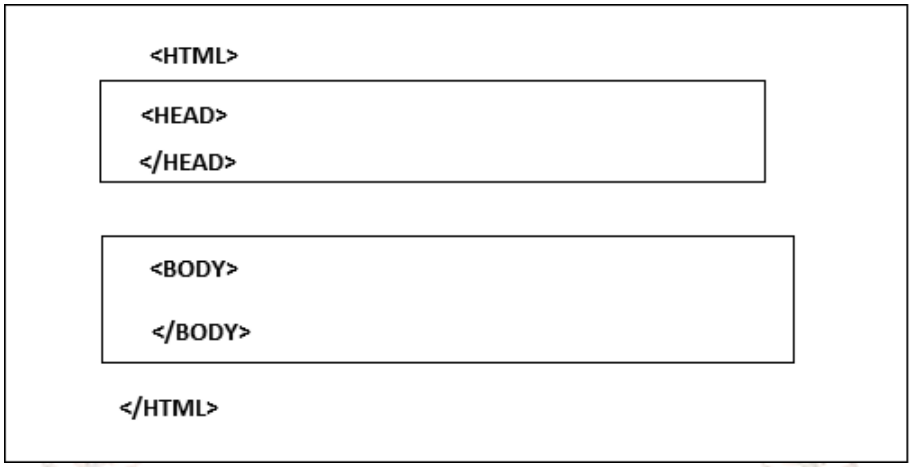

Figure 4.1: Basic Structure of HTML

So what do you need to get started?

- You will need something to key in the HTML code of your page. For this you can use Notepad, WordPad or your favorite word processor.
- A Web Browser - Internet Explorer, Netscape or Opera to view the HTML code you wrote - your web page.

How whole thing does works (from creation of code to getting web page)?

It's a 3 Step Process

a. The Browser gets the HTML code from the source.
b. The Browser interprets the code.
c. The Browser then displays the information.

Whatever HTML document you create, always remember that you have to save it in text-only format (Remember the T in HTML, it stands for Text!). If you are using Notepad, this should not be a problem, but if you are using WordPad or Microsoft Word to create your HTML documents, then while saving, always save as a Text Document. The files that you create should have suffix htm or html.

2.1 Common Tags

<TITLE>: This tag is used to specify the title of the document and Appears in the title bar (top left) of the browser window.

Use **</TITLE>** to close the tag.

<I>: This causes the text to appear slanted (italicized).

Use **</I>** to stop the browser from displaying italicized text.

<B>: This causes the text to appear bold form text.

Use **</B>** to stop the browser from displaying bold form text.

<STYLE>: This tag is used to style a text as well as body or a part of a page. Use </STYLE> to close the tag

<BASE>: This tag is used to specify a base URL for relative links

<SCRIPT>: This tag defines the scripting statements or points to an external script file at the client side

<NOSCRIPT> : This tag is used to display the text for those browsers which do not support the script tag.

<META>: This tag is used to describe the properties like expiry date, name of author, list of keywords, etc.

<LINK>: this tag defines a link between a document and an external resource.

Writing your own Html page

For write your own html page, Open the word processor of your choice (Notepad or Microsoft Word). Type the following the code

Example 1:

<HTML>

<TITLE> my first page </TITLE>

<BODY>

Hello Manipal University Jaipur

</BODY>

</HTML>

Save the file in Text format with suffix .html or htm **(Example1.html).** Open any web browser, click on the File menu, choose the Open option, browse for the file you created just now and click on OK.

