Master of Computer Application

(Open and Distance Learning Mode)

Semester – II

Internet and Web Technology

Centre for Distance and Online Education (CDOE)
DEVI AHILYA VISHWAVIDYALAYA, INDORE
“A+” Grade Accredited by NAAC
IET Campus, Khandwa Road, Indore - 452001
www.cdoedavv.ac.in
www.dde.dauniv.ac.in

Program Coordinator

Dr. Anand More
School of Computer Science and IT
Devi Ahilya Vishwavidyalaya, Indore – 452001

Content Design Committee

Dr. Pratosh Bansal
Centre for Distance and Online Education
Devi Ahilya Vishwavidyalaya, Indore – 452001

Dr. C.P. Patidar
Institute of Engineering & Technology
Devi Ahilya Vishwavidyalaya, Indore – 452001

Dr. Shaligram Prajapat
International Institute of Professional Studies
Devi Ahilya Vishwavidyalaya, Indore – 452001

Language Editors

Dr. Arti Sharan
Institute of Engineering & Technology
Devi Ahilya Vishwavidyalaya, Indore – 452001

Dr. Ruchi Singh
Institute of Engineering & Technology
Devi Ahilya Vishwavidyalaya, Indore – 452001

SLM Author(s)

Mr. Ravindra Yadav
B.E., M.E.
IET, Devi Ahilya Vishwavidyalaya, Indore – 452001

Mr. Vikas Vankhede
B.E., M.E.
IET, Devi Ahilya Vishwavidyalaya, Indore – 452001

Edition : 2022 (Restricted Circulation)
Published by : Centre for Distance and Online Education (CDOE), Devi Ahilya Vishwavidyalaya
Printed at : University Press, Devi Ahilya Vishwavidyalaya, Indore – 452001

Internet and Web Technology

Table of Content

Lesson 1: INTRODUCTION TO HTML

Lesson 2: CREATING SIMPLE WEB PAGES

Lesson 4: JAVA SCRIPT AND ITS OBJECT

Lesson 5: PHP BASICS

Lesson 6: CONTROL STRUCTURES

Lesson 7: FUNCTIONS IN PHP

Lesson 8: STRINGS IN PHP

Lesson 9: ARRAYS

Lesson 10: FILE AND DIRECTORY HANDLING

INTRODUCTION TO HTML

Structure

1.0 OBJECTIVES

After this lesson, you will be able to:

- Know about the World Wide Web and Browsers
- Know the structure of a HTML document.
- Create and modify HTML documents using a simple text editor.

1.1 What Is The Worldwide Web ?

The World Wide Web is the largest information service on the Internet. It is a collection of millions of computers linked together on a computer network. The network ties all computers together with one another. Computers throughout the world connect to the web through modems and phone lines that dial-up an Internet Service Provider (ISP).

WWW is an application of hypertext: text with links. Hypertext concepts have been present in print for a long time, in the references made from the text to footnotes, and to other works in bibliographies. However hypertext in electronic documents simplifies the act of following the links, and to the user a hypertext system can be viewed as a seamless whole. WWW extends the hypertext concept by enabling the links to be to documents, images, etc anywhere on the global Internet.

The idea behind hypertext is that instead of reading text in rigid, linear structure (such as a book) you can skip easily from one point to another. You can get more information, navigate through the text and visual information based on what interests you at that time.

The browsers are used to navigate Web pages and other information on the World Wide Web. Earlier you had to navigate the Internet's various services by typing the commands. They involved simple text-only connections. The graphical web browsers came later and paved the way to display both text and graphics in full color.

Tim Berners-Lee, the originator of WWW, describes it as a "*distributed heterogeneous collaborative multimedia information system*". What this means is that it is:

1.1.1 distributed

information in WWW is distributed globally across thousands of web sites, all over the Internet. Each site contributes for the information it publishes. We, as consumers go to that site and view the information. Once we have gone through it, we can jump to any other site by clicking on the hyperlinks.

Each web site's address is a unique address called Uniform Resource Locator (URL). URLs are starting to serve the same purpose in the Internet world that the bibliographic reference serves in the print medium. (We will learn more about URL later)

1.1.2 heterogenous

it accesses a range of different information sources on the Internet that previously required different software.

1.1.3 collaborative

construction of the information resources is shared between people at many different sites.

1.1.4 multimedia

text, images, and sounds are available through the system.

Using WWW involves starting a browser, which initially presents a "home page" of information useful to the particular user. Links in this home page lead to other documents, indexes, and navigation tools. Home pages can be easily customised for an organisation or a particular user.

While WWW documents are held on a wide variety of machines, using different operating systems, they are held in a standard format, HyperText Markup Language, (HTML) which can be interpreted by a user's browser software. HTML is derived from Standard Generalised Markup Language (SGML), the well-known typesetting and document exchange format. HTML specifies that a piece of text is a header, or a link, and leaves it to the browsing software to determine how the header or link appears to the viewer.

The combination of a standardised format - HTML, and a standardised locator, - the URL, has made WWW a widely accepted tool to navigate and provide information on the Internet.

1.2 Web Browsers

A web browser is the program that you use to view pages and navigate the World Wide Web. A wide range of Web Browsers are available for graphical-user-interface-based systems (for eg. Windows and Macintosh) and text-only systems for dial-up Unix connections. Most of the browsers are freeware or shareware. For example, both Internet Explorer and Netscape Navigator are available free of cost. These are the most popular browsers for the World Wide Web.

The web browser deals with formatting and displaying web documents. Each web page is a file written in a language called Hyper Text Markup Language (HTML) that includes the text of the page, its structure, and links to other documents, images or other media.

The browser takes the information it gets from the Web server and formats and displays it for your system. In fact, retrieving document from the web and formatting them for the client system are the two main tasks of the browser.

The Web browsers are written by different people. Each person has their own idea about how Web documents should look. Therefore, any given Web document will be displayed differently by different browsers. Different browsers may format and display the same file differently, depending on the capabilities of that system and the default layout options for the browser itself. Depending on the browser the client uses and the features it includes, he/she may be able to play multimedia files, read mails, play applets and make use of other advanced features offered by the browser.

Therefore, you need to keep this principle foremost in your mind at all times: *you cannot guarantee that your document will appear to other people exactly as it does to you.* Different browsers running on different platforms may have different style mappings for each page element and actual styles on the screen.

On the other hand, you want to write documents which look acceptable to most people. How? Well, it's almost an art form in itself, but my recommendation is that you assume that most people will set their browser to display text in a common font such as Times at a point size of somewhere between 10 and 15 points.

1.3 HTML: Behind The Scenes

HTML documents are essentially plain text files. They contain no images, no sounds, videos and no animations; however that can contain pointers or links to these file types. That is how web pages end up looking as if they contain non-text elements.

HTML is a markup language means that you start with the text of your page and add special tags around the words and paragraphs. HTML is composed of *tags and attributes* that serve to identify parts and characteristics of HTML documents. It has a defined set of tags we can use. We cannot create our own tags to give new appearance. Some tags provide document structure; others reference other files. Attributes provide additional information within tags. They are optional parts of the tags and modify or more thoroughly specify information in the tags such as alignment, color or height. When we read a HTML page into a browser such as say Netscape or Internet Explorer, the browser interprets the HTML tags and formats the text and the images on the screen. The browser has mappings between the names of page elements and actual styles on the screen.

HTML tags are *always* enclosed in angle-brackets (< >) and are case-insensitive; that is, it doesn't matter whether you type them in upper or lower case.

Tags typically occur in begin-end pairs. These pairs are in the form

<tag> ... </tag>

where the <tag> indicates the beginning of a tag-pair, and the </tag> indicates the end. (The three dots indicate an arbitrary amount of content between the tags.) The usual way to refer to each tag is "*tag*" for the first and "slash-*tag*" for the second, where *tag* is the actual name of the tag being discussed.

These pairs define *containers*. Any content within a container has the rules of that container applied to it. For example, the text within a "boldface container" would be boldfaced. Similarly, paragraphs are defined using a "paragraph container."

Thinking of tag-sets as containers will help in another way: it will help you remember that tags should always be balanced. In other words, you should keep containers nested within each other, just as you would have to do in the real world. Let's try some visual examples where we actually draw the containers:

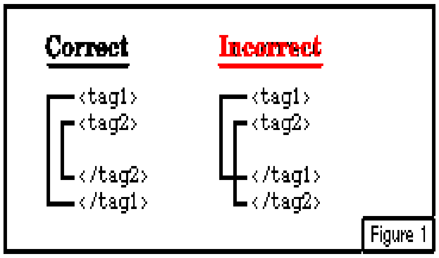


Figure 1

If you start overlapping containers as shown on the right, about the best you can expect is that the document will be formatted in unexpected ways.

One more thing to keep in mind with regards to containers. Since HTML is based on these structures, it is often the case that the arrangement of text within a container is irrelevant. For example, within a paragraph container, all of the text can be in one long line, or in a series of separate lines, or with every word on its own line, or with every word separated from every other by nineteen spaces. These would all be displayed exactly the same.

Therefore, try to keep in mind this thought: *whitespace doesn't matter.* (Whitespace is all of the blank areas in a text file--empty lines, extra spaces, and so on.)

Not every tag in HTML is paired. Some tags, such as the line-break tag, stand on their own (that is, they have no closing tag). These are known as *empty tags.*

When a web browser reads from a plain text file, it looks for special codes or "tags" that are marked by the < and > signs. The general format for a HTML tag is:

```
<tag_name>string of text</tag_name>
```

i.e to apply tags to information in your document, place the opening tag before the information and place the closing tag after the information

As an example, the heading tag is used as

```
<h3>Welcome to my site </h3>
```

This tag tells a web browser to display the text Welcome to my site in the style of header level 3 (We'll learn more about these tags later). HTML tags may tell a web browser to bold the text, italicize it, make it into a header, or make it be a hypertext link to another web page. It is important to note that the ending tag,

```
</tag_name>
```

contains the "/" slash character. This "/" slash tells a web browser to stop tagging the text. Many HTML tags are paired this way. If you forget the slash, a web browser will continue the tag for the rest of the text in your document, producing undesirable results

NOTE: A web browser does not care if you use upper or lower case. For example, <h3>...</h3> is no different from <H3>...</H3>

Unlike computer programming, if you make a typographical error in HTML you will not get a "bomb" or "crash" the system; your web page will simply look, well... wrong. It is quick and easy to go inside the HTML and make the changes. An interesting aspect of HTML is that if the browser does not know what to do with a given tag, it will ignore it!

For creating the web documents initially, you need only a HTML editor to create and save the HTML file and a web browser (Internet Explorer or Netscape Navigator) to view and test the HTML document.

1.4 Creating A New Html Document

If you are just starting out, we most STRONGLY recommend that you use the simplest text editor available -- the Windows *NotePad*.

Also, it will help you if you first create a new directory/folder on your computer and just make sure that you save all of the files you create(all the documents related to a particular project) in the same folder. This makes using links, images etc easier.

An HTML document contains two distinct parts, the head and the body. The head contains information about the document that is not displayed on the screen. The body then contains everything else that is displayed as part of the web page.

The basic structure then of any HTML page is:

```
 <html>
<head>
<!-- header info used to contain extra information about  this document,
      not displayed on the page -->
</head>


<body>


<!-- all the HTML for display -->
        :      :
        :      :
        :      :
</body>
</html>
```

1.4.1 HTML

The first and last tags in a document should always be the HTML tags. These are the tags that tell a Web browser where the HTML in your document begins and ends.

1.4.2 HEAD

The HEAD tags contain all of the document's header information. When I say "header," I don't mean what appears at the top of the browser window, but things like the document title and so on.

1.4.3 TITLE

This container is placed within the HEAD structure. Between the TITLE tags, you should have the title of your document. This will appear at the top of the browser's title bar, and also appears in the history list.

What you type should probably be something which indicates the document's contents, but it doesn't have to be. The length of the title is pretty much unlimited, but don't go overboard. If you don't type anything between the TITLE tags, or don't include the TITLE tags at all, then the browser will typically use the actual file name for the title.

You should only have one TITLE container per document.

1.4.4 BODY

BODY comes after the HEAD structure. Between the BODY tags, you find all of the stuff that gets displayed in the browser window. All of the text, the graphics, and links, and so on -- these things occur between the BODY tags.

Comment Tags

If you want to leave yourself notes in an HTML document, but don't want those notes to show up in the browser window, you need to use the comment tag. To do that, you would do the following:

```
<!-- Hi, I'm a comment. -->
```

Your note would go where the text *Hi, I'm a comment.* appears. Yes, you do need an exclamation point after the opening bracket, but not before the closing bracket. Also, there is no end tag; that is, a tag like </!-- text --> does not exist. This is an example of an empty tag.

The text between the tags is NOT displayed in the web page but is for information that might be of use to you or anyone else who might look at the HTML code behind the web page.

You can put comments pretty much anywhere, but you have to be aware of one important thing: you shouldn't put any HTML markup within a comment tag. Theoretically, you should be able to, but most browsers handle this less than gracefully (i.e., they either mess up or crash). When your web pages get complicated, the comments will be very helpful when you need to update a page you may have created long ago.

Here are the steps for creating your first HTML file.

1. If it is not open already, launch your text editor program.
2. Go to the text editor window.
3. Enter the following text (you do not have to press RETURN at the end of each line; the web browser will word wrap all text):

```
  <html>
  <head>
<title> My first web page </title>
 </head>
  <!--  this is a comment tag       -->
<body>
This is my personal web page
```

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

Save the document as a file called "trial.html" and keep it in the folder/directory you create for your website. Also, if you are using a word processor program to create your HTML, be sure to save in plain text (or ASCII) format.

By using this file name extension, a web browser will know to read these text files as HTML and properly display the web page.

1.5 Displaying Your Document In A Web Browser

To view the HTML document that you developed, the steps are:

1.5.1 Open the web browser (Ex: Internet Explorer)

1.5.2 Select File... Open from the File menu.

1.5.3 Type the local filename or click Browse button to browse your hard drive

1.5.4 Use the dialog box to find and open the file you created, "trial.html"

1.5.5 You should now see in the title bar of the workspace window the text "My first web page" and in the web page below, the one sentence of <body> text you wrote, " This is my personal web page"

1.6 Modifying An Html Document

Now that you have created your first HTML document, you will learn how to swiftly make changes in your document and view the updates within your web browser.

Steps for making changes in your HTML Document

1. Go to the text editor window.
2. Below the text you typed from the previous example, press RETURN a few times and type the following text:
3. My institute
4. Here is my family – my sisters and brothers

Note that this text should be above the </body> and </html> tags at the bottom of your HTML file.

Select Save from the File menu to update the changes in your HTML file.

1.7 Reloading the Document in your Web Browser

Return to the web browser workspace where the previous version of your file was displayed. Note that the new text you entered in the previous steps may not yet be visible. To see the changes, use the Reload button or menu item in your web browser. This instructs your web browser to read in the same HTML file and display it with whatever changes have been made. You should now see the new text that you entered.

Note that the web browser ignores all blank lines and extra spaces (carriage returns) that you enter in the HTML file. It will also ignore any extra space characters (beyond the one between words).

However, when you are writing HTML, it will help you greatly to separate major sections by some blank lines... when you need to go back and edit content, it makes it easier to locate the correct location to make the changes.

1.8 Model Questions

1. What are HTML tags?
2. Where is the text of the title tag displayed?
3. What steps are involved in creating a simple HTML document?
4. How do you create a comment tag?
5. How can you display your HTML document in a web browser?

1.9 Tick the right answer

1. The actual contents of a Web page are displayed in:

   ○ real time

   ○ the fastest possible manner

   ● the browser display window

   ○ the history list

   ○ the nick of time

2. Web browsers do not pay attention to the whitespace in an HTML document.

   ● True

   ○ False

3. HTML tags for which no corresponding close-tag exists are called:

   ○ solo tags

   ● empty tags

   ○ single tags

   ○ lonely tags

   ○ unmatched tags

4. Once you have created a Web page, your users will see the page exactly the same way it appears to you.

   ● True

   ○ False

5. Most HTML tags have the structure:

○ <tag> ... <tag>

○ <tag> ... <!tag>

● <tag> ... </tag>

○ <tag1> ... <tag2>

○ <tag> ... <notag>

****

Lesson - 2

CREATING SIMPLE WEB PAGES

Structure

2.0 Objectives

After this lesson, you will be able to:

- Identify the different levels of headings in HTML and the tags associated with them.
- Identify the HTML tags for the text styles: bold, *italic*, and typewriter (mono-spaced).
- Identify HTML codes for creating unordered, ordered, and nested lists for a web page.
- Place an inline image within your HTML document and set its attributes
- Create hyperlinks in the document
- Design a web page table with rows and columns of text in a gridded display
- Write the HTML for the basic web form elements - text fields, password fields , text area fields ,radio buttons , checkboxes

2.1 Headings

The heading structures are most commonly used to set apart document or section titles. They break up large areas of text and arrange information according to a logical hierarchy. HTML defines six levels of headings.

The format for an HTML heading tag is:

```
<hN>Text to Appear in Heading</hN>
```

where N is a number from 1 to 6 identifying the heading size. Heading 1 (H1) is "most important" and Heading 6 (H6) is "least important."

EX: <h1> My Institute </h1>

Remember that these heading structures go into the body of the document. The headings being discussed here have nothing to do with the HEAD structure from the previous chapter.

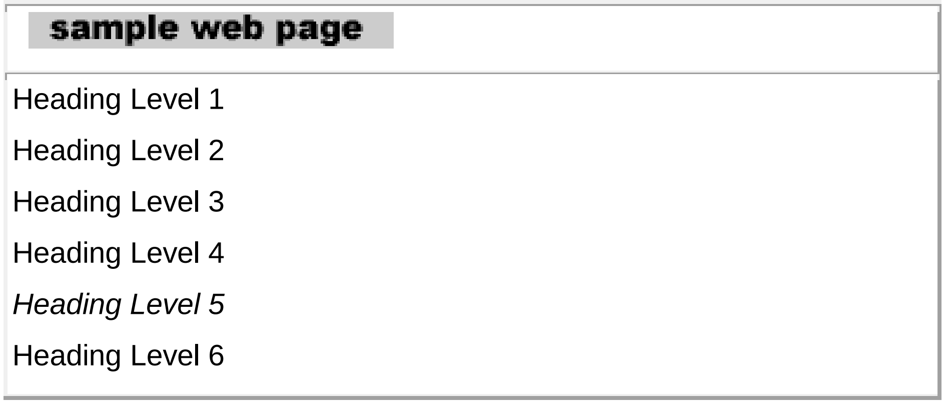
The headings when they are displayed are not numbered. By default, browsers will display the six heading levels in the same font, with the point size decreasing as the importance of the heading decreases. Here are all six HTML pairs, in descending order of importance:

```
<H1>Heading Level 1</H1>
<H2>Heading Level 2</H2>
<H3>Heading Level 3</H3>
<H4>Heading Level 4</H4>
```

<H5>Heading Level 5</H5>

<H6>Heading Level 6 </H6>

These six lines, when placed into the body tag of the HTML document, will simply display the six levels of headings like shown below:



sample web page

Heading Level 1

Heading Level 2

Heading Level 3

Heading Level 4

Heading Level 5

Heading Level 6

Since, as we have discussed, *whitespace doesn't matter*, you might have thought that the above block of HTML code would just string the content into one line of text. However, because headings are meant for section titles and the like, themselves define them as existing on a line. A heading always begins at the margin of a line and always forces a line break at the end of the heading. In other words, you cannot have two heading levels on the same line.

This also means that you cannot highlight text in the middle of a paragraph by marking it as a heading. If you try this, the paragraph will be split in two, with the heading text on its own line between the two pieces

Like an outline, your heading levels should have logical, consistent order and be parallel. Headings can be of any length including many lines of text. The text inside the heading tags are displayed either in larger or bolder text , are centered or underlined – so that they stand out from regular text. The exact appearance of headings may vary from browser to browser. For ex: a first level heading in one browser might display it as 16 point Arial Bold and another browser may display it as 14 point Times New Roman, Bold. Thus specific font style, font size, emphasis associated with each heading may vary with browser.

The align attribute can be used to align the text in the headings to right, center or left. By default, the browsers align headings on the left.

Include the align attribute in the initial heading tag as shown below:

<h1 align=center> First level Heading centered </h1>

<h2 align=right> This second level heading is right aligned </h2>

2.2 Paragraphs

Paragraphs are quite common in Web pages. They are one of the most basic structures in HTML. The beginning of a paragraph is marked by <P>, and the end by </P>. Paragraph tag is actually unpaired tag. You can simply use the <p> tag where you want the paragraph to start. However, by using both the starting and ending paragraph tags it is easy to identify where a particular paragraph starts and ends. So use the paragraph tags around the text you want to format as a paragraph.

Ex:

<p> enter your paragraph text here </p>

We've seen earlier that a web browser will ignore all of the CARRIAGE RETURNS typed into your text editor. But, wherever a browser sees the paragraph tag, it inserts a blank line and starts a new paragraph.