In the browser you should be seeing a web page like the one in the following

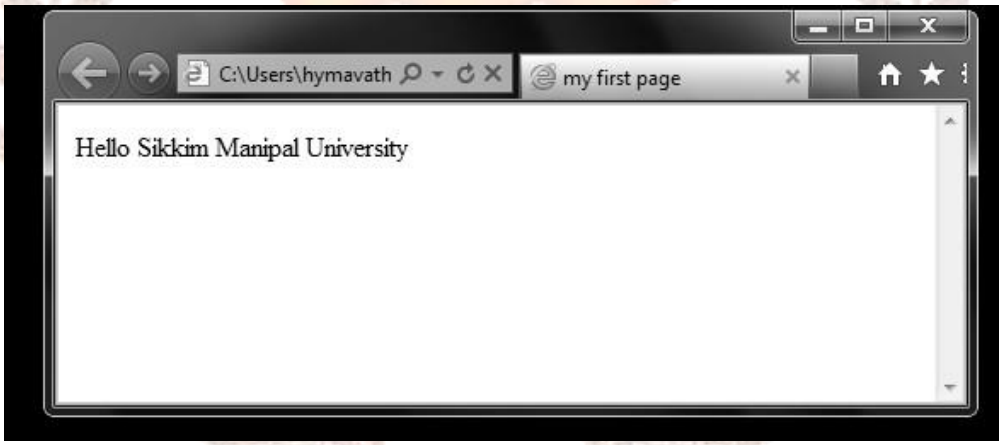

Figure 4.2: your own html page for example1 code.

Look at the title bar of the browser. It displays the title as you specified in your code (my first page). By convention tags are written in uppercase. You can still write them in lower case, as the browser ignores the case of the tags. The browser also ignores the case of the Tag Attributes. The values of these attributes may however be case sensitive.

Use both <I> and <B> tags together. If one wants both italicized and bolded text

<B><I>

This text is both bold and italicized.

</I></B>

Example 2:

```
<HTML>
<TITLE> my second page </TITLE>
<BODY>
<B> My University name is<I> Sikkim Manipal
</I>
</B>
</BODY>
</HTML>
```

Figure 4.3: Web page for bold and italic tag example

SELF-ASSESSMENT QUESTIONS - 2

4. Each web page begins and ends with a special command of _____ .

3. PARAGRAPH AND LINE BREAK UP TAGS

There are some tags that do not follow this open and close format. Some of these are:-

<HR>: This tag draws a horizontal line across the document frame or window. HR stands for Horizontal Reference.

**
:** The
 tag inserts a line beak. Even if you give lots of space (carriage returns) hit enter a lot of times in your html code, the browser does not go onto the next line. It is this tag that tells the browser to move onto the next line.

<P>: This tag is used to display a paragraph and instructs the browser to skip a line before continuing to display the rest of the document.

Below example with the <HR>,
 and <P> tags and see their effect.

Example 3:

```
<html>
<head>
<title>Example of paragraph element</title>
</head>
<body>

HR tag draws a horizontal line across the document frame <HR>

<p>A markup language like HTML is simply a collection of codes/elements tags<BR> that indicate the structure and format of a web document.
 </p>
</body>
</html>
```

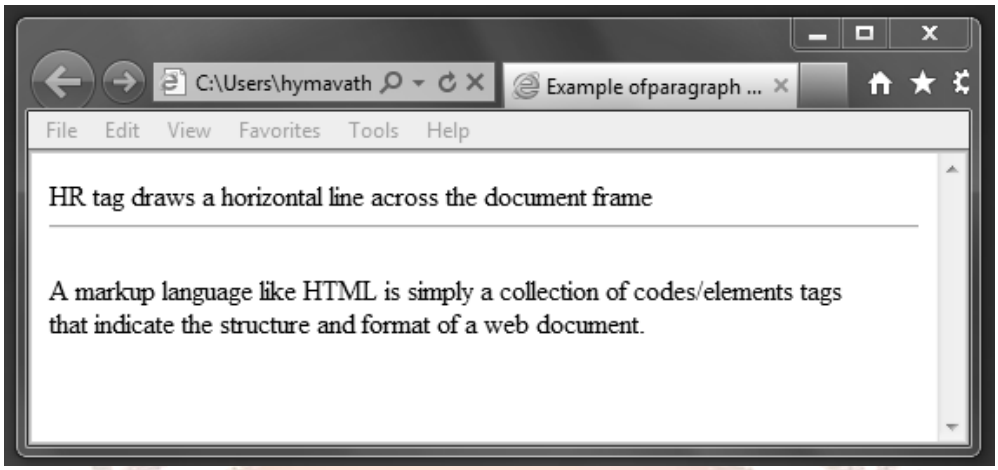

Figure 4.4: Web page for paragraph tag example

You can look at the code that the browser interpreted by clicking on the View Menu and selecting the Source option. The browser collapses multiple spaces and tabs into a single space, so you can put as many spaces, where ever you want in your code. No space is however allowed within the tag names. If you want to put multiple spaces in your web page, you can use &**nbsp.** Every time the browser comes across & nbsp it puts a space.

To align text in a paragraph, you can make use of the ALIGN attribute of the P tag.

To left align the contents of the paragraph

** this paragraph is left aligned **

Similarly to right align the contents of the paragraph set the ALIGN attribute's value to **"RIGHT".** Set ALIGN to **"CENTER"** to center the paragraph.

Example 4:

```
<HTML>
<TITLE> Font and Text Alignment </TITLE>
<BODY>
<p ALIGN = "LEFT" >
 This paragraph is left aligned </p>
```

```
<p ALIGN = "RIGHT" >
This paragraph is right aligned </p>
<p ALIGN = "CENTER" >
This paragraph is centered </p>
</BODY>
</HTML>
```

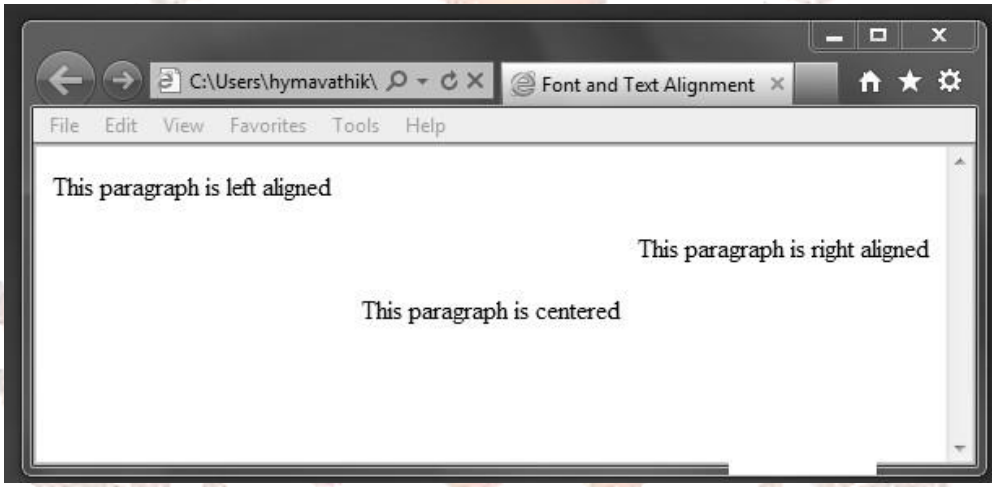

Figure 4.5: Web page for text alignment example

3.1 Headings Tags

If you are writing a web document, you would need to give headings to indicate the contents of the various parts. The font sizes of these texts will be larger than that of the rest. So how do you incorporate headings into your web page? HTML provides you with the tags from **H1 to H6**, with H1 being the most prominent and H6 the least. H7 and H8 can also be used on some browsers. Add a big heading and a small heading to the page you earlier created, by inserting the following code in the body of your HTML document.

<H1> This is a BIG heading </H1>

<H6> This is a small heading </H6>

Example 5:

```
<HTML>
<TITLE> Example3 </TITLE>
 <BODY>
Web Technology <BR>
        <B> my course name is </B>
        <I> BCA </I>
                <H1> This is a BIG heading </H1>
                        <H6> This is a small heading </H6>
</BODY>
</HTML>
```

Figure 4.6: Web page for heading tag example

3.2 Adding Comments

In HTML, a comment begins with <!- - and ends with - -> any text you place after <!- -is comment. Browser ignores text .comment is for your reference, it does not get displayed on the web page. In a comment, you can freely include special characters, such as quotation marks, ampersands, and angle brackets, etc. Your comment can also extent multiple lines. The browser will stop ignoring text once it reads.