As in headings tag, you can have alignment attribute inside the opening tag of paragraph. The values for alignment attribute are align=left, align=right, align=center.

Ex:

<p align= center> this whole paragraph will be centered </p>

2.3 OTHER TYPES OF BREAKS

To separate major sections of a web page, use the horizontal rule or hr tag. This inserts a straight line.

The HTML format for a horizontal rule tag is:

<hr>

The hr tag has no closing tag and has no text associated with it. These horizontal lines can be used to visually separate sections of text on the page. It looks like this:

The attributes that can be associated with this tag are size, alignment and width. The size attribute indicates the thickness of the line in pixels, the default thickness being 2. The width indicates the horizontal width of the line. This can be specified either as exact number of pixels or as a percentage of the screen width. When you specify as percentage, it changes with the change in size of window.

The attributes can be used as follows:

<hr size= “4” >

<hr width= “50%”>

<hr width= “ 75%”   size= “3”   align= “center”>

The values for align attribute are left, right and center. By default the rule lines are centered.

Line break

So what if we want to end a line after a certain word, but don't want to start a new paragraph? Well, what we need is a line break, which is invoked by using the <BR> tag. This forces a line break wherever you place it in the content (that is, whatever is after the <BR> tag will start from the left margin of the next line on the screen.). The <br> tag which forces text to a new line like the <p> tag, but without inserting a blank line. You might use this tag when making a list of items, when writing the lines of a poem, etc

And no, there is no </BR> tag. The line break tag is an empty tag. And truly, the concept of a line break beginning and ending doesn't really work, since a line break is a one-shot occurrence.

2.4 HTML Style Tags

HTML offers several tags for adding style to your text. These are the character level tags which apply formatting to individual characters or words (unlike paragraph level tags which apply formatting to sections of text).

The tags I will cover here are sometimes called *forced style tags,* because their very nature forces a certain style within the document

The four most commonly used forced style tags are very simple:

Boldface

Everything between <B> and </B> is boldfaced.

Italics

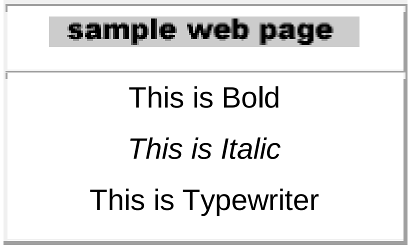
Everything between <I> and </I> is *italicized.*

Underline

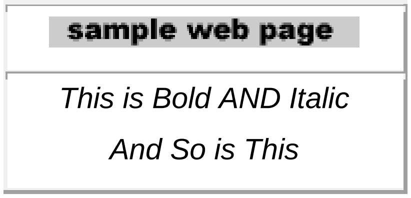
Everything between <U> and </U> should be underlined.

Typewriter Text

Everything between <TT> and </TT> is in typewriter text (a monospaced font in most browsers). This is typically used for variable names, or to show snippets of HTML.

HTML	Result
<b>This is Bold...</b> <i>This is Italic...</i> <tt>This is Typewriter...</tt>	 sample web page This is Bold *This is Italic* This is Typewriter

Note how you can combine the style tags as long as they are correctly nested, the italic tags are both *within* the bold tags. Note also, that the order does not matter.

HTML	Result
<i><b>This is Bold AND Italic</b></i> <b><i>And So is This</i></b>	 sample web page *This is Bold AND Italic* *And So is This*

Furthermore, you can also add style to the text that appears in heading tags. Note how the different style tags are opened and closed around the words they style and how the heading tags surround the whole text for the heading.

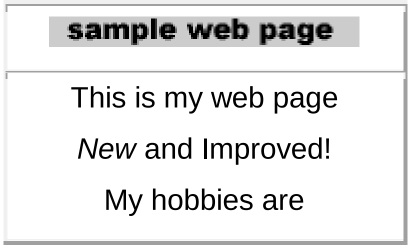
HTML

This is my web page

<h2><i>New</i> and
<tt>Improved!</tt></h2>

My hobbies are

Result



This is my web page

New and Improved!

My hobbies are

The other common character formatting tags are

Character tag	Effect
<cite>	Indicates citations or references
<em>	Applies emphasis;
<strike>	Applies strikethrough to text
<strong>	Applies stronger emphasis; usually displayed as bold
<sub>	Formats text as subscript
<sup>	Formats text as superscript

2.5 Lists

Lists provide information in a structured, easy to read format. They help in drawing the attention of the visitor to important information. While simple in concept, lists can be very powerful in execution. There are two types of lists:

- Unordered lists (items preceded with a "bullet")
- Ordered or numbered lists (a sequentially numbered list)

These lists are easy to format in HTML, and they may even be nested (lists of lists) to produce an outline format. Lists are also handy for creating an index or table of contents to a series of documents or chapters.

2.5.1 Unordered Lists

The "unordered list" is -- a list of items, each one preceded by a "bullet" (a distinctive character; typically, a small black circle).

The list begins and ends with the tags <UL> and </UL> respectively.

Each item in the list is marked using the <LI> tag, which stands for "List Item." <LI> has a corresponding </LI>, but this closing tag is not required to end the list item (although you could use one if you really wanted to).

You can use as many list items as you like, up to your browser's built-in maximum, if any.

Here's the markup for a simple list:

```
<UL>
<LI>Monday
<LI>Tuesday
<LI>Wednesday
<LI>Thursday
<LI>Friday
</UL>
```

If you loaded an HTML page containing the markup above, you would see the days of the week, each one preceded by a "bullet." To wit:

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday

Almost anything can be put into a list item -- line breaks, entire paragraphs, images, links, or even other lists. For example:

```
<UL>
<LI>Monday
<LI>Tuesday
<LI>Wednesday
  <UL>
  <LI>6am - 9am
  <LI>9am - 12n
  <LI>12n - 3pm
  <LI>3pm - 6pm
  </UL>
<LI>Thursday
<LI>Friday
</UL>
```

In the above case, under "Wednesday" in the 'outer list,' you would find another unordered list (the three-hour blocks of time), which is referred to as a *nested list.*

Here's how it looks:

- Monday
- Tuesday
- Wednesday
    - 6am - 9am
    - 9am - 12n
    - 12n - 3pm
    - 3pm - 6pm
- Thursday
- Friday

2.5.2 Ordered Lists

On the face of it, ordered lists look a lot like unordered lists, and a lot of the same rules apply to both constructs. The only difference in HTML is that instead of using <UL> and </UL>, an ordered list is contained within the tags <OL> and </OL>. Ordered lists are based on list items, just as unordered lists are.

However, when an ordered list is displayed in a Web browser, it uses an automatically generated sequence of item markers. In other words, the items are numbered. The markup for a simple ordered list, based on the earlier example is:

```
<OL>
<LI>Monday
<LI>Tuesday
<LI>Wednesday
<LI>Thursday
<LI>Friday
</OL>
```

The above markup will look similar to the previously discussed simple unordered list, with the important difference that each day of the week is numbered instead of preceded by a "bullet." In other words, it looks like this:

1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday

Ordered lists are as nestable as unordered lists, and you can nest unordered lists in ordered lists, as well as the other way around.

You can experiment with the <OL> tag. For example, you can change the type of list (alphabetical, numerical or roman list), starting of list etc.

To change the type of list:

```
<OL type = “A”>
        <LI> First Item </LI>
        <LI> Second Item </LI>
        <LI> Third Item </LI>
        <LI> Fourth Item </LI>
</OL>
```

Output:

```
A. First Item
B. Second Item
C. Third Item
D. Fourth Item
```

You can also specify the stating of the list like:

```
<OL start = 5>
        <LI> First Item </LI>
        <LI> Second Item </LI>
        <LI> Third Item </LI>
        <LI> Fourth Item </LI>
</OL>
```

Output:

```
5. First Item
6. Second Item
7. Third Item
8. Fourth Item
```

You can specify value of type to “a”, “I”, “i” for small alphabetical list, Capital Roman latter list and small Roman list respectively.

2.6 Adding Colors

Web pages without color look dull. You can enliven your Web page with a splash of color. For example, add background color to the page, change the text colors or highlight words.

Colors can be specified using hexadecimal numbers, which combine proportions of Red, Green and Blue.

Background color

The background color which you specify fills up the entire browser window. To specify background color to the document, include the bgcolor attribute in the body tag. Specify the value of the color in quotation.

```
< body bgcolor= " #FFFFFF">
```

To use the color names simply use the name of the color as the value to the bgcolor

```
<body bgcolor= "green">
```

Text color

To change the color of the text you need color names or numbers as in the case of the background. The text attribute controls the color of all the page's body text that is not in link, including headings, body text, text inside tables and so on

```
<body text= "white">
```

```
<body bgcolor= "red"  text = "black">
```

2.7 Graphics

The way a web browser displays graphics in HTML format indicates the location of a graphic file in a single format that can be interpreted by different types of computers. For example, when the information in that format is received by Macintosh computer, the web browser knows to display it as a picture format for Macintosh. However, when that same information is received by Windows browser, it is displayed as a Windows graphic.

In technical jargon, we would say that this picture format is platform independent. HTML itself is platform independent, since plain text characters can be understood by any computer.

The standard format that can display within a web page is GIF or Graphics Interchange Format. The GIF compresses the picture information (reduces the file size) and translates it to binary code that can be sent over the Internet. GIF compression is most effective on graphics that have contiguous areas of solid color, and compression is even greater when the color is continuous in the horizontal direction. GIF images have the feature of defining a color to be "transparent" so images can appear to have non-rectangular boundaries. They can also be saved in the "interlaced" format so that when you see a web page, the images start to appear soon and "dissolve" to the final image.

The other file format used on the web is JPEG (named after the Joint Photographic Expert Group that designed this format). JPEG compression is very effective for photographic images where the colors can vary spatially over short distances ("grainy" images). JPEG offers some dramatic compression in filesize, sometimes by a factor of 10 (e.g. a 1500 kb file reduced to 150 kb), which may be at a trade-off for some image quality. JPEG images do not have the ability to have transparency.

Some Points to Consider When Using Graphics

If your web pages include graphics, consider the following:

- Large and numerous images may look great on a high-end computer, but they will frustrate users who must wait for images to be sent over the network.

- Not all of the viewers have a 21-inch computer monitor! Keep graphic images no wider than 480 pixels and no higher than 300 pixels to avoid forcing users to scroll or resize their web browser window.
- Rather than displaying all of the images on the web page, have them linked as external images that are downloaded only when a viewer clicks on a hypertext item   If you have numerous pictures to display, try to break the web page into a series of linked pages.
- Many web browsers "cache" images (storing them on your computer) meaning that using the same file in several web pages will load them from the viewer's computer rather than loading them across the Internet.
- Most importantly, every time you add an image to a page, you need a good reason for doing so. Be sure that every image enhances the content, design or both.

2.7.1 HTML Tags for Inline Graphics

To add an image to the HTML document, include a single tag with a reference to the image. The image reference is called source indicated by the SRC= attribute, that points to the image file. Any valid URL can be used within the src attribute value. If the image is saved in the same folder, then simply the filename can be specified.

An "inline" image is one that appears within the text of a WWW page, such as

- this picture is an inline image

The HTML format for the inline image tag is:

```
<img src="filename.gif">
```

where filename.gif is the name of a GIF file that resides *in the same directory/folder* as your HTML document. By "inline", this means that a web browser will display the image in between text.

Note how the text immediately follows the picture above. What if we want the picture sitting on its very own line?   To force the image to appear on a separate line,

simply insert a paragraph tag before the image tag:

```
<p> <img src="filename.gif">
```

2.7.2 Alternative text

If your web pages will be viewed by users using a text-only browser (such as lynx), they will not be able to view any inline images. Or sometimes, users will turn off the loading of inline images to save time on downloading over slow network connections. An alt attribute for the <img ...> tag allows for substitution of a descriptive string of text to hold the place of the image. Alternative text describes the image that is inserted.

To add alternative text to the images simply add alt attribute to the img tag like this:

```
<img alt="Our institute logo”  src="logo.gif">
```

Now if logo.gif is not displayed by any reason, the alternative text, “Our institute logo” will be displayed in its place.

2.7.3 Height and Width attributes

Another option you may want to include in your <img...> tags are two attributes that give the dimensions of the image in pixels. Normally, web browser has to determine these dimensions as it reads in the image; the loading of your page can be faster if you specify these numbers in your HTML.

The format for including this option is:

```
<img src="filename.gif" width=X height=Y >
```

where X is the width of the image and Y is the height of the image in pixels.

You can usually use some sort of graphics program or utility to determine these numbers. Another way to find the dimensions of an image is to load it into your web browser -- you may be able to drag and drop the icon for the image into your browser window -- and the height and width will be displayed in the title bar of the browser.

2.7.4 Border

The border attribute can used to control the border around an image. To turn the borders off set border = “0”. Siiliarly, to increase the border size increase the value of the border.

Ex: `<img src="sweethome.gif"  width=600 height=200   border= “6” >`

2.8 Linking The Documents

Anchor

The real point of the Web, of course, is that documents can be linked to each other, or to other types of files such as movies or sound clips, through the use of hyperlinks. These links allow authors to link documents together in intuitive ways, as opposed to traditional linear texts such as books, articles, or almost anything else printed.

In order to create a hyperlink, you'll need to know two things. The first is the URL -- that is, the location of the file to which you want the link to go.   The second is knowledge of how links work.

The World Wide Web uses an addressing scheme known as URLs, or Uniform Resource Locators . These hypertext links, the ones usually underlined in blue, are known as anchors

The Basic Anchor

The simplest possible anchor starts with <A> and ends with </A>. It marks the text or image as a link. However, you will never ever use the <A> tag by itself, because it doesn't do anything. You'll need to enhance the <A> tag with attributes like...

HREF

HREF stands for "Hypertext REFerence," which is another way of saying, "The location of the file I want to load." Most anchors are in the form <A HREF="*URL*">, where *URL* is the location of the resource to which you want the link to point.

URL tells the browsers the file to link to. It identifies file locations on the web or your local hard drive. These addresses can be HTML documents or elements referenced by documents such as images,applets, scripts etc. The URL is always enclosed in quotes.

Thus the basic structure of a link is as follows:

< A href= " http://www.servername.com/myfolder/index.html" > My Home Page </A>

In the above example,

A is the anchor tag and href is its attribute

http://www.servername.com/myfolder/index.html is the address of the file to link to

My Home Page is the Link text. It is between the open and close of the anchor tag

</A> is the closing anchor tag

A URL (and therefore, by implication, an anchor) can point to any resource available on the Web. This is usually another HTML page, but it can also be a graphic, a sound file, a movie, or any other kind of file. This fact lets you set up links to large graphics without actually having to display them in the page. For example, if there were a graphic file called "welcome.gif" in the directory "my pictures" of a server with the address "www.site.edu," the URL would be:

```
http://www.site.edu/ my pictures /welcome.gif
```

Therefore, a text anchor referring to this graphic file would look something like:

```
<A HREF="http://www.site.edu/ my pictures /welcome.gif">See my welcome message!</A>
```

A user who selects the anchor thus created will cause his Web browser to download the graphic file, which will then be displayed by his browser or by a helper program. The same general principles hold true for referring to sound files, movie files, multimedia files, and any other non-HTML files. So if I wanted to refer to a sound file called "welcome.au" in the same directory as the welcome graphic, I might set up a link like this:

```
<A HREF="http://www.site.edu/ my pictures /welcome.au"> Hear my welcome message! </A>
```

URL Anatomy

The Uniform Resource Locator (URL) is the address of the document on the web. It is what the WWW uses to find the location of files and documents from computers on the Internet. The URL is what you will need to build a link from the web page that you are creating to connect to some other piece of information available on the Internet. On your web browser screen, the URL for this document is typically displayed in the upper part of the Web browser window. The URL includes:

- a file path to the particular item of interest.

. A URL has four parts:

Protocol, host name, folder name and file name.

Ex: http://www.site.com/emeyer/files/index.html

The protocol is the way in which page is accessed, that is the type of procol or program the browser will use to get the file. One of the most common protocols is HTTP, HyperText Transfer Protocol which indicates that information is located on an http server. To include a link to information located on an FTP server, the protocol indicator is ftp://. If the file is on the local computer, the protocol indicator would be file:///

In the above example, http is the protocol

The host name is the system on which the information is stored. Each server has a specific address and all documents stored in the server share the same host name.

www.site.com is the server name in the above example where the file is stored

Folder name indicates the folder in which the files are stored. (emeyer/files in the above case)

File name is the name of the specific file of interest that has to be displayed. Here , it is index.html

2.8.1 Link to local files

The simplest anchor link is one that opens another HTML file in the same directory as the presently displayed web page. The HTML format for doing this is:

```
<a href="filename.html"> text that responds to link </a>
```

Think of it as "a" for anchor link and "href" for "hypertext reference".

The filename must be another HTML file. Whatever text occurs after the first > and before the closing </a> symbols will be the "hypertext" that appears underlined and "hyper."

Say you have created two separate html files in the same directory. The first file is saved as aboutme.html which is the home page for your personal web site and second file as studies.html where you gave the information about your studies. Now you want to connect these two files i.e when user clicks on the word About my studies in the home page (aboutme.html) then it should display the studies.html file to give information about your studies.

To make the words About my studies as hypertext, the coding should be this way:

```
<a href= “studies.html” >  About my studies  </a>
```

This makes the words About my studies as hypertext, which when clicked by the user displays the studies.html file.

2.8.2 Anchor Link to a Graphic

With the anchor tag, you can also create a link to display a graphic file. When the anchor link is selected, it will download the image file and display the image by itself in your web browser.

The simplest anchor link is to a file in the same directory/folder as the document that calls it. The format for creating a hypertext link to a graphic is the same as for linking to another HTML document:

```
<a href="filename.gif">text that responds to link</a>
```

where filename.gif is the name of a GIF image file.

2.8.3 Links to other directories

The anchor tags can also link to an HTML document or graphic file in another directory/folder in relation to the document that contains the anchor. For example, we may wish to keep all of the graphics in a separate directory/folder called pictures. As you create more and more HTML files, keeping the image files in its own area will make things a bit more organized for you.

With HTML you can direct your web browser to open any document/graphic at a directory level *lower* (i.e. a sub-directory or folder within the directory/folder that contains the working HTML file) by using the "/" character to indicate the change to a sub-directory called "pictures."

Say our web structure is like this:

---

my site

trial.html

pictures

sweethome.gif

puppy.gif

My site is the folder created to save all the files related to the site. Pictures is the sub-folder to hold all the graphics required for the site. To link to sweethome.gif placed in the pictures sub-directory, from the trial.html file, the code is

```
<a href="pictures/sweethome.gif"> Visit my home  </a>
```

2.9 TABLES

HTML tables are grids made up of rows and columns. These rows and columns create individual cells. Cells can contain either text or images. Tables can more effectively portray the information than paragraphs. They help in presenting the complicated data in a readable format. Tables are containers, so a truly minimal (if useless) table would have the following markup:

```
<TABLE>
</TABLE>
```

All other table elements will go between those two tags.

When you're planning a table, think of it as having this basic structure:

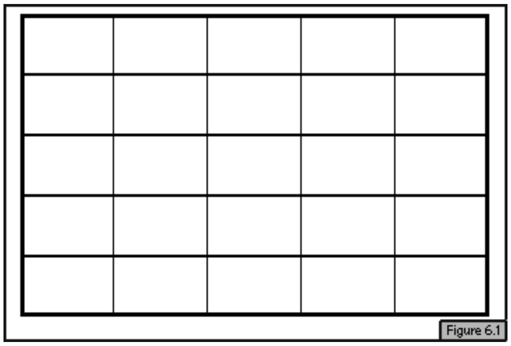


Figure 6.1

The height of each row, incidentally, is determined by the height of the tallest cell in that row. The browser itself will automatically figure out how tall that is, based on the data it has to display and the way that data has been arranged.

2.9.1 Creating Basic tables

Creating tables involves creating the table structure and entering the data in the cells. The structure of the table involves the number of rows and columns, the column headings etc

Each row is specified using the "table row" container, comprised of the tags <TR> and </TR>. Therefore, a blank three-row table would be written as:

```
<TABLE>
<TR></TR>
<TR></TR>
<TR></TR>
</TABLE>
```

This still isn't enough for a useful table, however. Each row needs to be divided into a number of cells. (cells are the individual squares in the table)

Note that there is no way to explicitly specify how many cells are in a given row-- that number is implicit in the cells themselves. In other words, if a row container has four cell containers in it, then that row obviously has four cells. Now, the question is how to specify a cell container?

There are two ways, actually, because there are two types of cells: data cells and heading cells i.e a cell can contain normal table data or a table heading.