Example:

<!- -this is comment line-->

Important points about comments:

- Use comments as a reference guide, avoid use of excessive comments.
- Add comments to major parts of your code
- Use of comments is great way to communicate with other people working on the same web page.
- Browser and such engines ignore comments.

SELF-ASSESSMENT QUESTIONS - 3

5. tag draws a horizontal line across the document frame.
6. Every time the browser comes across   it puts a_____.

4. LIST TAGS & HYPER LINKS

4.1 Creating List

In general terms the phrase "list" refers to a record of short pieces of information arranged in specific format. Examples in the aspect can be a shopping list, list of names of employees in an organization, list of name of places etc.

Lists are good from a structural point of view as they help create a well- structured, more accessible, easy-to-maintain document. In HTML we use concept of list so as to display information carrying specific order. Lists are used to present information in an easy to read form. You can have ordered lists, unordered lists as well as any combination of the two (nested list).

Table 4.2: List tags

Tag	Description
	Defines list items
	Defines an order list
	Defines an unordered list
<dl>	Defines definition of list
<dt>	Defines an item in a definition list
<dd>	Defines a description of an item in a definition list

Unordered List Tag

An unordered list starts with the **<ul>** tag. Each list item starts with **<li>** tag. The list items are marked with bullets (typically small black circles).

Add the following code to a page and see the effect.

<UL>

<LI> this is the first item of the unordered list. </LI>

<LI> this is the second item of the unordered list </LI>

<LI> this is the third item of the unordered list </LI>

</UL>

Your browser should be showing you something like this

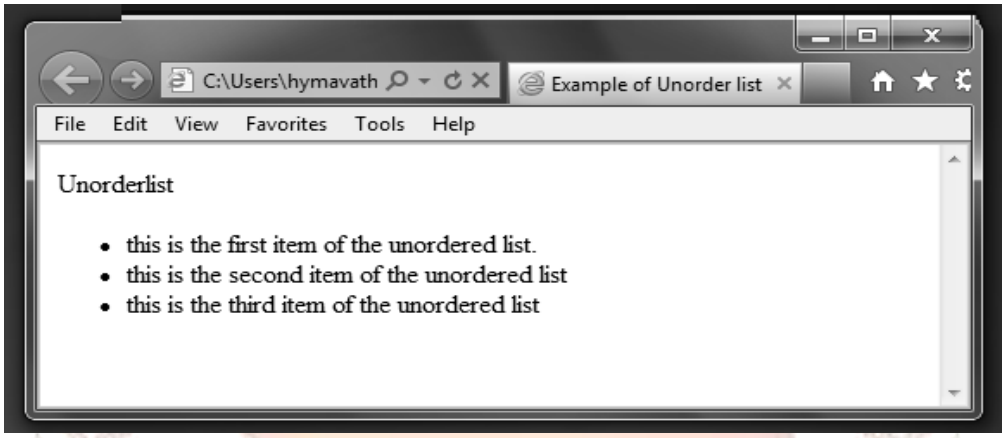

Figure 4.7: web page for unordered list example

Order List Tags

To create a list that puts a number in front of every list item instead of bullets, from the previous list, you only have to change the tag type from UL to OL (Ordered List). The items will be numbered starting from 1.

START is an attribute of Ordered Lists that can be used to specify the number assigned to the first item in the list. This number should be a positive integer.

<OL START = "3" >

Roots the list to have items numbered from 3 onwards.

TYPE is another attribute of Ordered Lists that can be used for specifying the kind of numbering sequence used for each list item.

The type attribute can take any of following values

A: Specifies a sequence of uppercase alphabets.

a: Specifies a sequence of lowercase alphabets.

I: Specifies a sequence of uppercase Roman numerals.

i: Specifies a sequence of lowercase Roman numerals.

1: Specifies a sequence of numbers.

Example of Usage

```
<OL TYPE="I">
<LI> 1st item of the ordered list </LI>
<LI> 2nd item of the ordered list </LI>
<LI> 3rd item of the ordered list </LI>
</OL>
```

This gives an ordered list with the list items numbered in uppercase Roman Numerals.

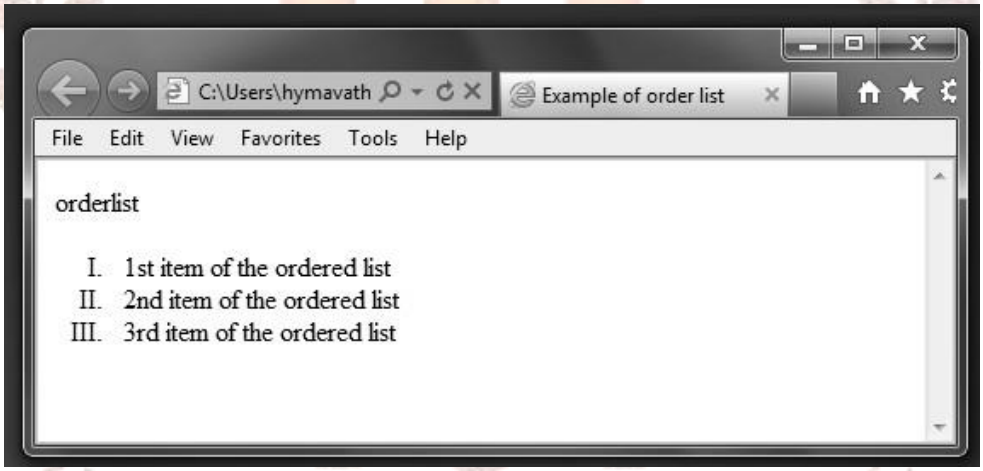

Figure 4.8: Web page for order list example

Nested Lists Tags

One can have an Ordered List inside an Unordered List and vice versa. This can be achieved by putting the nested ordered (or unordered) list as a list member using LI.

Many times you want to create sub lists or sub items have shown in below example. You can do this by using nested lists

Example:

- Nice list
- Another item
    1. sub list item
    2. sub list item
    3. sub list item
- Last list item

Add the Nested list code below, to a web document and view the result in a browser.

```
<HTML>
   <TITLE>Lists </TITLE>
   <BODY> Unordered List
   <UL TYPE = “circle” >
   <LI>this is the first item of the unordered list </LI>
   <LI>this is the second item of the unordered list </LI>
   <LI> this is the third item of the unordered list</LI>
   </UL>
   <BR> Ordered List
   <OL TYPE="I">
        <LI> 1st item of the ordered list < /LI>
        <LI> 2nd item of the ordered list </LI>
   </OL>
   <BR> Nested List
   <OL>
<LI> Main Item 1 </LI>
<LI> Main Item 2 Having Sub Item
   <UL>
<LI> Sub Item 2.1 </LI>
<LI> Sub Item 2.2 </LI>
  </UL>
  </LI>
```