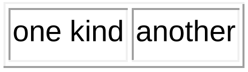
Data cells

A data cell is specified using the container <TD>...</TD>, and there can be any number of these cells in a given row. The contents of a data cell are generally referred to as "table data," which is where the letters TD come from. Any kind of information-- text, images, and so on-- is considered data for the purposes of tables. Therefore, most of your cells will contain data of one kind or another:

A data cell is specified using the container <TD>...</TD>, and there can be any number of these cells in a given row. The contents of a data cell are generally referred to as "table data," which is where the letters TD come from. Any kind of information-- text, images, and so on-- is considered data for the purposes of tables.

```
<TABLE >
<TR>
<TD>one kind</TD>
<TD>another</TD>
</TR>
</TABLE>
```

This table looks like this:

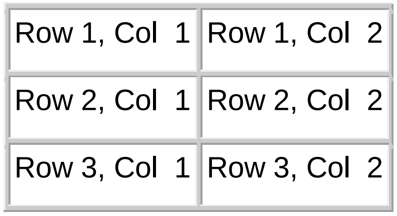


one kind	another

For example:

```
  <TABLE border = “1”>
    <TR>
      <TD>Row 1, Col 1</TD>
      <TD>Row 1, Col 2</TD>
    </TR>
    <TR>
      <TD>Row 2, Col 1</TD>
      <TD>Row 2, Col  2</TD>
    </TR>
        <TR>
      <TD>Row 3, Col  1</TD>
      <TD>Row 3, Col  2</TD>
    </TR>
  </TABLE>
```

The above code results in



Row 1, Col 1	Row 1, Col 2
Row 2, Col 1	Row 2, Col 2
Row 3, Col 1	Row 3, Col 2

Data cells can be arbitrarily large, and as indicated before, a cell can contain nearly anything: text, both regular and mono spaced; lists, both ordered and unordered; images of any size; horiztonal rules; forms

Keep in mind this concept:

Tables start being built from the top left, then across the columns of the first row, then to the second row, across the columns of the second row...

2.9.2 Changing table border

Ordinarily, a table will be rendered without any borders, so that the data is arranged in discrete cells, but the cells themselves can't be seen. This can be useful, but most of the time, it helps to be able to see the actual structure of the table. To do this, just add the word border to the TABLE tag, so that the opening tag looks like this:

```
  <TABLE border>
```

With border attribute you specify whether the border lines are displayed around the table and if so how wide the borders should be. If border has a numeric value, the border around the outside of the table is drawn with that pixel width.

```
< table border = “7” >
```

This creates a table with a border width of 7 pixels.

2.9.3 Heading cells

Table heading label the rows, columns or both.

<th> … </th> elements are used for heading cells. Headings are usually displayed in a boldface font. It should be closed with a closing </th> tag.

The heading cells appearing in the top row of a table can be defined this way

```
<tr>
   <th> Name </th>
   <th> Qualification </th>
    <th> Designation  </th>
</tr>
```

The rows of data cells can follow this heading.

The Table Header tags <th>...</th> function exactly like the <td>...</td> tags except that any text is automatically made bold and all items are automatically centered.

2.9.4 Caption

Caption indicates what the table is about. It acts as a title of the table. But captions are optional. The caption element is placed inside the <table> element just before the table row definition and has a closing tag </caption>

```
<table>
  <caption> Institute Information </caption>
<tr>
```

2.9.5 Coloring Tables

Most web browsers now support HTML to color tables, rows, and individual table cells. To change the background color of a table, a row or a cell, use the bgcolor attribute of <table>, <tr> or <td> elements. Just like in <body>, the value of bgcolor is specified either as a hexadecimal triplet or color name.

Table Part	HTML
table color the area behind the entire table	<table bgcolor= “red”>
row color the area behind a single row	<tr bgcolor= “ green”>
cell color the area behind a single cell	<td bgcolor= “ blue”>

Each background color overrides the background color of its enclosing element.

2.9.6 Setting a table background image

Newer versions of browsers support table background images. Browsers that does not support images, will display the default background color instead of the image. The background images are tiled – the image is repeated on the screen until the available background space is filled. Use the background attribute of the table to set a image in the background of the table.

<table background = " institute.gif" >

2.9.7 Alignment

We can make adjustments to the alignment of the table and the content within the cell. The align attribute aligns the content horizontally, while valign attribute aligns content vertically.

Table alignment

By default tables are left aligned to the page. The align attribute is inserted in the opening <table> tag. The values of these attribute can be left, right or center.

<table align= " left" > - this will align the table to left margin and the text following it will be wrapped in the space between the the table and right side of the page.

<table align="left" border=4 width= 600>

the above setting will give a border of four pixels width to the table, align it to left margin sets the width of the table to 600 pixels. Width can also be specified as a percentage of the width of the page.

<table align="left" border= 4 width= " 70%">

Cell alignment

The data inside a cell also can be aligned for the best effect. By default , heading cells are centered both horizontally and vertically and data cells are centered vertically but aligned left.

Within this tag you can use several attributes to control the alignment of items in a single cell:

Horizontal Alignment	Vertical Alignment
<td align=left> aligns all elements to the left side of the cell (this is the default setting)	<td valign=top> aligns all elements to the top ol the cell
<td align=right> aligns all elements to the righ side of the cell	<td valign=bottom> aligns all elements to the bottom of the cell
<td align=center> aligns all elements to cente of the cell	<td valign=middle> aligns all elements to the middle of the cell (this is the default setting)

You can combine these attributes:

<td align=left valign=bottom>

This HTML will produce a cell with items aligned along the left and bottom of the cell.

2.9.8 Cell size

To specify the width attribute on individual cells, use width attribute with <th> or <td> tags. Width can be specified in exact pixel width or as a percentage of the full table width.

```
<table border = “1”  width= “100%” >
 <caption> Institute Information </caption>
     <tr>
   <th width= “40%”> Name </th>
   <th width = “30%”> Qualification </th>
    <th width = “30%”> Designation  </th>
</tr>
```

In the above example, the table spans 100% of screen width. The first name column occupies 40% of the table width and the rest two columns 30% each.

2.9.9 Cell spacing and Cell padding

Cell spacing refers to the amount of white space between the cells and cell padding refer to the amount of space between the edges of the cells and the content of the cell. The default values for cell spacing is 2 pixels and for cell padding 1 pixel. Anyhow, if you want to increase these spaces then add cellspacing and cellpadding attributes to <table element and specify the value in pixels as shown below:

```
<table border = “2”  cellspacing =  “6”  cellpadding =  “5”>
```

2.10 Designing Forms

Forms are used to add another level of interactivity to your web pages, to allow communication between your viewers and your web site, to gather information, and to offer different means of navigation.

The role of forms is to gather different kinds of user input, i.e. fields to type in text, menus to select items from, radio buttons to choose items. The web browser takes this information, and wraps it up into a packaged format that can be sent directly to a web server, where there is a customized program sitting and waiting for the form information. These programs can unpackage the information, manipulate it, store data, and send a feedback page back to the viewer.

The form is part of the web page that you create using HTML elements. Each form contains form element that has special controls like textboxes, checkboxes, radio buttons, submit button etc. These controls make the user interface for the form. By interacting with these controls, users fill up the form. For ex- through a form and its controls you can get information from the user – the user’s name, his e-mail id , hobbies, how he liked your site etc., through different form controls. Once he or she enters the information or makes choices, submits the form. Then, Form identifies the controls with in the form that contain data and builds a form data set to contain it. The data set is encoded and sent to the Web server to be processed.

All of the elements inside the <FORM> tags can send some information about their contents and whether or not they have been activated by the web page viewer. Each element includes a defined "type" plus a unique indentifying name for that element. When the form data is sent to the web server, each element sends its name and its current state or value.

The FORM Tag

How does the browser know where a form begins and ends. For that matter, how does the browser know where to send the data? where is that program located which will handle the form data? The data has to be sent to a specific location.

This is accomplished by using the <FORM> tag. This tag has two attributes which must be used if the form has to work correctly. The attributes are METHOD and ACTION.

Here's what an empty form would look like:

```
<FORM method="post" action="/cgi-bin/program1">

</FORM>
```

METHOD has two possible values: GET and POST.

The ACTION attribute contains the URL of the CGI program which processes the data sent by the browser. In the example above, the program (program1) resides in the cgi-bin directory of the server which contains the form itself. The value of ACTION can be either a relative or a full URL.

Any tag which is allowed inside of the <BODY> container is allowed inside a form. Headings, paragraphs, lists, tables, images, links-- anything and everything goes. In addition, there are certain tags which are allowed to exist inside a form, and nowhere else. These special HTML elements -form controls are used in a form to gather information from the user visiting your site. This information is sent to the location of the action attribute when the form is submitted. The input element enables you to create many different types of controls- textboxes, checkboxes, radio buttons etc., with in the form

Text Input Elements (type="text")

Text controls enable you to gather information from a user in small quantities. This creates a single line text input field in which users can type information such as name, email id etc. The default width is 20 characters, and you can create fields of other sizes by the value in the size option. You can limit the number of characters by the value of the MAXLENGTH option. Text input fields will be empty when the page loads, unless you provide an initial text string for its VALUE option.

Text controls are part of input element, included in the form element. Create an input element with text as the value for the type attribute.

A text field named "text1" that is 30 characters wide.

```
<input type="text" name="text1" size="30">
```

A text field named "text2" that is 30 characters wide but will only accept 20 characters.

```
<input type="text" name="text2" size="30" maxlength="20">
```

A text field named "text3" that is 40 characters wide with default value.

```
<input type="text" name="text3" size="40" value="We are not alone">
```

We are not alone

Password Input Elements (type="password")

Password inputs are strikingly similar to text inputs, in that they accept any input from the keyboard, they can have SIZE and MAXLENGTH attributes, and (as always) they require names The difference is that when the user types in a password field, the computer displays bullets or asterisks instead of the characters being typed.   To create a password control, create an input element and assign password to the type attribute.

A password field named "pass1" that is 30 characters wide.
<input type="password" name="pass1" size="30">

A password field named "pass2" that is 30 characters wide but will only accept 20 characters.
<input type="password" name="pass2" size="30" maxlength="20">

A password field named "pass3" that is 40 characters wide with default value.
<input type="password" name="pass3" size="40" value="We are not alone">

Text Area Input Elements (type="textarea")

... are text fields that have more than one line and can scroll as the viewer enters more text. The rows and cols attributes sets the number of rows and columns of the text area. The name attribute establishes a label for the input field. You can also include default text to appear in the field.

A textarea field named "comments" that is 45 characters wide and 6 lines high.

```
<textarea name="comments" rows="6" cols="45" >
Please type your comments here ….
</textarea>
```

Radio buttons (type="radio")

... are sets of controls that are linked so that only one radio button among each set is selected at a time; if you click one radio button, the others in the set are automatically de-selected. A set of radio buttons is defined by providing them the same name. The value sent in the web form is the value of the radio button that was last selected. Adding the option checked to one of the buttons in a set will make that button highlighted when the page loads. This INPUT type is best used when you want the user to select one of a limited number of choices

For example, suppose you wanted to find out which computer operating system the users prefer. Of the four options provided, the user should only be able to pick one. Radio buttons can be used here like this:

Your favorite computer operating system:

○ DOS

○ Windows95

○ OS/2

○ UNIX

What happens is that only one option can be chosen. If an option is already selected, then choosing another option will de-select the previously chosen option and select the new option. Since there is nowhere for the user to enter a value, however, the value of each option must be specified in the HTML markup itself, using the VALUE attribute. When the form is submitted, this value is sent to the server if the radio button is selected.

To create a radio button, set the type of input control as “radio”

The HTML code for the above example is:

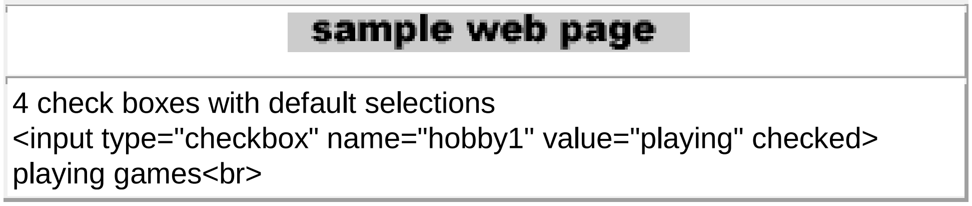
```
      <P>
    Your favorite computer operating system:<BR>
      <INPUT type="radio" name="fav_os" value="dos">DOS<BR>
     <INPUT type="radio" name="fav_os" value="win95">Windows95<BR>
     <INPUT type="radio" name="fav_os" value="os2">OS/2<BR>
     <INPUT type="radio" name="fav_os" value="unix">UNIX<BR>
 </P>
```

it is vital that each of the INPUT tags in a radio-button logical input have the same name. In the above example, all radio buttons have the same name – “fav_os”. Otherwise, the browser has no way of knowing that the different INPUTs are all part of the same logical input. When you select a radio button, the browser checks to see if any other radio buttons with the same NAME are selected. If so, it de-selects that button for you automatically.

So, in the above example, assume that user has selected the second option, "Windows 95." The value of fav_os would then become win95. If he changes his mind and select the last option, then the value of fav_os would be unix . If no option is selected, then the value of fav_os would be nothing (not zero, but literally nothing).

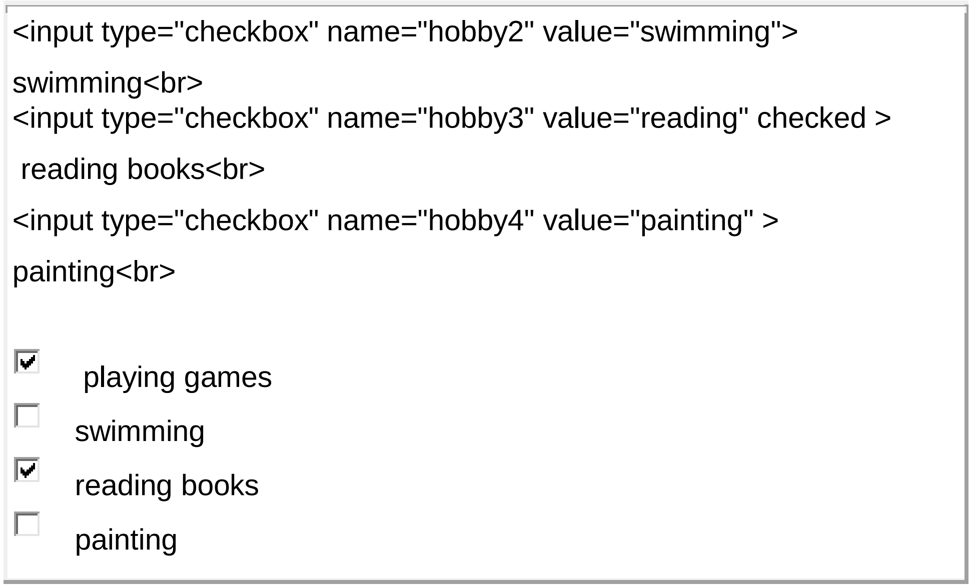
Checkboxes

The HTML markup for a checkbox logical input looks very similar to that for radio buttons. Check boxes appear as little squares that contain checkmarks inside it when selected. To create a check box, set the type attribute of input element to “checkbox”. Unlike radio buttons, checkboxes are not affected by other buttons. So you can have more than one in a group checked at a time. Note that every checkbox can have a unique name. If there is no check in the box, clicking it will place an X or a check mark there. If the box is checked, clicking it again will remove the mark. The value sent in the web form is the value of the checkbox if it was selected; otherwise the value will be empty. Adding the option CHECKED to a checkbox will make that checkbox highlighted when the page loads.



sample web page

4 check boxes with default selections
```
<input type="checkbox" name="hobby1" value="playing" checked>
playing games<br>
```



```
<input type="checkbox" name="hobby2" value="swimming">
swimming<br>
<input type="checkbox" name="hobby3" value="reading" checked >
 reading books<br>
<input type="checkbox" name="hobby4" value="painting" >
painting<br>
```

☑ playing games

☐ swimming

☑ reading books

☐ painting

Once again, VALUE is required for each INPUT tag, because the user has no way to input any values himself-- merely to select from a list of options. In a checkbox logical input, the user can select some, or all, or none of the options provided. When this form is submitted to the script for processing, each checkbox that is checked returns a value associated with the check box.

You can even group check boxes together and assign the same control name. This allows multiple values to be chosen and applied to one property. let's assume that we want to know which OS the users have used. Thus:

```
<P>
What operating systems have you used?<BR>
<INPUT type="checkbox" name="os_used" value="mac">Macintosh<BR>
<INPUT type="checkbox" name="os_used" value="dos">DOS<BR>
 <INPUT type="checkbox" name="os_used" value="win95">Windows95<BR>
<INPUT type="checkbox" name="os_used" value="os2">OS/2<BR>
<INPUT type="checkbox" name="os_used" value="unix">UNIX<BR>
</P>
```

Once again, VALUE is required for each INPUT tag, because the user has no way to input any values himself-- merely to select from a list of options. In a checkbox logical input, the user can select some, or all, or none of the options provided.

What operating systems have you used?

☐ Macintosh

☐ DOS

☐ Windows

□ Windows95

□ OS/2

□ UNIX

So how do multiple responses get transmitted if there's only one value allowed for a given name? Let's assume that the user checks the boxes for DOS, Windows95 and Unix. The value of os_used would be dos|win95|unix, where the | represents a separator (usually a null character). There is one value for the NAME defined as os_used, but it contains all of the options which the user selected. The program to which we are sending the form data, need to be able to take the value and split it up into its components.

Submit and Reset (type="submit" and type="reset")

... creates buttons on the form. The Submit button tells the web browser to gather up all the selections, values, and entered text in the form elements and send it off to the place defined in the ACTION part of the form tag. The Reset button restores the form to its default state, how it looked when the viewer first entered the page. The VALUE option defines the text string that is placed on these buttons.

```
<INPUT type="submit">
```

...will create a button on the screen which says something like "Submit Query" (the actual label may vary). Selecting a Submit button triggers the posting of the input data to the CGI program.

Similarly, the markup...

```
<INPUT type="reset">
```

...places a button which is labelled something like "Reset Form" (again, the actual label will vary). Selecting the button will cause the entire form to be reset to its default state, wiping out any and all changes made by the user.

Usually, the two buttons are placed together at the bottom of a form, but there is nothing which says they have to go together, or at the bottom of the form. That's a matter of custom more than anything else. You can change the labelsof the buttons by using a VALUE attribute. For example:

```
<INPUT type="submit" value="Send This report">
<INPUT type="reset" value="clear">
```

2.11 Model questions

1. Compare the differences between using the <br> and <p>
2. Use the <p>, the <hr>, or the <br> tags to create paragraphs or sections in your own document.
3. What is the basic structure of an URL?
4. Add an inline image to your web page using a GIF picture file that is stored on your computer or one that you have downloaded from the Internet.
5. What is the structure of a web page form?
6. What are the differences in function and HTML coding for radio buttons and check boxes?

7. Create a form on the web page to collect personal information of the user. Use appropriate form elements.

2.12 Quiz

1. The tags with which unordered list begins and ends

   <UL> and </UL>

2. The tag used in a table to specify the data

   <td>

3. The click of which button on form sends its contents to a file specified in Action attribute

   Submit

4. A table on a web page can not have a different background from the Web page background T/ F

   F

5. The href attribute of anchor tag refers to

   Hypertext Reference

6. Which attribute for the <img ...> tag allows for substitution of a descriptive string of text to hold the place of the image

   alt

7. Which INPUT type is best used when you want the user to select only one choice out of a limited number of choices on the Form

   Radio button

8. Any tag which is allowed inside of the <BODY> container is allowed inside a form also T/ F

   T

9. URL stands for

   Uniform Resource Locator

10. Which tag is used to define the cells appearing in the top row of a table as heading

    <th>

Lesson 3

CASCADING STYLE SHEETS

Structure

3.0 Learning Objectives:

This chapter will cover following topics:

- Introduction to CSS
- Features of CSS
- Basic programming using CSS
- Types of CSS
- Summary

3.1 Introduction to CSS:

CSS stands for Cascading Style Sheets.*Cascading Style Sheets are widely-used in DHTML*for making interactive and dynamic web pages and websites. DHTML stands for Dynamic Hyper Text Markup Language. CSS is generally used to give style to the HTML documents. CSS can be used to define the layout of webpages and websites. CSS can be used to handle the style of the webpages and website in a simple and efficient way.

3.2 Features of CSS:

The main features of CSS are following:

Less source code- you don't need to write html tags again and again, if you are using cascading style sheets. This leads to less source code.

Changes are easy-You can easily make a global change by simply changing the style. All the pages of document will be changed by itself.

Dynamic and Interactive WebPages-CSS can be usedfor making dynamic and interactive Web pages.CSS can generate dynamic page contents in a web site as compare to HTML which can develop only static websites. We can easily develop dynamic and interactive web pages using CSS.

Layout of multiple web-pages in one go-The layout of multiple web-pages can be changed simultaneously by using cascading style sheets

Simple and Efficient-Cascading style sheets are very simple and straight forward. Even a newcomer with basic knowledge of HTML tags can learn CSSeasily. CSS is easy to learn and use. The efficiency of HTML increase with the use of cascading style sheets.