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

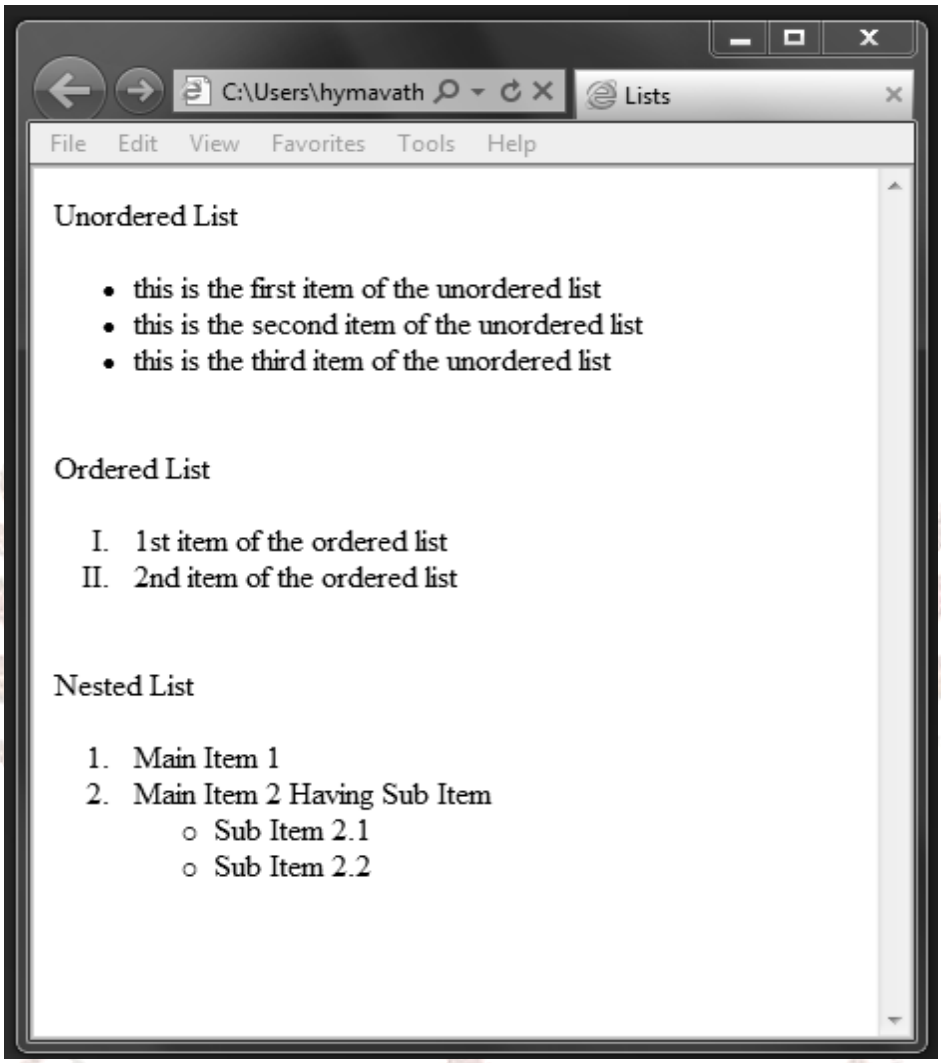

Figure 4.9: Web page for Nested List example.

The Difference between HTML lists and Text

You may be deliberating what the difference between an HTML list and some text with bullets or numbers written by hand. Well, there are some advantages of using a list tag:

1. If you have to change the order of the list item in an order list, you simply move around the list item elements. If you wrote the numbers in manually you would have to go through and change every single number to correct order.

2. Using an HTML list gives the content the proper semantic structure, as well as a "list-ish" visual effect. This has important benefit, rather than just reading out jumble of text in paragraph manner.
3. Text and list are not same. Using text instead of list makes more effort for your document's readers. So if your document needs a list, you should use correct HTML list.

4.2 Creating Hyper Text Links

Twitch of the World Wide Web is the ability to define links from one page to another, and to follow links at click of button.

A hyperlink (or link) is a word, group of words, or image that you can click on to jump to another document. When you move the cursor over a link in a Web page, the arrow will turn into a little hand.

In HTML **<a>** tag defines a hyperlink. If you have a long page and you want to give the visitor the option to click somewhere and directly go the region, then you need a mechanism for telling the browser how and where to find the reference to that desired section. The named anchor comes in handy in this scenario. The most important attribute of the <a> element is the **href attribute,** which indicates the link's destination.

**syntax: Text or Image **

Value of the href attribute of an element holds the target resource of the hyperlink.

Text written with in the **<a>** and **** tags is called **anchor text.**

Example

```
<html>
<head>
<title>Hyperlink</title>
</head>
<body>
<a href="http://www.w3schools.com/"> by clicking the link it Opens the web page w3schools</a>
```

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

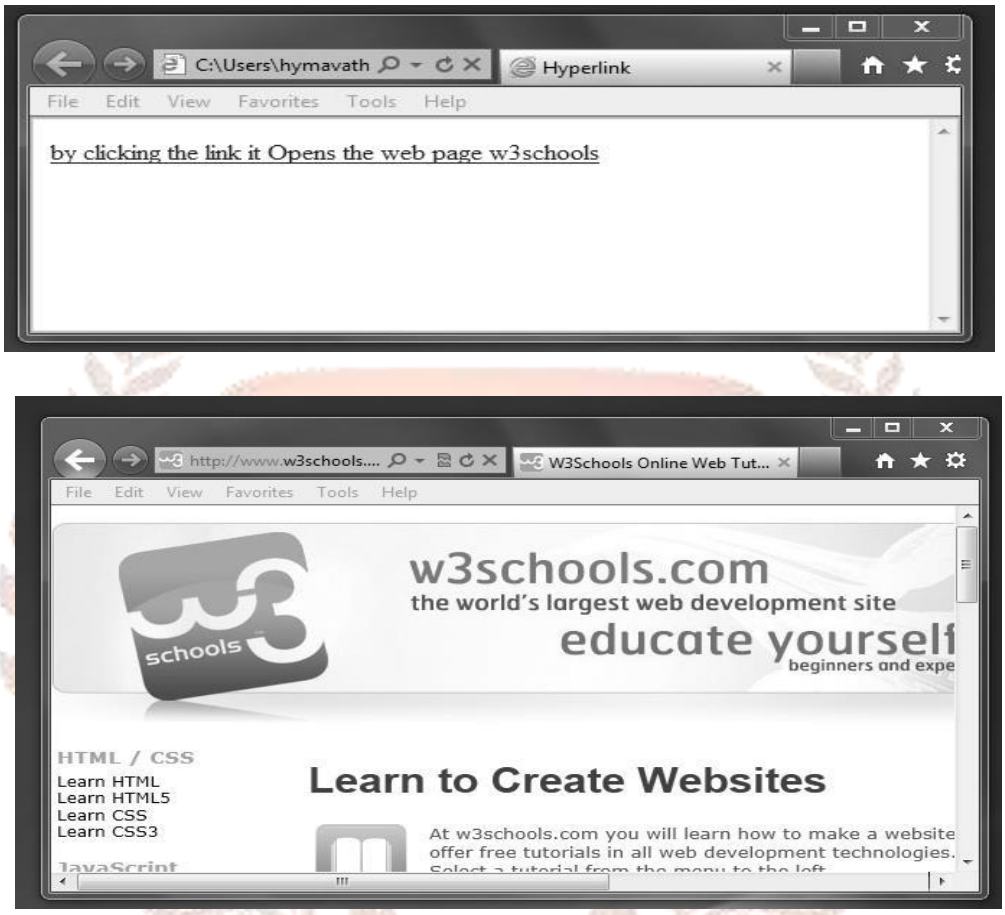

Figure 4.10: Web page for hyper link example

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Linking to a Place in the Same HTML File

Suppose, let say you have one page that is pretty long. May be you would like to give someone a way back to top of the page when you are at the bottom, the way to do this linking to a place

in the same file, which is a specific area of your page you want to make link to that. By using Named anchor link you could link to that section from any part of page.

<A NAME = “my anchor">

This is a Named anchor</A>

Creates a named anchor called my anchor

Clicking on a link created by

<A HREF = "#my anchor">

Click to go to Named Anchor </A>

By specifying the URL of another page and the desired section in that page (separated by a #) in the <A HREF> tag, one can directly jump to the desired section of page.

Possible attributes using anchor tag<A>

Table 4.3: Possible attributes using <a> tag

Attribute name	Possible value	Meaning
Name	String	A name for link end, must be unique with the document (casesensitive)
HREF	URL	Network address for the linked resource, could be another HTML document, file an image etc.
Title	String	A title for linked resource
REL	String	The forward relationship (alsoknown as link type)
REV	String	The reverse relationship

Let see how to create link to top of the page, to do this go to top of the page and type following tag:

** **

You can place any name inside the quote (here name=top).now, go anywhere between body tags and type this link

** back to top **

The # sign is here to let browser known the destination is named anchor within same page.