Case Sensitive-CSS is case insensitive as HTML. There is a no difference between uppercase and lowercase characters in CSS. The code can be written in uppercase as well as lowercase.

Maintenance - After making a website, if a change is required in style or look, itcan be easily done by making a global change using cascading style sheet. All the web-pages of the website will be updated by changing in code.

3.3 Default syntax of Cascading Style Sheet

The layout of HTML web page is defined by the CSS rule.

- **CSS Rule**

  The general syntax of a Cascading style Sheet rule is:

HTML element1, HTML element2…. { property1: property value1; property 2:property value2;…..}

Where HTML element1, HTML element2…. are the HTML elements where style have to be applied,

Property1, property2 …… are the properties and property value1, property value2…..are the values of the properties.

For example:

H1, H2, H3 {color: green; text align: left;}

Where

- H1,H2,H3 are the HTML elements where style have to be applied.
- Color and text align are the properties.

- Green and left are the property values.

First Program of CSS:

```
<html>
<head>
<title>first program of cascading style sheets</title>
<style>
        h1,h2,h3 {
               color: red;
                         }
</style>
</head>
<body>
<h1>Good Morning</h1>
<h2>Welcome</h2>
<h3>Very first program of CSS</h3>

</body>
</html>
```

You have to save this program with .htmlextension.This program will display all the three headings of the webpage in the red color.

Note: style tag is used to give the style to the document on the web page. It is generally used in the head section of the HTML document.

The output will be

```
Good Morning

Welcome

Very first program of CSS
```

Second Example:

```
<!DOCTYPE html>
<html>
            <head>
            <style>
            body {
                  background-color: red;
             }
h1,h2 {
  color: blue;
  text-align: left;
}
p {
  font-family: Ariel;
  font-size: 15px;
}
</style>
</head>
<body>

<h1>Second example of CSS</h1>
<h1>Bio-data</h1>
<p>I am Mohan Kumar</p>
<p>I am a student of ADCA</p>

</body>
</html>
```

You have to save this program with .htmlextension. This program will display all the headings in blue color with left alignment and the paragraph text will in Ariel font with 15 font-size.

Note: style tag is used to give the style to the document on the web page. It is generally used in the head section of the HTML document.

The output will be

Second example of CSS

Bio-data

I am Mohan Kumar

I am a student of ADCA

3.4 Types of Cascading Style Sheets:

Cascading Style Sheets can be of following three types.

1. Inline Style Sheets
2. Internal style sheets
3. External Style Sheets

Inline style sheet: In the case of inline style sheet, the style is applied to a single element. The style attribute is added to the HTML element, where style is to be applied.

The following program will illustrate the use of inline style sheets.

```
<!DOCTYPE html>

<html>

<head>

<title>Program of inline style sheet</title>

</head>

<body>

<h1 style="color:brown;text-align:left;">Resume</h1>

<h2 style="color:green;text-align:left;">Personal information</h2>

<p style="color:yellow;">I am Anuj Kumar. I am a student of ADCA. </p>

</body>

</html>
```

Note:

- contents of heading 1 will be displayed in brown color with left alignment.
- contents of heading 2 will be displayed in green color with left alignment.
- contents of paragraph will be displayed in yellow color.

The output will be:

Resume

Personal information

Iam Anuj Kumar. I am a student of ADCA.

Internal style sheet: In the case of internal style sheet, the style is applied to a single document. The style is given by using <STYLE> tag in the <HEAD> section of the HTML document.

The following program will illustrate the use of internal style sheets.

```
<!DOCTYPE html>
<html>
<head>
<title>Program of internal style sheet</title>
<style>
body {
background-color: White;
}
h1,h2 {
  color: green;
}
p
{
color: blue;
}
</style>
</head>
<body>
<h1>Student data</h1>
<h2>ADCA CLASS</h2>
<p>There are 70 students are in ADCA. The course is run by DCSA</p>
```

```
</body>

</html>
```

Note:

- The background color will be white.
- Contents of heading 1 and heading 2 will be displayed in green color.
- Contents of paragraph will be displayed in blue color.
-

The output will be:

Student data

ADCA CLASS

There are 70 students are in ADCA. The course is run by DCSA

External style sheet: In the case of External style sheet, the style can be applied to the multiple files.

We have to make two files.

- One CSS file with extension .CSS
- Second HTML file where style is to be applied.

The following programs will illustrate the use of internal style sheets.

1. CSS File

```
Body
{
Background-color:white
}
```

Note: You have to save this file as abc.css.

2. **HTMLFile**

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Program of External  style sheet</title>
<link rel="stylesheet" href="abc.css">
</head>
<body>
<h1>Department of Computer Science and Applications</h1>
<p>DCSA runs mainly four courses</p>
<p>MCA
<p>MSc
<p>ADCA
<p>PhD
</body>
</html>
```

Note:

- The background color will be white as defined in abc.css

The output will be:

Department of Computer Science and Applications

DCSA runs mainly fourcourses

MCA

MSc

ADCA

PhD

3.5 Summary:

Cascading Style Sheets are widely-used for making interactive and dynamic web pages. There are three types of CSS-Inline, Internal and External CSS.In the case of inline style sheet, the style is applied to a single element. The style attribute is added to the HTML element, where style is to be applied. In the case of internal style sheet, the style is applied to a single document. The style is given by using <STYLE> tag in the <HEAD> section of the HTML document.In the case of External style sheet, the style can be applied to the multiple files. For implementing external CSS, We have to make two files. First CSS file with extension .CSS and second HTML file where style is to be applied.

3.6 Objective types question/answers to check your progress

Q1: How many file are required for external CSS?

Ans: Two files are required for external CSS. To define the style .css file is required and one HTMLfile for creating webpage.

Q2: CSS stands for.

Ans: Cascading Style sheets

Q3: CSS is used for.

Ans: CSS is used to give style and look to Webpages and websites.

Q4: Write the extension used to save the external cascading style sheet file.

Ans: . css

Q5: What is Inline CSS?

Ans: In the case of inline style sheet, the style is applied to a single element. The style attribute is added to the HTML element, where style is to be applied.

Q5: What is Internal CSS?

Ans:In the case of internal style sheet, the style is applied to a single document. The style is given by using <STYLE> tag in the <HEAD> section of the HTML document.

Q7: What is External CSS?

Ans: In the case of External style sheet, the style can be applied to the multiple files. For implementing external CSS, We have to make two files. First CSS file with extension .CSS and second HTML file where style is to be applied.

3.7 Book References:

1. HTML and CSS : The Complete Reference, by Thomas A. Powell, Publisher:McGraw-Hill
2. CSS: The Definitive Guide by Meyer Eric A.; Publisher: O'Reilly Media, Inc, USA

3.8 Web References:

1. https://en.wikipedia.org/wiki/css
2. https://www.w3schools.com/css
3. https://www.tutorialspoint.com/css
4. https://www.javatpoint.com/css-tutorial

3.9 Suggested Readings:

1. HTML, CSS & JavaScript Web Publishing in One Hour a Day, Sams Teach Yourself: Covering HTML5, CSS3, and jQuery by Laura Lemay , Rafe Colburn , Jennifer Kyrnin, Sams Publishing
2. Start Programming Using HTML, CSS, and JavaScript by Iztok Fajfar

3.10 Model Questions

Q1: What do you understand by cascading style sheet?Describe the features of CSS.

Q2: What is CSS? Explain different types of CSS with the help of proper examples.

Q3: Explain the applications of cascading style sheets in web designing.

Q4: What is inline cascading style sheet? Write a Program to illustrate the use of inline CSS.

Q5: What is internal cascading style sheet? Write a Program to illustrate the use of internal CSS.

Q6: What is external cascading style sheet? Write a Program to illustrate the use of external CSS.

****

Lesson-4

JAVA SCRIPT AND ITS OBJECT

Structure

4.0 Introduction

What is a Programming Language?

- A programming language is a set of codes that we can use to give instructions to be followed by the computer.
- Popular and well-known programming languages include Java, C++, COBOL, BASIC, LISP and more. Most popular programming languages consist of words and phrases that are similar in form to the English language.
- A well-written program will be easily readable by anyone with a little programming experience, regardless of whether they have any direct experience of the language in question. This is because modern programming languages share a large number of common concepts. In particular, they all have a notion of variables, arrays, loops, conditionals, and functions. We will meet these concepts again in more depth later in the course.
- Traditionally, programming languages have been used to write (for the most part) “stand-alone” applications. Things like Microsoft Word, Mozilla Firefox and Lotus Notes are all examples of

such applications. Once installed on a PC, these applications run without necessarily requiring any other software to be installed alongside them.

- Web Applications differ from these traditional applications in many respects, but the most striking is that they all run inside your web browser. Examples of popular web applications are things like Google, Hotmail, Flickr, GMail and any of the vast array of "weblogging" systems.
- These applications are also written using programming languages, but as a rule they are built using multiple, interdependent technologies. These technologies are easily (though not completely) broken down into two categories: server-side and client-side.
- Client-side scripts allow the developer to alter pages dynamically, and to respond to user actions immediately rather than having to wait for the server to create a new version of the page. However, there are security and cross-browser compatibility issues to be aware of, and these are often non-trivial.
- Server-side applications allow the developer to keep her code secure and secret, thus allowing for more powerful applications to be created. In addition, since the server running the code is always a known quantity, applications that run successfully in one browser will run successfully in all browsers. However, despite all this power, there is no direct way for a server-side application to alter a page without having to force the client-side to load another page. This makes it completely impractical for things like drop-down menus, pre-submission form checking, timers, and warning alerts and so forth.

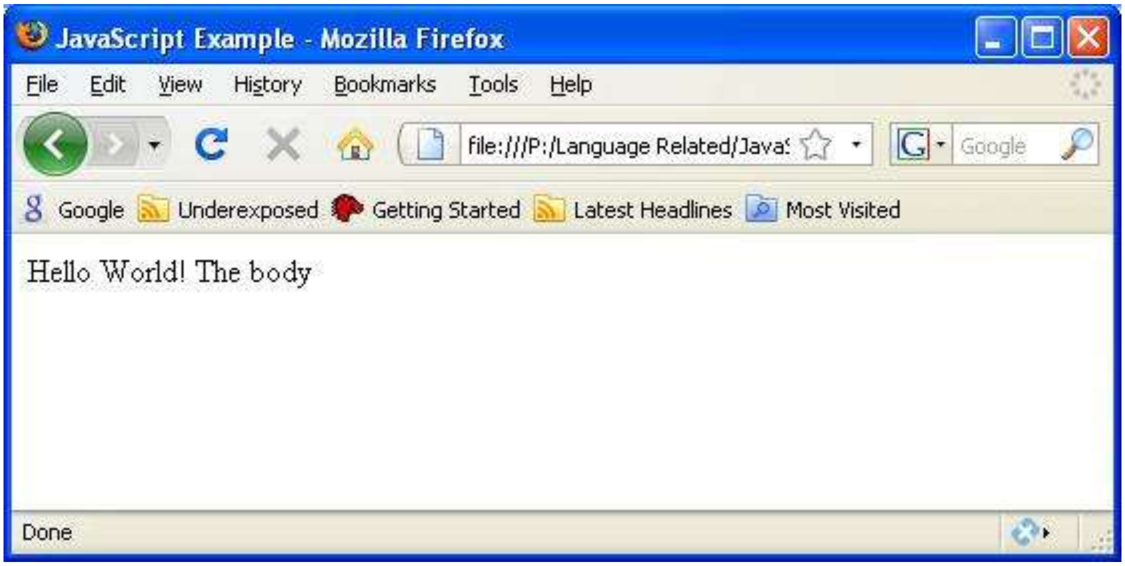
  JavaScript is a dynamic language that executes within a browser. JavaScript code is embedded within an HTML page using the JavaScript tag. The <script> tag is used to embed JavaScript code. JavaScript code can be embedded in:

  - An external file
  - The header of the page
  - The body of the page

In this example, JavaScript is embedded within the header. As soon as the page is loaded this code is executed.

```
<html>
<head>


<title>JavaScript Example</title> <script language="JavaScript 1.2"> <!--


document.write("Hello World!"); //-->
</script>
</head>
<body>The body</body> </html>
```

The Document *write* method displays the text.


JavaScript Example - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///P:/Language Related/JavaS
Google
Google Underexposed Getting Started Latest Headlines Most Visited
Done

```
Hello World! The body
```

Notice that the JavaScript code is enclosed in HTML comment tags:

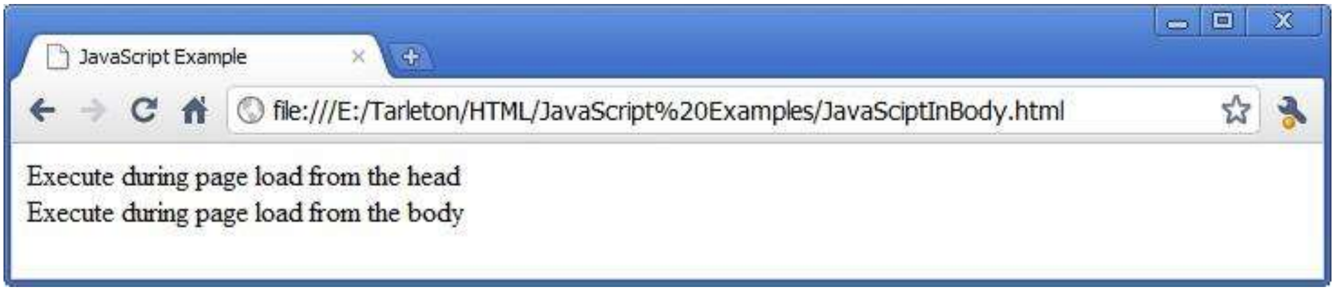
```
<!--


//-->
```

These are often used to surround JavaScript code. In older browsers JavaScript was not recognized or handled. To avoid the display of this code in a page, the browser would ignore the contents of the comment. However, in a browser that supports JavaScript the comments tags are ignored and the code is executed.

4.1 Internal JavaScript Code

JavaScript code that is not found in a function is executed as the page containing it is loaded. To illustrate this, JavaScript code is placed in the head and body section of an HTML page.

```
<html>

<head>

<title>JavaScript Example</title> <script type="text/javascript">

        document.write("Execute during page load from the head<br>"); </script>

</head>
<body>
<script type="text/javascript">
        document.write("Execute during page load from the body<br>"); </script>


</body>


</html>
```


JavaScript Example
file:///E:/Tarleton/HTML/JavaScript%20Examples/JavaSciptInBody.html

```
Execute during page load from the head
Execute during page load from the body
```

JavaScript code found in a function is not executed until the function is called. If we modify the previous example by adding a function to return a string, the function is not loaded when the page is loaded.

```
<html>
```

```
<head>

<title>JavaScript
Example</title>                    <script
type="text/javascript">
function displayString() {
        return "<h1>Main Heading<h1>"
}

        document.write("Execute during page load from the head<br>");
</script>

</head>

<body>
<script type="text/javascript">
        document.write("Execute during page load from the body<br>"); </script>
</body>

</html>
```

The output will be the same.

4.2 Functions

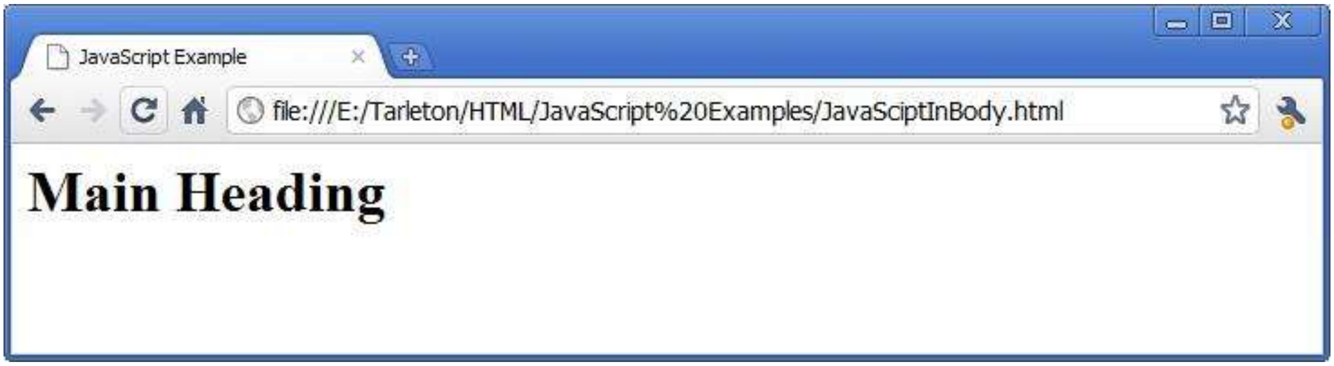
A function consists of the function keyword followed by the name of the function , a set of open and close parentheses enclosing an optional parameter list and a body enclosed in a set of curly braces.

```
        function functionName(parameterList) { // body
        }
```

A function uses the return keyword to return a value from a function.

```
<html>
<head>
<title>JavaScript Example</title>
```

```
<script type="text/javascript">
function getHeader() {
        return "<h1>Main Heading</h1>"
}
</script>
</head>

<body>
<script type="text/javascript">
document.write(getHeader()); </script>
</body>

</html>
```

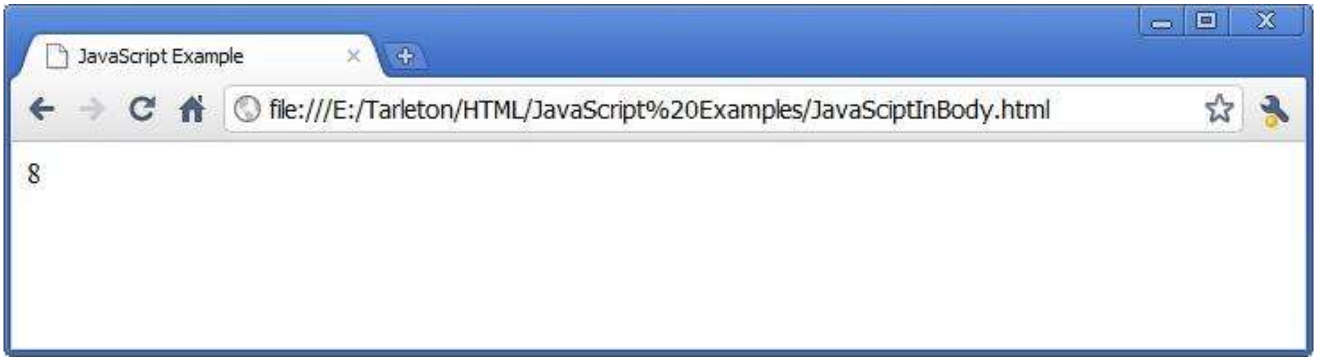

JavaScript Example
file:///E:/Tarleton/HTML/JavaScript%20Examples/JavaSciptInBody.html

Main Heading

Parameters are separated by commas in the function declaration.

```
<html>
<head>
<title>JavaScript Example</title> <script type="text/javascript"> function multiply(num1, num2) {
        return num1*num2;
}
</script>
</head>
<body>
```

```
<script type="text/javascript"> document.write(multiply(2,4)); </script>
</body>

</html>
```



file:///E:/Tarleton/HTML/JavaScript%20Examples/JavaSciptInBody.html

```
8
```

4.3 External JavaScript Code

It is advantageous to group common functions in an external JavaScript file. This permits the reuse of the functions in the file in multiple HTML pages.

JavaScript functions are stored in a file using the .js extension. If we placed the following functions in a file named scripts.js we can reference and subsequently use the functions from an HTML page.

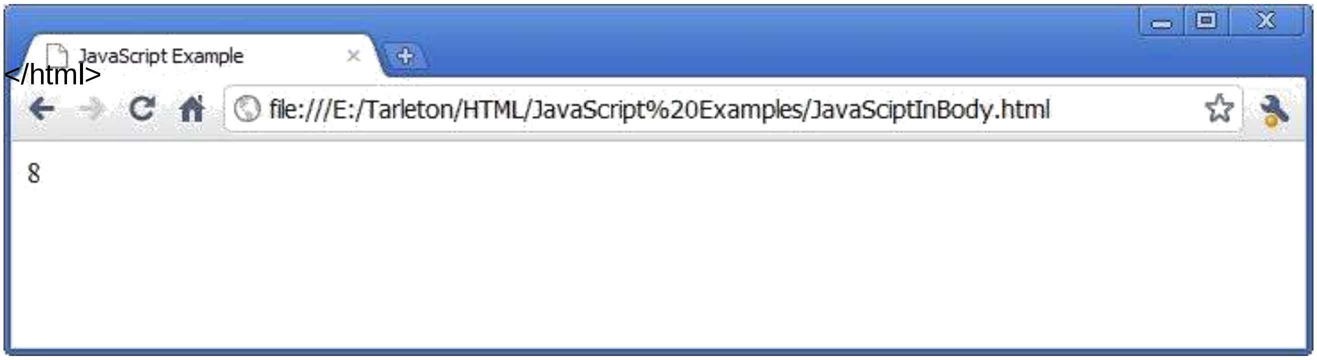
```
// functions.js function
getHeader() {
        return "<h1>Main Heading</h1>"
}
function multiply(num1, num2) {
        return num1*num2;
}
```

Notice that the C++ style comment can be used in JavaScript. Also notice that the <script> tag is not and should not be used in a JavaScript file.

In the HTML file, the <script> tag can also be used to indicate the location of a JavaScript file. The *src* attribute is assigned the path and filename of the file.