Example: create anchor named “cool”,

** <!— named anchor -->**

by using #cool ,you can access a named anchor section from any place of the page(one at third paragraph, the bottom, the 500th word etc.).

<!— to linked to the cool section place below tag -->

its go to the cool section of the page It can make navigating same pages a whole lot easier.

Creating Link Lists

Some time you may require creating directory of list and those may link to some other pages. Then you would use both hyper link and list concept

Example: To create a links on the home page for other three pages.

```
<html>
    <TITLE>home page </TITLE>
    <body>
    <OL>
        <LI> <a href= “page1.html”>page1</a> </LI>
        <LI> <a href= “page2.html”>page2</a> </LI>
        <LI> <a href= “page3.html”>page3 </a> </LI>
    </OL>
    </body>
    </html>
```

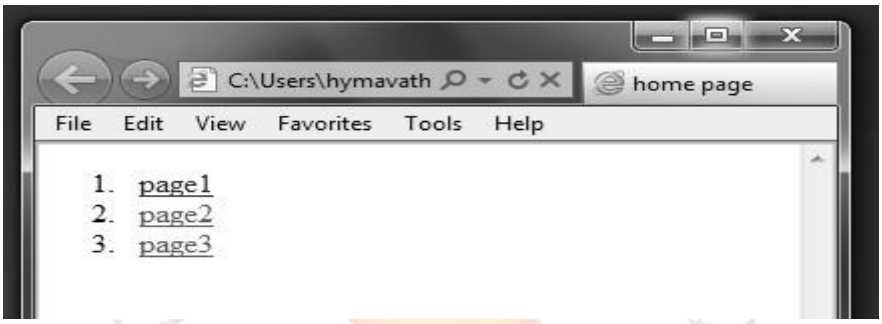

Figure 4.11: example home page that links to three other pages

SELF-ASSESSMENT QUESTIONS - 4

7. _____ is an attribute of Ordered Lists that can be used to specify the number assigned to the first item in the list.
8. In HTML _____ tag defines a hyperlink.

5. SUMMARY

- A markup language is a language that annotates text so that the computer can manipulate the text.
- An SGML Document Type Definition (DTD) specifies valid tag names and element attributes.
- A markup language like HTML is simply a collection of codes/elements tags that indicate the structure and format of a web document.
- Lists are good from a structural point of view as they help create a well- structured, more accessible, easy-to-maintain document
- A hyperlink (or link) is a word, group of words, or image that you can click on to jump to another document. In HTML <a> tag defines a hyperlink.

6. TERMINAL QUESTIONS

1. What is markup language and briefly explain any two popular markup languages.
2. Briefly explain the structure tags of HTML document.
3. Explain how to display information in list form and explain different list tags available in HTML?
4. Briefly explain the Paragraph tag.
5. Explain hypertext link concept used in HTML?
6. to display a paragraph and instructs the browser to skip a line before continuing to display the rest of the document.

7. ANSWERS

Self Assessment Questions

1. Standard Generalized Markup Language
2. Markup
3. URLs
4. <HTML>
5. HR
6. Space
7. START
8. <a>

Terminal Questions

1. A markup language is a language that annotates text so that the computer can manipulate the text. In recent year's number of markup languages have been developed. For more details refer to section 4.1.1.
2. Each web page begins and ends with a special command: **<HTML>** and the ending **</HTML>.** Inside the web page, there are two basic parts: the **HEAD** and the **BODY**. For more details refer to section4.2.
3. Lists are used to present information in an easy to read form. You can have ordered lists, unordered lists as well as any combination of the two (nested list). For more details refer to section 4.4.1.
4. In HTML paragraph instructs the browser to skip a line and specifies the alignment.For more details refer to section 4.3.
5. In HTML <a> tag defines a hyperlink. For more details refer to section 4.4.2.

8. REFERENCES

- The Essential Guide to CSS and HTML Web Design (Essentials) by raig Grannell.
- HTML and css the complete reference, Thomas A. Powell.
- Build your own website the Right way using HTML&CSS, 2nd edition by lan Lioyd.