```
<html>
<head>
<title>JavaScript Example</title>
```

```
<script type="text/javascript" src="functions.js"> </script>
</head>

<body>
<script type="text/javascript">
document.write(multiply(2,4)); </script>
</body>
</html>
```


JavaScript Example
file:///E:/Tarleton/HTML/JavaScript%20Examples/JavaSciptInBody.html

```
8
```

4.4 <script> Attributes

There are two attributes of the <script> tag that are of immediate interest:

- type – The value assigned to this attribute specifies the scripting language
- src – The location of an external scripting file

The *src* attribute specifies that the code is actually found in a file which should be loaded and then executed. The .js extension is normally used for JavaScript code files. The following example illustrates the use of these attributes.

```
<html>

<body>

<script type="text/javascript" src="corefunctions.js"> </script>

</body>

</html>
```

4.5 JavaScript Language Elements

It is useful to discuss JavaScript in terms of language elements including:

- Variables
- Operators

- Expressions
- Statements

- Objects
- Functions and methods

Variables

Variables are used to hold data. A JavaScript identifier:

- Starts with a letter or underscore, and
- Is followed by letters, underscore or digits

JavaScript is a case-sensitive language

Scope

The scope of an identifier is either

- Global – An identifier that is accessible anywhere on the page Local – Is accessible only within the function it is declared within
- A global variable is typically declared simply by assigning a value to it.

```
globalVariable = 100;
```

A local variable is declared within a function using the var keyword.

```
function someFunction() { var counter = 0; globalVariable = 100;
}
```

The identifier, *counter*, is local to the function and can only be used in that function. However, the identifier, *globalVariable*, is not preceded by the *var* keyword and is thus a global variable that can be used anywhere on the page, inside or outside of the function.

4.6 Data Types

There are six data types in JavaScript :

- Numbers – Integer or floating point numbers
- Booleans – Either true/false or a number (0 being false) can be used for boolean values Strings – Sequence of characters enclosed in a set of single or double quotes
- Objects – Entities that typically represents elements of a HTML page Null – No value assigned which is different from a 0
- Undefined – Is a special value assigned to an identifier after it has been declared but before a value has been assigned to it

JavaScript is a dynamically typed language. The data type of the identifier is not assigned when the identifier is declared. When a value is assigned to the identifier the identifier takes on that type. The data type of the variable is not important until an operator is applied to the variable. The behavior of the operator is dependent of the data type being acted upon.

For example:

var name = “Sally” name = 34

The string, Sally, is first assigned to the variable. Next, the integer 34 is assigned to the variable. Both are legal but usage of the identifier is inconsistent. It is better if we are consistent when assigning a data type to a variable. This leads to less confusing code.

4.7 Literals

Literals are simple constants such as:

34

3.14159

“frog beaks”

„/nTitle/n“ true

For string, escape sequence can be used to embed special values. An escape sequence consists of the back slash character followed by a character that has special meaning. Escape sequences recognized by JavaScript include:

Character	Meaning
\b	backspace
\f	form feed
\n	new line
\r	carriage return
\t	tab

\\	backslash character
\"	double quote
\“	Single quote
\ddd	Octal number
\xdd	Tow digit hexadecimal number
\xdddd	Four digit hexadecimal number

4.8 Operators

The JavaScript operators include:

Precedence	Operator	Associativity	Meaning
1	member	Left-to-right	.
			[]
	New	Right-to-left	New
2			

1			.
			[]
	new	Right-to-left	new
2	function call	Left-to-right	()
3	++	n/a	Increment by 1

	--		Decrement by 1
4	!	Right-to-left	logical not
	~		bitwise not
	+		unary plus
	-		unary minus
	typeof		type of
	void		void
	delete		delete
5	*	Left-to-right	Multiplication
	/		Division
	%		Modulo division
6	+	Left-to-right	addition
	-		subtraction
7	<<	Left-to-right	shift left
	>>		shift right
	>>>		arithmetic shift right

8	>	Left-to-right	Greater than
	>=		Greater than or equal
	<		Less than
	<=		Less than or equal
9	= =	Left-to-right	equality
	!=		not equal
	= = =		strict equality
	!= =		strict inequality
10	&	Left-to-right	bitwise and
11	^	Left-to-right	bitwise xor
12	|	Left-to-right	bitwise or
13	&&	Left-to-right	logical and
14	||	Left-to-right	logical or
15	(condition)?value1:value2	Right-to-left	tertiary operator
16	=	Right-to-left	assignment
	+=		

	-=		
	*=		
	/=		
	%=		
	<<=		
	>>=		
	>>>=		
	&=		
	^=		
	|=		
17	,	Left-to-right	comma operator

4.9 Arrays

Arrays are allocated using the *new* keyword.

```
names = new Array(10);

numbers = new Array(5);
```

Array indexes start at 0 and extend to the size of the array minus 1. To assign a value to an element of an array open and close brackets are used.

```
names[0] = "Rabbit";

names[1] = "Happy";

names[9] = "Dover";
```

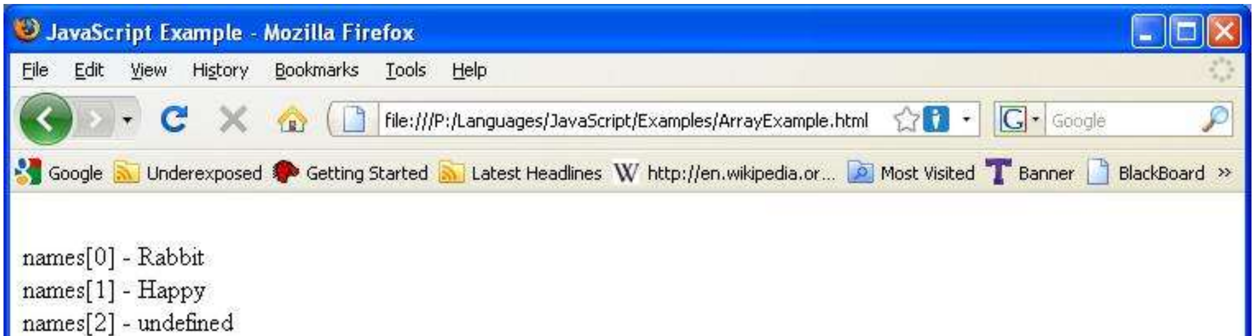
The size of an array can be increased dynamically by assigning a value to an element pass the end of the array. Array can be created that initially has no elements at all. In addition, they are not of a fixed size but can grow dynamically.

```
        pictures = new Array();

        pictures[35] = "Mona Lisa";
```

The array, pictures, initially has no elements. After "Mona Lisa" has been assigned the array has 36 elements. The unassigned elements are set to Undefined.

The *length* property of arrays returns the number of elements in the array.

```
<html>
<head>
<title>JavaScript Example</title>
<script language="JavaScript1.2"> <!--
names = new Array(10);
names[0] = "Rabbit";
names[1] = "Happy";
names[9] = "Dover";

document.write("<br>names[0] - " + names[0] );
document.write("<br>names[1] - " + names[1] );
document.write("<br>names[2] - " + names[2] );
pictures = new Array(); pictures[35] = "Mona Lisa";
document.write("<br>pictures[35] - " + pictures[35] );
document.write("<br>pictures[30] - " + pictures[30] ); document.write("<br>pictures.length - " +pictures.length); //-->
</script>
</head>
<body>
</body>
</html>
```



```
names[0] - Rabbit
names[1] - Happy
names[2] - undefined
```

4.10 Converting Between Data Types

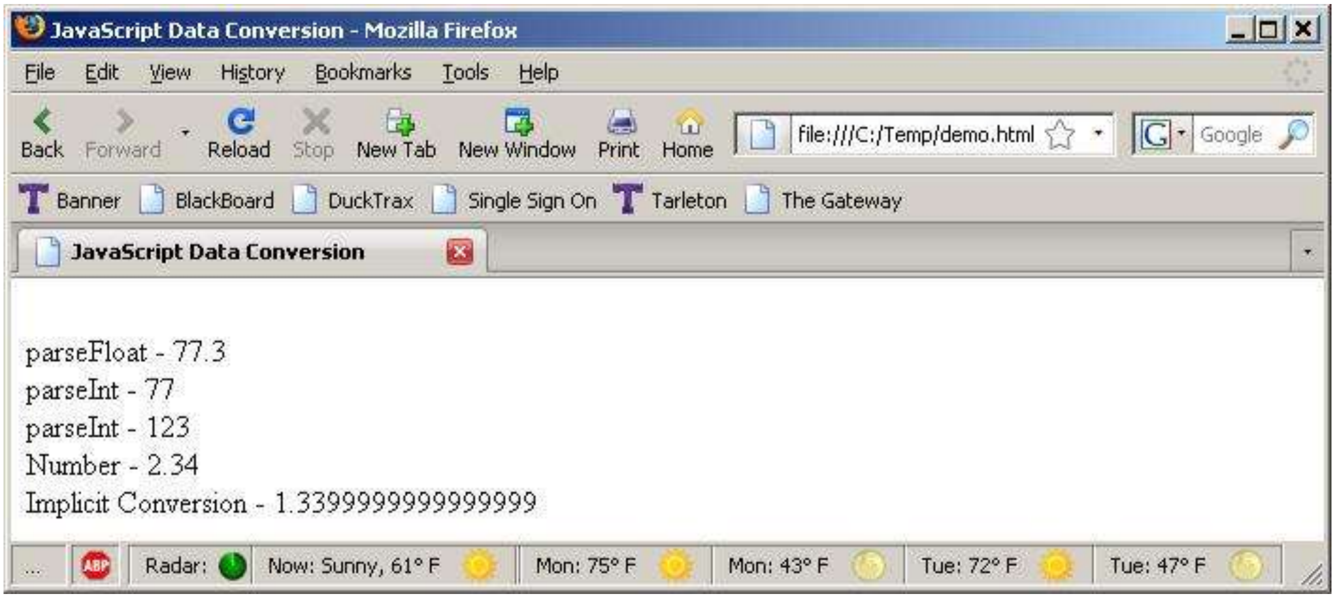
There are a number of techniques for converting between data types. To convert from a string several parse and other functions are available.

- parseFloat – Converts a string to a
- float parseInt – Converts a string to
- an integer Number – Converts a string to a number

The last example below uses an arithmetic expression to implicitly convert the string to a number.

```
<html>

<head>

<title>JavaScript        Data

Conversion</title>      <script

language="JavaScript 1.2">

<!--

document.write("<br>parseFloat - " +

parseFloat('77.3')); document.write("<br>parseInt

- " + parseInt('77')); document.write("<br>parseInt –

" + parseInt('123.45'));

document.write("<br>Number - " +

Number("2.34")); document.write("<br>Implicit

Conversion - " + ("2.34"-1)); //-->

</script>
```

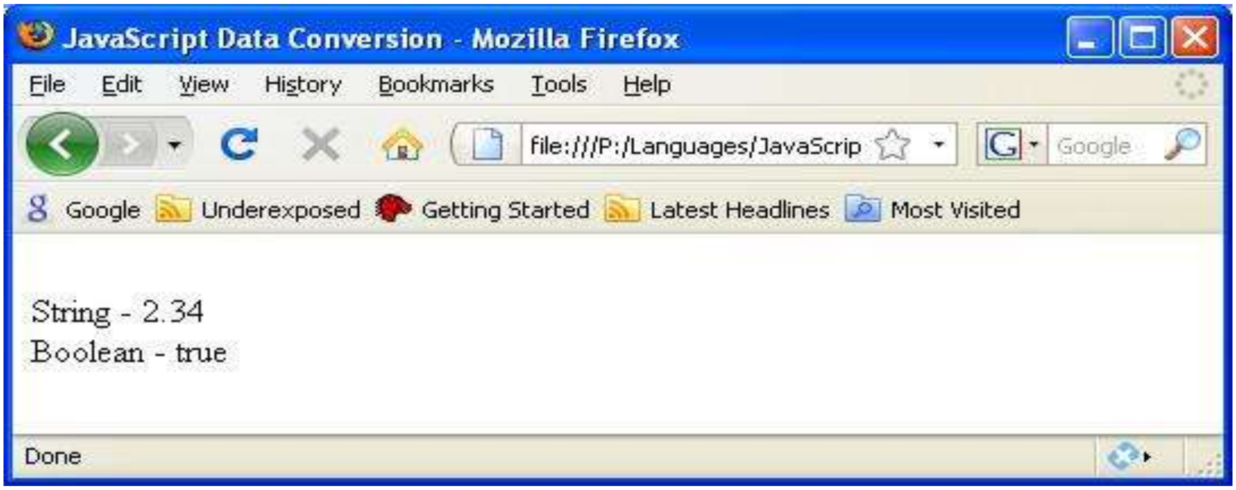
```
</head>
<body>
</body>
</html>
```


JavaScript Data Conversion - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Back Forward Reload Stop New Tab New Window Print Home
file:///C:/Temp/demo.html
Google
Banner BlackBoard DuckTrax Single Sign On Tarleton The Gateway
JavaScript Data Conversion
Radar: Now: Sunny, 61° F Mon: 75° F Mon: 43° F Tue: 72° F Tue: 47° F

```
parseFloat - 77.3
parseInt - 77
parseInt - 123
Number - 2.34
Implicit Conversion - 1.3399999999999999
```

A number can be converted to a string or Boolean using the String and Boolean functions.

```
<html>
<head>
<title> JavaScript Data Conversion </title>
<script language="JavaScript 1.2">
document.write("<br> String - " + String(2.34));
document.write("<br> Boolean - " + Boolean(2.34)); </script>
</head>
<body>
</body>
</html>
```


JavaScript Data Conversion - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///P:/Languages/JavaScrip
Google
Google Underexposed Getting Started Latest Headlines Most Visited
Done

```
String - 2.34
Boolean - true
```

4.11 Regular Expressions

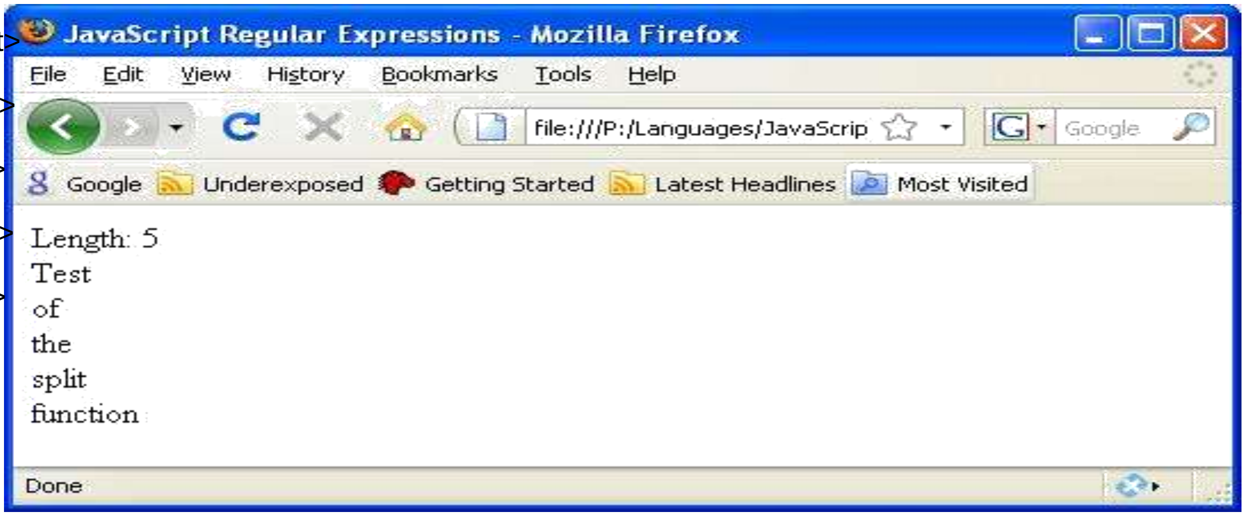
A regular expression is a way of performing pattern matching. A pattern is defined and then applied to a target string. The form of a regular expression and how they are applied to a target string varies somewhat between languages.

In JavaScript, a regular expression is defined using a series of characters that define the pattern enclosed in a pair of forward slashes. For example to match white spaces the \s is used.

```
re = /\s/g;
```

The \s means that all white spaces are to be matched and the *g* means that this needs to be applied to the entire target string. The split function can be used to illustrate this pattern. The split function is executed against a target string and will break the target up into individual string based on the split functions regular expression argument. The split function returns an array of strings.

```
<html>

<head>

<title>JavaScript Data Conversion</title>

<script language="JavaScript 1.2">

re=/\s/g;

target="Test of the split function";

result = target.split(re);

document.write("Length: " + result.length + "<br>");

for(i=0;i<result.length;i++) {

        document.write(result[i]+"<br>");
}
</script>
</head>
<body>
</body>
</html>
```


JavaScript Regular Expressions - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///P:/Languages/JavaScrip
Google
Google Underexposed Getting Started Latest Headlines Most Visited
Done

```
Length: 5
Test
of
the
split
function
```

There are several character sequences that have special meaning in a regular expression. The tutorial found at http://www.zytrax.com/tech/web/regex.htm provides an overview of regular expressions. Here we will look at only a few.

The \ is an escape sequence character which means do not treat the following character as a literal. Consider the following example:

```
…
re=/s/g;
target="Test of the split function"; result
= target.split(re);
…
```

JavaScript Regular Expressions - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///P:/Languages/JavaScrip
Google
Google Underexposed Getting Started Latest Headlines Most Visited

The split function split the target based on the presence of the letter s. The \s in the previous example treated the s as a special character which represented white spaces. Other escape sequences include:

Escape Sequence	Meaning
\d	Any digit in the range 0-9
\s	White space
\w	Any character in the range 0-9, A-Z and a-z
\b	Match any character at the beginning of a word

These escape sequences are case sensitive. An upper case letter for these escape sequences generally means NOT. That is for \D match any character not in the range 0-9.

Metacharacters also convey special meaning in a regular expression.

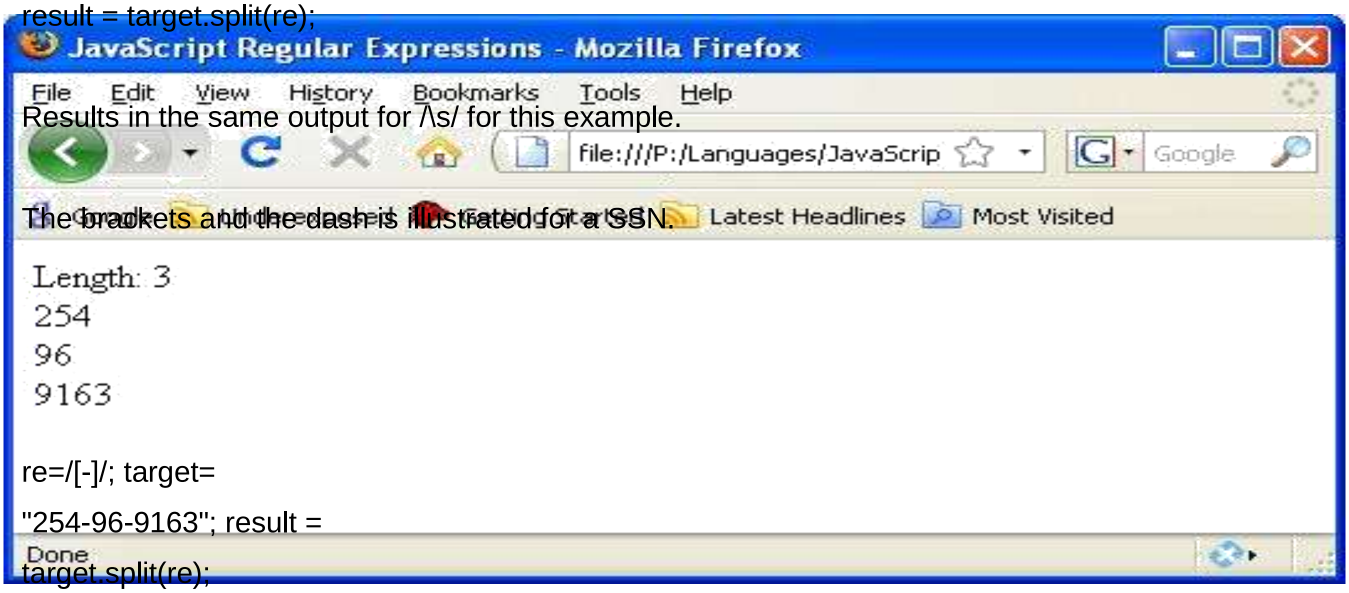
Metacharacter	Meaning
[ ]	Match any character within the brackets
-	Is used within brackets to indicate a range [a-d]
^	When used within braces it means negation
^	When used outside of a set of brackets it means to match only at the beginning of a target ^First
$	Means to only match at the end of a target [word$]
.	Match any character at that position [ton.]

Using the regular expression:

```
re=/[ ]/;
target="Test of the split function";
result = target.split(re);
```

Results in the same output for /\s/ for this example.

The brackets and the dash is illustrated for a SSN.



```
Length: 3
254
96
9163
```

```
re=/[-]/; target=
"254-96-9163"; result =
target.split(re);
```

4.12 Regular Expression Functions

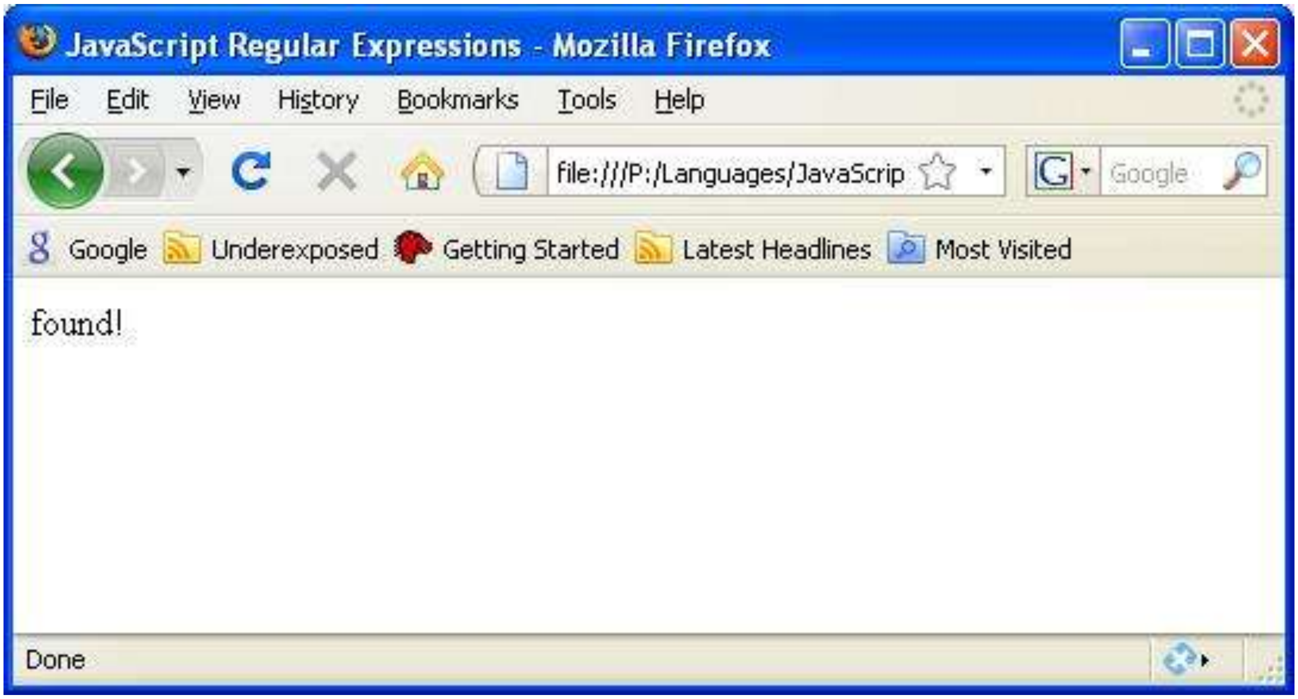
There are other JavaScript functions that use regular expressions other than the split function including:

- test – Will return true/false depending if a match
- occurs match – Returns a match if found
- search – Returns the index of the first match

replace – Replaces matches with a given string

The test function will return a true or a false.

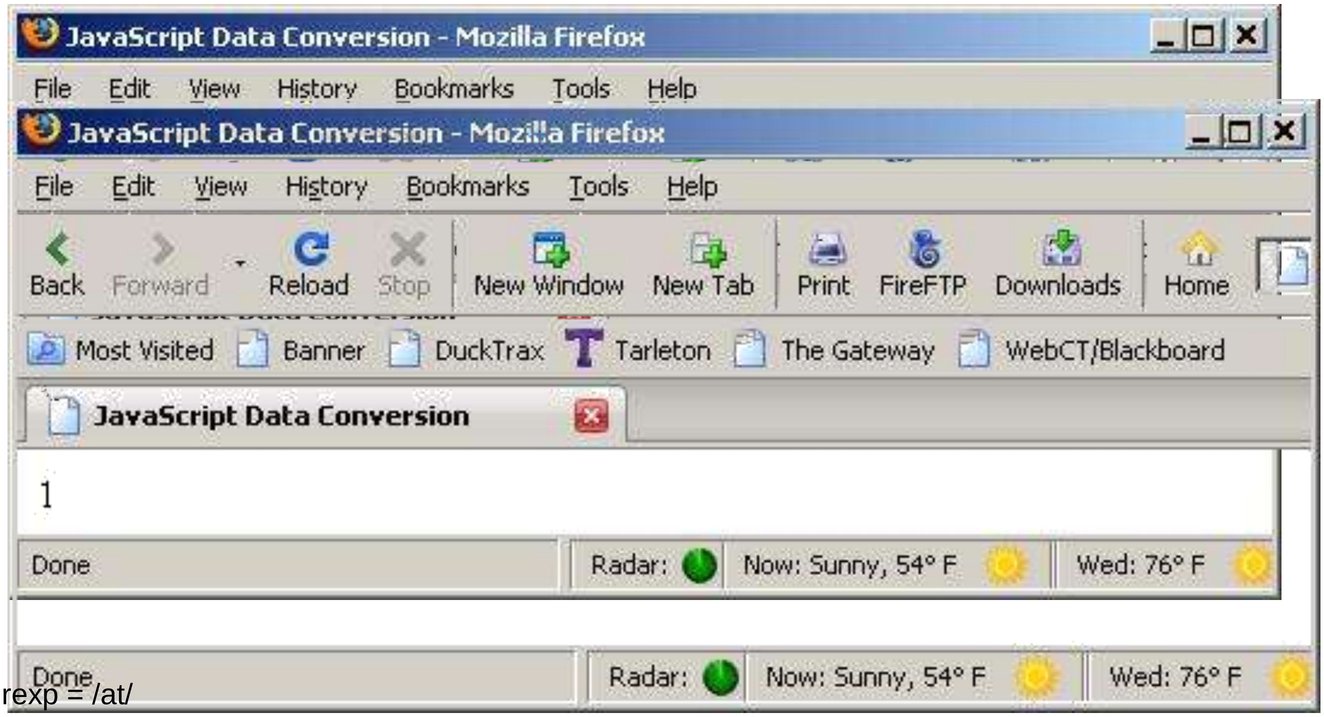
```
rexp =
/at/ if(rexp.test("catalog")) {
        document.write("found!<br>"); } else {

        document.write("not found!<br>");
}
```



```
found!
```

```
<html>

<head>

<title>JavaScript Data Conversion</title>

<script language="JavaScript 1.2">
```

```
rexp = /at/

document.write("catalog".match(rexp));

</script>

</head>

<body>

</body>

</html>
```


JavaScript Data Conversion - Mozilla Firefox
File Edit View History Bookmarks Tools Help
JavaScript Data Conversion - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Back Forward Reload Stop New Window New Tab Print FireFTP Downloads Home
Most Visited Banner DuckTrax Tarleton The Gateway WebCT/Blackboard
JavaScript Data Conversion
Done Radar: Now: Sunny, 54° F Wed: 76° F
Done Radar: Now: Sunny, 54° F Wed: 76° F

```
1
```

```
rexp = /at/

document.write("catalog".search(rexp));
```

4.13 Math Object

The JavaScript Math object provides several properties and methods that can be useful.

Property	Description
E	Euler's number (~ 2.718)
LN2	the natural logarithm of 2
LN10	the natural logarithm of 10
LOG2E	the base-2 logarithm of E
LOG10E	the base-10 logarithm of E
PI	PI
SQRT1_2	the square root of 1/2
SQRT2	the square root of 2

Method	Description
abs(x)	Returns the absolute value of x
acos(x)	Returns the arccosine of x (radians)
asin(x)	Returns the arcsine of x, in (radians)
atan(x)	Returns the arctangent of x as a value
atan2(y,x)	Returns the arctangent of the quotient of its arguments
ceil(x)	Returns x, rounded upwards to the nearest integer

cos(x)	Returns the cosine of x (radians)
exp(x)	Returns the value of Ex
floor(x)	Returns x, rounded downwards to the nearest integer
log(x)	Returns the natural logarithm (base E) of x
max(x,y,z,...,n)	Returns the number with the highest value
min(x,y,z,...,n)	Returns the number with the lowest value
pow(x,y)	Returns the value of x to the power of y
random()	Returns a random number between 0 and 1
round(x)	Rounds x to the nearest integer
sin(x)	Returns the sine of x (radians)
sqrt(x)	Returns the square root of x
tan(x)	Returns the tangent of an angle

For example, to compute the area of a circle use the function:

```
function        areaOfACircle(radius)          {
                return Math.PI*radius*radius;
}
```

4.14 JavaScript Objects

There exist a number of predefined objects associated with the web browser and the HTML document loaded. Each of these objects has certain properties associated with them.

Document	Input Password
Events	Input Radio
Elements	Input Reset
Anchor	Input Submit
Area	Input Text
Base	Link
Body	Meta
Button	Object
Form	Option
Frame/IFrame	Select

Frameset	Style
Image	Table
Input Button	TableCell
nput Checkbox	TableRow
Input File	Textarea
Input Hidden	

An object frequently consists of sub elements which are separated by periods.

```
document.myform.text1.value
```

Objects also can have methods which are distinguished from properties by the use of the open and close parentheses. Here the values associated with the first form are reset.

```
document.forms[0].reset();
```

4.15 Window

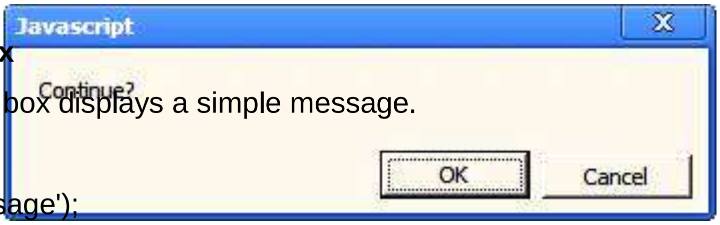
The window object can be used to create new windows and dialog boxes and includes these method:

- Open – Opens a new window
- Close – Closes the window
- alert – Displays an alert message box
- confirm – Displays a confirms dialog box
- prompt – Displays a prompt dialog box

It also possesses several properties including:

- document – Returns the Document object
- innerHeight – The height of the content area of the window
- innerWidth – The width of the content area of the window
- outerHeight – The height or the window including toolbars
- innerWidth – The width of the window

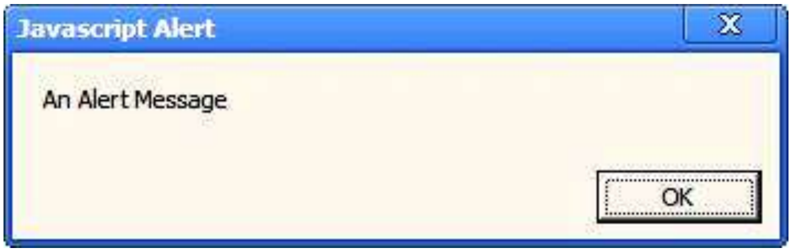
Alert Message Box



Continue?

The alert message box displays a simple message.

```
alert('An Alert Message');
```



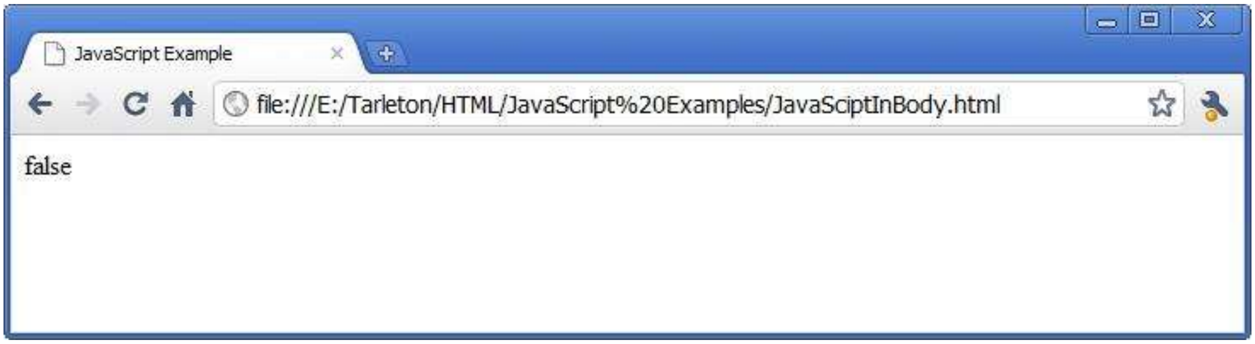
An Alert Message

Confirm Dialog Box

The confirm dialog box displays a confirm type message and then either returns a true or false value depending on which button is pressed.

```
var result = confirm("Continue?");
document.write(result);
```

If Cancel is selected, false is returned.



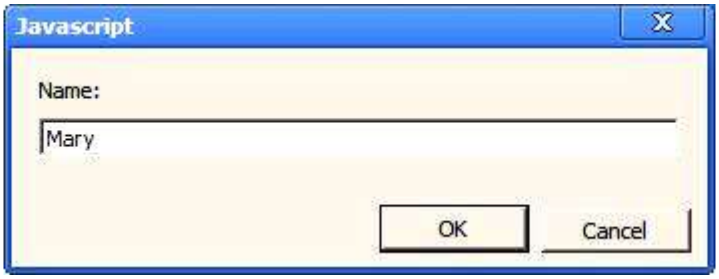
file:///E:/Tarleton/HTML/JavaScript%20Examples/JavaSciptInBody.html

```
false
```

Prompt Dialog Box

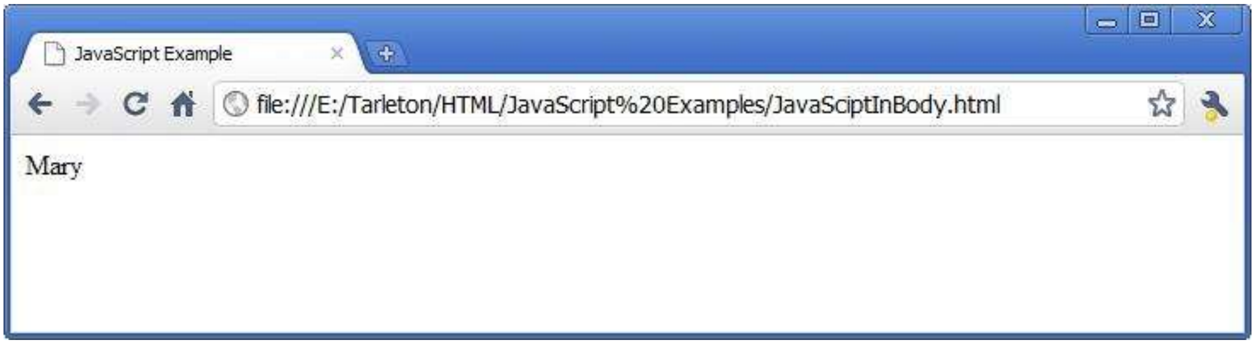
The prompt dialog box provides a way of getting input from the user. The prompt function has two arguments. The first is the prompt message and the second is a default value if any.

```
var result = prompt("Name:","");
document.write(result);
```



Name: Mary

The value returned is the value entered by the user.



file:///E:/Tarleton/HTML/JavaScript%20Examples/JavaSciptInBody.html

Mary

4.16 Document

The Document object provides access to all of the HTML elements of the current page. Useful properties include:

- cookie – Will return name/value pairs of the cookies used by the document
- domain – Returns the domain name of the server
- title – Returns or set the title
- URL – Returns the URL of the document.

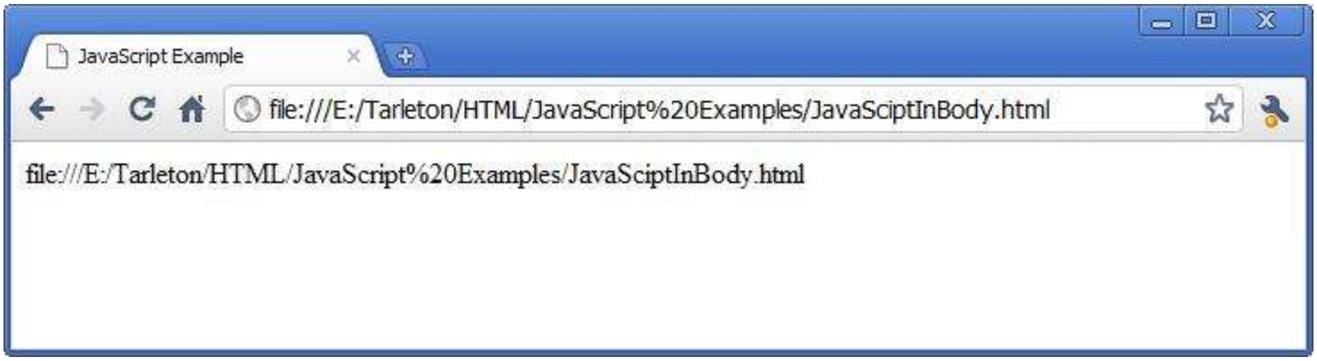
In addition, it consists of a series of array that hold the contents of the page. These objects can be accessed and modified. For example, the forms array contains a list of all of the forms that make up a page. Here the first form is selected. The value of the third element of the form is returned.

```
document.forms[0].elements[2].value
```

URL Property

The URL property is easy to use.

```
document.write(document.URL);
```



JavaScript Example
file:///E:/Tarleton/HTML/JavaScript%20Examples/JavaSciptInBody.html

file:///E:/Tarleton/HTML/JavaScript%20Examples/JavaSciptInBody.html

4.17 Frame

The Frame object refers to a frame of the web page. The Frames array is a list of the frames that make up a web page.

Properties of a frame include:

- frames – An array listing the frames that make up the page. Indexes start at 0
- length – The number of elements in the frames array
- self – Designates the current frame

- name – The name of the frame
- parent – The parent frame of the current frame

Methods of the frame object that are of interest include:

- blur – Removes the focus from the frame
- focus – Gives the frame focus
- setInterval -
- clearInterval
- setTimeout clearTimeout

4.18 JavaScript Events

Many elements of DOM support events. These events are normally the result of some user actions.

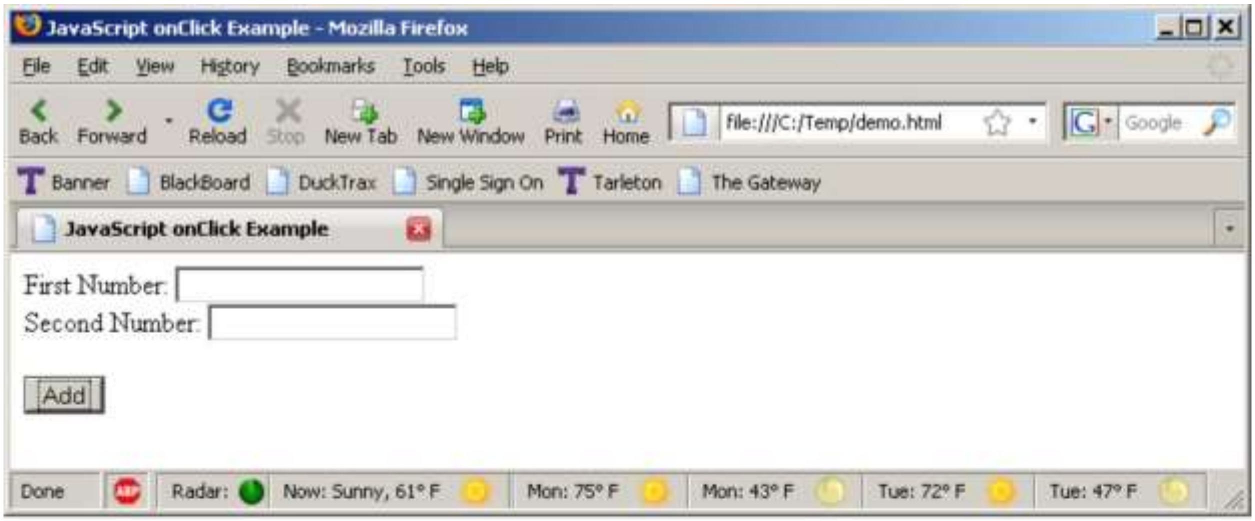
Event	Meaning

onload	Occurs when a window or frame has loaded
onunload	Occurs when a document is removed from a window or frame
onclick	The mouse is clicked on an element
ondblclick	The double click event
onmousedown	Mouse down event
onmouseup	Mouse up event
onmouseover	Mouse moves onto an element
onmousemove	Mouse moves over an element
onmouseout	Mouse leaves an element
onfocus	Element receives focus
onblur	Element loses focus
onkeypress	Key press event
onkeydown	Key is pressed down
onkeyup	Key is released
onsubmit	Submit button is pressed
onreset	Form reset event occurs
onselect	Some text in an element is selected
onchange	Element loses focus and its value changes

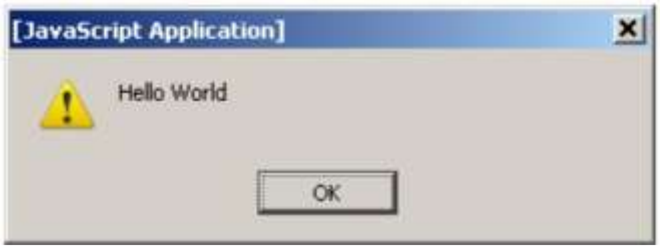
onClick Example

```
<html>
<head>
<title>JavaScript onClick Example</title> <script language="JavaScript">
<!--
function popup() { alert("Hello World")
}
//--> </script> </head> <body>

<form action="SampleServlet" method="POST">
```

```
    First Number: <input type="text" name="num1" size="20"><br> Second Number: <input type="text"
    name="num2" size="20"> <br><br>
    <input type="submit" onclick="popup()"value="Add"> </form>
</body>
</html>
```


JavaScript onClick Example - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Back Forward Reload Stop New Tab New Window Print Home
file:///C:/Temp/demo.html
Google
Banner BlackBoard DuckTrax Single Sign On Tarleton The Gateway
JavaScript onClick Example
Add
Done Radar: Now: Sunny, 61° F Mon: 75° F Mon: 43° F Tue: 72° F Tue: 47° F

First Number:
Second Number:



Hello World

4.19 Animation

JavaScript does not have a function such as Java‟s sleep method that pauses a task for a specified period of time. However, JavaScript has two functions that can be used to delay the execution of a function.

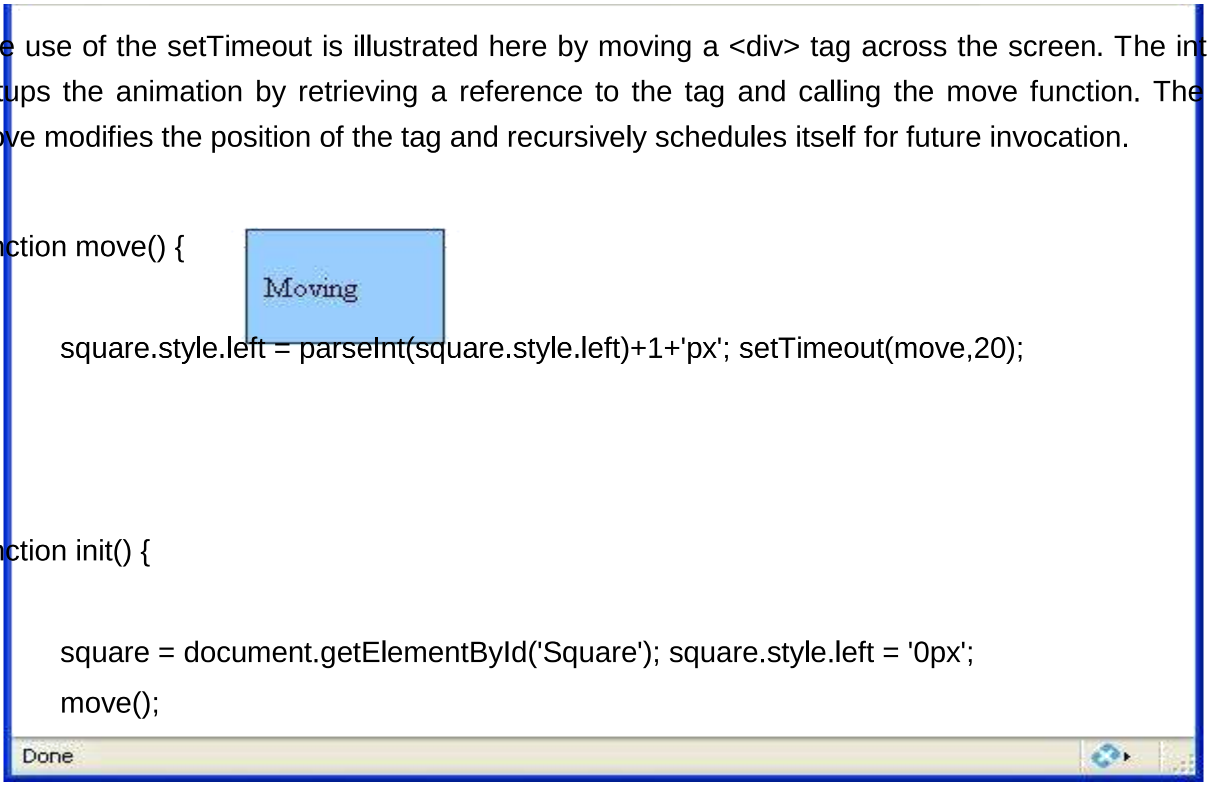
- **setTimeout** – Will execute a function a specific number of milliseconds in the future
- **setInterval** – Will execute a function every milliseconds

Both functions take on two arguments:

- Function – The first argument identifies the function to execute
- Time – The number of milliseconds

```
setTimeout(someFunction,500);       // The function will be executed 500 milliseconds
                                    // in the future
setInterval(someFunction,500);      // The function will be executed 500 every milliseconds
```

The use of the setTimeout is illustrated here by moving a <div> tag across the screen. The int function setups the animation by retrieving a reference to the tag and calling the move function. The function move modifies the position of the tag and recursively schedules itself for future invocation.



```
function move() {

        square.style.left = parseInt(square.style.left)+1+'px'; setTimeout(move,20);

}

function init() {

        square = document.getElementById('Square'); square.style.left = '0px';
        move();
}
```

The complete page follows:

```
<html>
<head>
<title>JavaScript  Animation</title>

<script language="JavaScript 1.2"> var square = null;

function move() {

        square.style.left = parseInt(square.style.left)+1+'px'; setTimeout(move,20);
}

function init() {

        square = document.getElementById('Square'); square.style.left = '0px';
        move();
}
window.onload = init;

</script>
</head>
```

```
<body>
<br>

<div id="Square" style="position:absolute; left:0px;

        top:8em;

        width:5em; line-height:3em; background:#99ccff; border:1px solid #003366; white-space:nowrap;
        padding:0.5em;"
>
Moving </div>

</body>
</html>
```

The same effect can be created using the setInterval function.

```
function move() {
        square.style.left = parseInt(square.style.left)+1+'px';
}
function init() {
        square = document.getElementById('Square'); square.style.left = '0px'; setInterval(move,20);
}
```

Lesson - 5

PHP BASICS

Structure

5.0 Objective

This chapter will cover following topics:

- Introduction to PHP
- Features of PHP
- Data types of PHP
- Operators in PHP
- Summary

5.1 Introduction to PHP:

PHP stands for *Hypertext Preprocessor. PHP is a computer programming language that* is widely-used for making interactive and dynamic web pages and websites. PHP is an open source programming language that can be used for web development. The code of PHP can be embedded into Hyper Text Markup Language (HTML). PHP is free alternative to Microsoft's Active Server Pages (ASP). PHP is written in C language.

5.2 Features of PHP:

The main features of PHP are following:

Open Source and free- The source code of PHP is available on the internet and we can download it free from the webpage "www.php.net". We can also make changes in the source code of PHP according to our need.

HTML-Embedded-PHP's code can be embedded into HTML by using </php and ?> tags.

Server Side Scripting Language-Programs of PHP are executed on the web server side. The result of PHP code is returned to the web browser as plain HTML. Therefore, we can say that PHP is a server side scripting language.

Dynamic and Interactive Web Pages- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages. PHP can generate dynamic page contents in a web site as compare to HTML which can develop only static websites. We can easily develop dynamic and interactive web pages using PHP.

Platform Independent- Programs of PHP can be run on different operating systems without making changes. The same program can run on Windows as well as on Linux operating system. Therefore, we can say that PHP is platform neutral language.

Simple and Efficient- PHP is very simple and straight forward language. It is C like language. Even a newcomer can learn PHP language easily. PHP is easy to learn and use. Program of PHP can be executed efficiently on the web server.

Case Sensitive- PHP is case sensitive as C language. There is a difference between uppercase and lowercase characters in PHP.

Default syntax of a PHP Program

A PHP script can be placed anywhere in the HTML document.

A PHP script starts with <?php and ends with ?>.

The general syntax of a PHP program is:

```
<?php

// you can write code of PHP here

?>
```

The default file extension for PHP files is ".php".

A PHP file normally contains HTML tags, and some PHP scripting code.

First Example

The code of PHP is enclosed in </php and ?> as shown in following first example. Echo function is used in this script to print the text on a web page.

```
<html>
   <head>
      <title>First Program of PHP</title>
   </head>
   <body>

      <?php
//First Program of PHP
         echo "Hello, I am the first program.";
      ?>

   </body>
</html>
```

PHP statements always ends with semi colon (;) as in C and C++. Save this program with .php extension. A PHP script is executed on the web server. The HTML result is sent back to the web browser. You can execute this program on php enabled web server.

Note: Echo function is used in this script to print the text on the web page.

The output will be

```
Hello, I am the first program.
```

Comments in PHP: A comment is a sequence of characters which are ignored by the compiler. As C++, We can use double slash // for single line comments in PHP.

PHP also support # for single line comment.

PHP use /* and */ for multiline comments.

The following program will illustrate the different comment styles of PHP.

```
<html>
   <head>
      <title>Program of PHP using comments</title>
   </head>
   <body>

      <?php
        //Program to illustrate the concept of comments
        // Program to print text with echo
        /* the author
           of first program
           is Mohan kumar*/
         echo "Hello, Good Morning";
      ?>
   </body>
</html>
```

The output will be

```
Hello, Good Morning
```

5.3 Data Types of PHP:

The basic data types of PHP are

- String
- Integer
- Float
- Boolean

5.3.1 String: String is a collection of characters. String data types can holds textual data. For example following are both string data types. Strings are delimited by single quotes and double quotes. Examples are given below:

```
$Firstname=”Mohan”
$Secondname=”Kumar”
```

$sen=" Mohan is a boy"

$a="5.5" is also a string. Anything enclosed by quotation marks is a string.

5.3.2 Integer: Integers are the numbers which have no decimal point. Integers are whole number. For example following are the integer data types

$a=5;

$b=12;

$c=112347; are integers.

5.3.3 Float:

Float data type referred to the numbers that contain fractional parts. For example

$a=5.5;

$b=10.37;

$c=105.456; are the float variables.

5.3.4 Boolean: A Boolean variable represent two values; true or false. You can also use 0 to represent false value and any nonzero number for true value. Examples are below:

$flag = false; // flag is false

$flag = true; // flag is true

$flag = 0;   // flag is false

$flag = 5 ; // flag is true

5.4 PHP Operators:

An Operator is used to perform mathematical or logical manipulations on data and variables. PHP operators can be classified into a number of categories as below:

- Arithmetic operators
- Assignment operators
- Relational operators
- Increment and Decrement operators
- Logical operators
- String operators
- Array operators

5.4.1 Arithmetic operators

Arithmetic operators are used to make mathematical expressions in PHP. They are shown below:

- Addition +
- Subtraction -
- Multiplication *

- Division /
- Remainder after division %

Arithmetic operators can be used as below:

- *$a + $ b*
- *$a - $b*
- *$a * $b*
- *$a / $b*
- *$a % $b*

Where a and b are called operands.

Program to add two numbers is as given below:

```
<html>
<body>
<h1>My second PHP script</h1>

<?php
//Program to illustrate operators

$a = 5;  // First Number
$b = 10; //Second Number
echo $a + $b; //Sum of two numbers will be printed
?>
</body>
</html>
```

The output will be

```
 15
```

Program to multiply two numbers is as given below:

```
<html>
<body>
<h1>My second PHP script</h1>
```

```
<?php
//Program to illustrate operators

$a = 20;  // First Number
$b = 40; //Second Number
$c = $a * $b;//Multiplication
echo $c; //Product of  two numbers will be printed
?>
</body>
</html>
```

The output will be

```
800
```

5.4.2 Assignment operators

- Assignment operator is used to assign a value to a variable. In PHP, the basic assignment operator is "=". It means that the left operand gets set to the value of the assignment expression on the right.

```
$a=10;
```

Here a is assigned value 10.

PHP has a set of shorthand assignment operators.

The syntax is

var op=exp;

Where var is variable, op is PHP operator and exp is an expression.

For example:

x += y is equivalent to x=x+y

x -= y is equivalent to x=x-y

x *= y is equivalent to x=x*y

x /= y is equivalent to x=x/y

5.4.3 Relational Operators

Sometimes, we have to compare two or more numbers and depending on the result, we have to take decisions.

Such comparisons can be done using relational operators.

The relational operators available in PHP are:

Equal	==
Identical	===
Not equal	<>
Not identical	!==
Less than	<
Greater than	>
Greater than equal to	>=
Less than equal to	<=

Examples of relational operators

- $a==$b It returns true if $a is equal to $b
- $a===$b It returns true if $a is equal to $b and are of same type
- $a<>$b It returns true if $a is not equal to $b
- $a!==$b It returns true if $a is not equal to $b or are not of same type
- $a>$b It returns true if $a is greater than $b
- $a<$b It returns true if $a is less than $b
- $a>=$b It returns true if $a is greater than or equal to $b
- $a<=$b It returns true if $a is less than or equal to $b

5.4.4 Logical Operators

Logical operators are used to combine two or more conditional statements. The logical operators available in PHP are:

- AND
- OR
- NOT
- XOR

Examples:

$a and $b It returns true if both a and b are true

$a or $b It returns true if either a or b is true

!$a It returns true if a is false and vice versa

$a xor $b It returns true if either a or b is true but not both

5.4.5 Increment and Decrement operators

Increment and decrement operators are the unary operators. Increment operator is used to increase the value of operand by 1, whereas decrement operators is used to decrease the value of operand by 1.

In PHP, we can use C like pre and post increment and decrement operators.

- Pre-increment ++$a It increase the value of a by one and then returns the value of a
- Post-increment $a++ It returns the value of a and then increase the value of a by one
- Pre-decrement --$a It decrease the value of a by one and then returns the value of a
- Post-decrement $a-- It returns the value of a and then decrease the value of a by one

5.5 Summary:

PHP is a programming language that is widely-used for making interactive and dynamic web pages. PHP is open source programming language that can be used for web development. The code of PHP can be embedded into Hyper Text Markup Language (HTML). A PHP script can be placed anywhere in the HTML document. The basic data types of PHP are: String, Integer, Float and Boolean. String is a collection of characters. String data types can holds textual data. Integers are the numbers which have no decimal point. A Boolean variable represent two values; true or false. You can also use 0 to represent false value and any nonzero number for true value. Float data type referred to the numbers that contain fractional parts. Arithmetic operators are used to make mathematical expressions in PHP. Assignment operator is used to assign a value to a variable. In PHP, the basic assignment operator is "=". It means that the left operand gets set to the value of the assignment expression on the right. Sometimes, we have to compare two numbers and depending on the result, we have to take decisions. Such comparisons can be done using relational operators. Logical operators are used to combine two or more conditional statements. The logical operators available in PHP are: AND, OR, NOT and XOR. Increment and decrement operators are the unary operators. Increment operator is used to increase the value of operand by 1, whereas decrement operators is used to decrease the value of operand by 1. In PHP, we can use C like pre and post increment and decrement operators.

5.6 Objective types question/answers to check your progress

Q1: PHP is open source or not.

Ans: yes, open source

Q2: PHP stands for.

Ans: Hypertext Preprocessor

Q3: PHP is used for.

Ans: Web development

Q4: Write the extension used to save the PHP programs.

Ans: .php

Q5: what are the data types of PHP?

Ans: integer , float, Boolean , string

Q6: Is PHP , platform independent language?

Ans: yes

Q7: What is the syntax for shorthand assignment operator?

Ans: var op= exp;

Where var is variable, op is PHP operator and exp is an expression.

For example:

x += y is equivalent to x=x+y

5.7 Book References:

- Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff
- Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress

5.7.1 Web References:

- http://php.net/manual
- https://en.wikipedia.org/wiki/Operator_(computer_programming)
- https://www.w3schools.com/php
- https://www.tutorialspoint.com/php
- http://www.w3programmers.com

5.8 Suggested Readings:

- PHP: The Complete Reference by Steven Holzner Publisher: Tata McGraw Hill
- Programming PHP by Kelvin Tetroi
- PHP 6 and MYSQL Bible by Steve Suehring, Wiley India edition
- Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff
- Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress

5.9 Model questions

Q1: What is PHP? Describe features of PHP.

Q2: What is operator? Explain different types of operators in PHP with examples.

Q3: Explain various data types available in PHP.

Q4: Write a Program in PHP for addition, subtraction, multiplication and division of two integers.

Q5: Write a Program in PHP to find greatest of three numbers.

Q6: Explain different logical operators available in PHP.

Q7: What is relational operator? Explain different types of it using suitable examples.

Q8: How can you use in PHP to make a small calculator.

Q9: How can you use increment and decrement operators in PHP?

Q10: What is assignment operator? Explain various short hand assignment operators available in PHP.

****

Lesson - 6

CONTROL STRUCTURES

Structure

6.0 Objectives:

This chapter will cover following topics:

- Decision Making in PHP
- If statement
- If-else statement
- Nested if-else
- if else-if ladder
- switch statement
- Loops in PHP
- While Loop
- Do-while Loop
- For loop
- For-each loop
- Summary

6.1 Decision Making in PHP

A PHP program is generally a set of statements, which are executed in the order in which they appear. There are a number of situations where we have to change the order of execution of statements based on certain decisions. In PHP, when a program breaks the linear flow and jump to another part of the program, it is known as branching. Conditional branching is the branching which is based on a condition. If branching takes place without any decision, it is called unconditional branching.

PHP supports following decision making statements:

6.1.1. if statement

6.1.2 . If-else statement

6.1.3. Nested if-else

6.1.4. if elseif ladder

6.1.5. switch statement

if STATEMENT

The general form of a simple if statement is:

```
If (condition)
{
        Statement block;
}
```

The condition is tested. If the condition is true, the statement block will be executed else control will go to next statement after the if block. Where, statement block can be a single statement or a group of statements.

Example:

```
<?php
//Program to illustrate if statement


$a = 5;
if ($a < 10)
 {
   echo "Good Morning";
}
?>
```

The output of this program will be

Good Morning

Note: value of a is less than 10.

THE if..else STATEMENT

The general form of a simple if-else statement is:

```
        If (condition)
        {
                Statement block-1;
        }
        Else
        {
                Statement block-2;
        }
```

The condition is tested. If the condition is true, the statement block-1 will be executed and if condition is false then statement block-2 will be executed.

Example:

```
<?php
//Program to illustrate if-else statement
$a = 5;
```

```
$b = 10;

if ($a > $b)
{
   echo "a is greater than b";
}
else
{
   echo "b is greater than a";
}
?>
```

Output:

The output of the program is

```
b is greater than a
```

Nested if STATEMENT

When a if statement is nested in another if , it is called nested if statement. The general form of a Nested if-else statement is:

```
If( condition1)
        {
                If( condition2)
                {
                Statement  block-1;
                }
                Else
                {
                statement block-2;
                }
        }
        Else
        {
                Statement  block-3;
        }
```

Firstly, condition 1 is evaluated. If the condition 1 is false, the statement block- 3 will be executed; otherwise condition 2 is evaluated. If the condition 2 is true, the statement block-1 will be executed; otherwise the statement block 2 will be executed and then the control is transferred to next statement.

Example:

```
<?php
//Program to illustrate nested if statement
$a = 5;
$b = 10;
$c = 7;
if ($a > $b)
{
        if ($b > $c)
        {
        echo "a is greater than b and c";
        }
        else
        {
        echo "b is smaller than c and greater than a";
        }
Else
{
       Echo “ b is less than a”;
}
?>
```

The output of this program is

```
 b less than a.
```

---

if else-if ladder

if else-if ladder in PHP is used to execute code for more than two conditions. The general form of if else-if ladder is:

```
if (condition-1)
{
   Statement block-1
}
elseif (condition-2)
```

```
{
   statement block-2;
}
 else
 {
    Statement block-3
}
```

Firstly, condition 1 is executed. If condition 1 is true, than statement block-1 is executed. If condition 1 is false then condition 2 is evaluated. If condition 2 is true then statement block-2 is executed and if it is false statement block-3 is executed.

Example:

```
<?php

//Program to illustrate if else if ladder

$a = 5;

$b = 10;

$c = 7;


if ($a > $b)

{

Echo “a is greater than b;

elseif ($b > $c)

          {
          echo "a is less than b and b is greater than c";
          }

          else

          {
          echo "a is less than b and and b is less than c";
          }

?>
```

The output of this program will be

“a is less than b and b is greater than c.”

Switch Statement: When there are a lot of conditions, then if-else become complex. We can use switch statement that allow us to make a decision from the number of choices. The conditional statement that allow us to make a decision from the number of choices is known as switch. Switch statement can make the program simple.

The general syntax is:

```
switch ( expression)
{
        case constant  1:
           execute this;
           break;
        case constant 2:
           execute  this;
            break;
        case constant 3:
           execute this;
              break;
        ………..
          ………
        default:
           execute this;
}
```

Firstly, expression is evaluated. Then value of the expression is compared with the values for each case in the switch. If there is a match, the statements associated with that case is executed. Break is used to prevent the code from running into the next case automatically.

Note: The default case is executed if no match is found.

Example:

```
<?php
//Program to illustrate switch statement

$day = 4";

switch ($day)
{
//switch block


   case 1:
      echo "Day is Sunday";
      break;
```

```
  case 2:
      echo "Day is Monday";
      break;

  case 3:
      echo "Day is Tuesday";
      break;

case 4:
      echo "Day is Wednesday";
      break;

case 5:
      echo "Day is Thursday";
      break;

case 6:
      echo "Day is Friday";
      break;

case 7:
      echo "Day is Saturday";
      break;
   default:
      echo "Wrong Day";
}
?>
```

Output:

The output of this program is

Day is Wednesday.

Note: case 4 will be executed.

6.2 Loops

In the last previous section, we have discussed the conditional statements in PHP. Conditional statements allow developers to make decisions in PHP programs. Whereas, Loops are the control statements in which a block of code can be executed may times. The contents of the loop are executed again and again depending upon the condition associated with the loop. A loop is a collection of statements that are repeated until a condition is reached.

Note: Each passage of loop is called iteration.

Types of Loops

PHP supports following type of loops:

6.2.1. While Loop
6.2.2. Do-while Loop
6.2.3. For loop
6.2.4. For-each loop

While Loop

The While loop is simplest of the three loops. The while loop executes a block of statements as long as the specified condition is true. As, the condition becomes false, control will come out of the while loop.

The general syntax of while loop is:

```
while ( condition)
{
        Statement- block
}
```

Example:

```
<?php
//Program to illustrate while loop
$i=1;
while ($i<=5)
{
        Echo $i;
           $i=$i+1;
}
```

The output of this program is

```
1
2
3
4
5
```

Note : Number of iterations = 5

do while Loop

In do while loop condition is executed at the end of the loop, not at the starting of the loop as in while loop. The contents of do while loop are executed at least once, even if the condition is false.

```
do
{
        Statement block
}
while ( condition);
```

Example:

```
<?php
//Program to illustrate do while loop
$i=1;
Do {
        Echo $i;
           $i=$i+1;
}
while ($i<=5);
?>
```

The output of this program is

```
1
2
3
4
5
```

Note : Number of iterations = 5

For Loop

The for loop is used when we have to repeat the code a specified number of iterations. The syntax of for loop is:

```
for (initialization; condition; increment or decrement)
```

```
{
   Statement-block;
}
```

Parameters:

- *Initialization:*: Initialize the value of loop counter
- *Condition*: Evaluated for each iteration. If it is TRUE, the loop continues. If it is FALSE, the loop will be terminated
- *Increment/decrement*: Increase or decrease the value of loop counter

Example:

```
<?php

//Program to illustrate for loop

$day = array("sunday", "monday", "tuesday", "wednesday","thrusday",”Friday”, “Saturday”);
for ($i=1; $i<=5;$i++)

 {

        echo $i;

  }

?>
```

The output of this program is

```
1

2

3

4

5
```

Note : Number of iterations = 5

The for loop uses the concept of a loop counter. The Loop counter is used to count the number of iterations. In the previous example varriable I is used as the loop counter.

Foreach Loop – Foreach loop is used to execute a block of code for each element of the array. This loop is used only on arrays. It is an extension to for loop. The syntax of foreach loop is

Syntax

```
foreach ($arrayname as $arrayitem)

 {
   Statement block;
}
```

This means for each item in the array, the statement block is executed.

Example

```
<?php
//Program to illustrate foreach loop

$day = array("sunday", "monday", "tuesday", "wednesday","thrusday",”Friday”, “Saturday”);

foreach ($day as $value)
 {
   echo "$value <br>";
}
?>
```

Output:

This program will display all the elements of the array.

Note : Number of iterations = 7

6.3 Summary

When a program breaks the linear flow and jump to another part of the program, it is known as branching. Conditional branching is the branching which is based on a condition. If branching takes place without any decision, it is called unconditional branching. We can use simple if, if-else, nested if, if else-if ladder, nested if and switch constructs for decision making. Loops are the control statements in which a code can be executed may times. The contents of the loop are executed again and again depending upon the condition. Each passage of loop is called iteration. PHP supports following type of loops:

- While Loop
- Do-while Loop
- For loop
- For-each loop

The While loop is simplest of the three loops. The while loop executes a block of statements as long as the specified condition is true. As, the condition becomes false, control will come out of the while loop. In do while loop condition is executed at the end of the loop, not at the starting of the loop as in while loop. The contents of do while loop are executed at least once, even if the condition is false. The for loop is used when we have to repeat the code a specified number of iterations. Foreach loop is used to execute a block of code for each element of the array. This loop is used only on arrays.

6.4 Objective types question/answers to check your progress

Q1: Is switch statement in PHP is decision making statement ?

Ans: yes

Q2: In PHP, is it possible to write if statement in another if statement?

Ans: yes , nested if

Q3: In switch statement when no match is found then which case will be execute?

Ans: default case

Q4: How conditional statement in PHP is differ from conditional statement in c?

Ans: “foreach” loop is used in PHP

Q5: Which loop is called entry and exit controlled loop in PHP?

Ans: entry = while loop

Exit=do while loop

Q6: Which loop is used only for array?

Ans: foreach loop

6.5 References:

6.5.1 Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff

6.5.2 Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress

6.5.3 http://php.net/manual

6.5.4 https://www.w3schools.com/php

6.5.5 https://www.tutorialspoint.com/php

6.5.6 http://www.w3programmers.com

6.5.7 http://whatis.techtarget.com/definition/loop

6.6 Suggested Readings:

6.6.1 PHP: The Complete Reference by Steven Holzner Publisher: Tata McGraw Hill

6.6.2 Programming PHP by Kelvin Tetroi

6.6.3 PHP 6 and MYSQL Bible by Steve Suehring, Wiley India edition

6.6.4 Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff

6.6.5 Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress

6.7 Questions

Q1: What is Decision Making? Describe various decision making constructs of PHP.

Q2: Write a program using switch statement in PHP to display the name of month of the year

if number of month is given.

Q3: Draw the difference between while and do-while loop.

Q4: What is loop? Explain different types of loops available in PHP.

Q5: Explain the purpose and use of foreach loop in PHP.

Q6: Write a program in PHP to find factorial of a number.

Q7: Write a program in PHP to generate Fibonacci series.

Q8: Write a Program in PHP to generate Pascal triangle.

Q9: Write a program in PHP to find a number is prime or not.

Q10: Write a program in PHP to find a number is odd or even.

Q11: What is iteration in a loop? Explain with an example.

****

Lesson - 7

FUNCTIONS IN PHP

Structure

7.0 Objectives:

This chapter will cover following topics:

- Introduction to Functions
- Benefits of Functions
- Library Functions
- User defined Functions
- Return statement
- Call by Reference
- Call by Value
- Default Arguments
- Recursion
- Summary

7.1 Function

A function may be defined as a self-contained block of statements that perform a specific task of some kind. Using a function is like hiring a person to do a specific work for you. A function can take values as input and then perform some operations and then return another value.

7.2 Benefits of Functions

7.2.1 By using functions, we can avoid rewriting the same code over and over.
7.2.2 Writing functions makes the easier to write program.
7.2.3 Functions makes the programs easy to understdand.

7.3 Library Functions

Library Functions are the functions whose code is already available. More than one thousands functions are built into standard PHP distribution. We can call any library function by simply specifying the function name. Pow() function is used in the following example:

```
<?php
//Program to illustrate concept of library function

$ans = pow(2,4); //returns 16
   echo "$ans <br>";
}
?>
```

Output:

In this program we have used the pow() library function of PHP.

This program will print the 16 as output.

7.4 User defined functions

Although, a lot of library functions are available in PHP. Sometimes, we have to go beyond what is available. In PHP, we can also build our own functions.

Creating a user defined function

The keyword function is used to create a function.

The syntax of creating a function is:

```
Function function name(parameters)
{
Body of function
}
```

Example:

```
Function message()
{
Echo “Good Morning, How are you”
}
```

You can call this function in your PHP code as

```
<?php
//Program to illustrate concept of user defined function

Message();  //function call
?>
```

The output of this program is:

```
Good Morning, How are you
```

7.5 Return Statement:

The return() statement can be used in PHP to return the value back to the calling function. The following example will illustrate the concept of return statement.

The syntax of return statement is:

```
Return variable;
```

Where variable is the variable name whose value is to be returned to the calling function

Example:

```
Function totalmarks ($m1,$m2,$m3)
{
$sum=$m1 + $m2 +$m3;
Return sum;
}
<?php
//Program to illustrate return statement
summarks = totalmarks(40,50,60);  //function call
echo summarks;
?>
```

The output of this program will be

```
150
```

7.6 Call by value:

In this method values of variables are passed as arguments. The value of the actual arguments is copied into corresponding formal arguments. The changes made to the formal arguments in the called function does not effect on the values of actual arguments. The example of call by value is shown below:

```
Function calculatetax ($pay)
{
//Program to illustrate pass by value
$tax= $pay*10/100;
$pay=$pay-$tax;
return ($pay)
}
<?php
$Pay=10000;
Echo (calculatetax($pay));  //print 9000
Echo $pay; //print 10000
?>
```

The output of this program will be

```
9000
10000
```

Note: The value of $pay is not changed.

7.7 Call by reference:

In this method address of variables are passed as arguments. The address of the actual arguments is copied into corresponding formal arguments. This means that in this case function can access the actual arguments and can make changes in it. The changes made to the formal arguments in the called function makes effect on the values of actual arguments. You have to add ampersand in front of variables you are passing as arguments. The example of call by value is shown below:

```
Function calculatetax (&$pay)
{
//Program to illustrate pass by reference
$tax= $pay*10/100;
$pay=$pay-$tax;
return ($pay);
}
<?php
$Pay=10000;
Echo (calculatetax($pay));  //print 9000
Echo $pay; //print 9000
?>
```

The output of this program will be

```
9000
9000
```

Note: The value of $pay is changed.

7.8 Default Arguments:

Default arguments are the arguments whose value is assigned automatically to a default value. if no value is passed then this default value is assigned to the argument.

Example:

```
Function calculatetax ($pay,$rate=20)
{
//Program to illustrate default arguments
$tax= $pay*$rate/100;
$pay=$pay-$tax;
return ($pay);
}

<?php
$Pay=10000;
Echo (calculatetax($pay));  //print 8000
Echo (calculatetax($pay,30));  //print 7000
?>
```

The output of this program will be

```
8000
7000
```

In this example, $rate is default argument and have value 20. When value of rate is not passed as argument, then 20 is automatically taken as its default value.

7.9 Recursion:

Recursion is a method with help of which programmers can express operations in terms of themselves. When a function calls itself, it is called recursion. Recursion is very similar to loop as it also repeat the same code.

The syntax of recursive function is:

```
fun()
{
fun();
}
```

We can calculate factorial of a number using recursion as given in this example:.

```
function fact($n)
{
//Program to illustrate recursion
//Recursive function

    if ($ < 2)
     {
      return 1;
   }
 Else
 {
      return ($n * fact($n-1));
 }
}
```

```
<?php
$n=5;
Echo (fact($n)); //Function Call
 //print 120
?>
```

The output of this program will be

120

Note: Factorial of 5 is 120.

7.10 Summary

A function may be defined as a self contained block of statements that perform a specific task of some kind. Using a function is like hiring a person to do a specific work for you. A function can take values as input and then perform some operations and then return another value. Library Functions are the functions whose code is already available. More than one thousands functions are built into standard PHP distribution. We can call any library function by simply specifying the function name. Although, a lot of library functions are available in PHP. Sometimes, we have to go beyond what is available. In PHP, we can also build our own functions. The keyword function is used to create a function. The syntax of creating a function is:

```
Function function-name (parameters)
{
Body of function
}
```

In call by values method, values of variables are passed as arguments. The value of the actual arguments is copied into corresponding formal arguments. The changes made to the formal arguments in the called function does not effect on the values of actual arguments. In call by reference method, address of variables are passed as arguments. The address of the actual arguments is copied into corresponding formal arguments. This means that in this case function can access the actual arguments and can make changes in it. The changes made to the formal arguments in the called function makes effect on the values of actual arguments. You have to add ampersand in front of variables you are passing as arguments. The return() statement can be used in PHP to return the value back to the calling function. Default arguments are the arguments whose value is assigned automatically to a default value. if no value is passed then this default value is assigned to the argument. Recursion is a method with help of which programmers can express operations in terms of themselves. When a function calls itself, it is called recursion. Recursion is very similar to loop as it also repeat the same code.

7.11 Objective types question/answers to check your progress

Q1: Built-in functions in PHP are called as-

Ans: Library Functions

Q2: What is the output of following library function Pow(2,4) ?

Ans: 16

Q3: write down the syntax for function definition.

Ans: Function function_name()

```
{
//body of function
}
```

Q4: which statement in PHP returns the value back to the calling function?

Ans: return()

Q5: Name two methods used to pass argument to a function.

Ans: call by value and call by reference.

Q6: Does the changes made in a formal arguments will affect the actual arguments in call by value?

Ans: no

7.12 References:

1. Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff
2. Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress
3. http://php.net/manual
4. https://www.w3schools.com/php
5. https://www.tutorialspoint.com/php
6. http://www.w3programmers.com
7. https://en.wikipedia.org/wiki/Recursion_(computer_science)

7.13 Suggested Readings:

1. PHP: The Complete Reference by Steven Holzner Publisher: Tata McGraw Hill
2. Programming PHP by Kelvin Tetroi
3. PHP 6 and MYSQL Bible by Steve Suehring, Wiley India edition
4. Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff
5. Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress

7.14 Model Questions

Q1: Define function. Discuss various types of functions in PHP with suitable examples.

Q2: Draw the difference between call by value and call by reference with examples in PHP.

Q3: Write a program in PHP to find factorial of a number using recursion.

Q4: Explain the concept of default arguments with help of an example in PHP.

Q5: What is Recursion? Write a program in PHP to generate Fibonacci series using recursion.

****

STRINGS IN PHP

Structure

8.0 Objective:

This chapter will cover following topics:

- Introduction to Strings
- strlen() function
- strrev() functiom
- str_word_count() function
- stropos() function
- str_replace() function

8.1 Intorduction to Strings

A string is a collection of characters, like "Mohan Kumar", “My age is twenty five”. String is a sequence of characters. String data types can holds textual data. We can also say that string is an array of characters.

Examples are given below:

$Firstname=”Mohan”

$Secondname=”Kumar”

$class=”PGDCA”

$sen=” Mohan is a boy”

$a=”5.5” is also a string.

Anything enclosed by quotation marks is a string.

Strings are delimited by single quotes and double quotes in PHP.

We can say that string is an array of characters.

8.2 PHP String Functions

We will discuss following string related library functions in this section:

- strlen() function
- strrev() functiom
- str_word_count() function
- stropos() function
- str_replace() function

8.2.1 Strlen() function

Strlen() function is used to find length of a string in PHP.

The syntax of strlen() fiunction is

Strlen(string)

This function will return the length of the string.

The following example will print the length of the string "Mohan Kumar":

Example

```
<?php
//Program to illustrate the concept of strlen() function

$n = strlen("Mohan Kumar");
echo $n;
 // Print 11
?>
```

Output:

This program will print

```
 11
```

Note: 11 is the length of given string. Space is also a character.

Another Example

```
<?php
//Program to illustrate the concept of strlen function

echo strlen("My age is twenty five");
 // Print 21
?>
```

Output:

This program will print

```
 21
```

Note: 21 is the length of given string. Space is also counted as a character.

8.2.2 str_word_count() function

str_word_count() function is used to find number of words in PHP.

The syntax is:

Str_word_count(string);

This function will return the number of words in the given string.

The following example will print the length of the string " My age is twenty five":

Example

```
<?php
// Program to illustrate str_word_count function


$n = str_word_count("My age is twenty five");
echo $n;
 // prints 5

?>
```

Output:

This program will print

5

Note: 5 is the number of words in the string “My age is twenty five”

Another Example

```
<?php
// Program to illustrate str_word_count function
echo str_word_count("Mohan Kumar");
 // prints 2

?>
```

Output:

This program will print

2

Note: 2 is the number of words in the string “Mohan Kumar”

8.2.3   strrev() function

strrev() function is used to reverse a string in PHP:

The syntax is:

Strrev(string);

The string will be reversed.

Example

```
<?php
// Program to explain strrev function
echo strrev(“mohankumar");
 // prints ramuknahom

?>
```

Output:

This program will print

Ramuknahom

Note: Reverse of mohankumar.

8.2.4 strpos() function

strpos() function is used to search a text in a string. If text is found in the string, the function returns the position of first match in string. If no match is found, then it returns false.

The syntax is:

Strops(string 1, string 2);

The following example will search the word “is” from the string " My age is twenty five":

Example

```
<?php
// Program to illustrate strops function

echo strpos("My age is twenty five ", "is");
 // prints 3
?>
```

The output of the code above will be:

 3

Note: The position of first character in a string is 0, not 1.

8.2.5 str_replace() function

The str_replace() function finds and replaces some text with some other text in a string.

The syntax is:

Str_replace(string 1, string 2);

Where string 1 will be replaced by string 2.

The following example will replace the text "twenty" with "thirty":

Example

```
<?php
//Program to illustrate str_replace function

echo str_replace ("twenty", “thirty”, "My age is twenty five ");
// prints My age is thirty five
?>
```

The output of the code above will be:

```
My age is thirty five
```

Note: twenty is replaced by thirty

8.3 Summary

A string is a collection of characters, like "Mohan Kumar", “My age is twenty five”. String is a sequence of characters. String data types can holds textual data. *Anything enclosed by quotation marks is a string.* Strings are delimited by single quotes and double quotes in PHP. We can say that string is an array of characters. Strlen() function is used to find length of a string in PHP.

The syntax of strlen() fiunction is

Strlen(string)

str_word_count() function is used to find number of words in PHP.

The syntax is:

Str_word_count(string);

strrev() function is used to reverse a string in PHP:

The syntax is:

Strrev(string);

strpos() function is used to search a text in a string. If text is found in the string, the function returns the position of first match in string. If no match is found, then it returns false.

The syntax is:

Strops(string 1, string 2);

The str_replace() function finds and replaces some text with some other text in a string.

The syntax is:

Str_replace(string 1, string 2); Where string 1 will be replaced by string 2.

8.4 Objective types question/answers to check your progress

Q1: Which function is used to find the length of the string?

Ans: strlen()

Q2: str_word_count() is used for:

Ans: to count number of words in PHP

Q3: Is it possible to reverse a string and name that function?

Ans: Yes, strrev()

Q4: What is the position of the first character in string 0 or 1?

Ans: 0

Q5: str_replace(string1,string2), what this function does ?

Ans: string1 is replaced by string2.

Q6: Does the space is also count as a character in string?

Ans: yes.

8.5 References:

8.5.1 Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff

8.5.2 Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress

8.5.3 http://php.net/manual

8.5.4 https://www.w3schools.com/php

8.5.5 https://www.tutorialspoint.com/php

8.5.6 http://www.w3programmers.com

8.5.7 https://en.wikipedia.org/wiki/String_(computer_science)

8.6 Suggested Readings:

8.6.1 PHP: The Complete Reference by Steven Holzner Publisher: Tata McGraw Hill

8.6.2 Programming PHP by Kelvin Tetroi

8.6.3 PHP 6 and MYSQL Bible by Steve Suehring, Wiley India edition

8.6.4 Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff

8.6.5 Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress

8.7 Model Questions

Q1: What is String? How can you represent strings in memory?

Q2: What is a string function? Explain various string functions available in PHP.

Q3: Write a program in PHP to reverse a string.

Q4: Write a program in PHP to find the string is a palindrome or not.

Q5: Write a program in PHP to find length of a string.

Q6: Write a program in PHP to illustrate the use of stringpos() function.

Q7: Write a program in PHP to find the number of words in a string.

Q8: Write a program in PHP to find the number of sentences in a paragraph.

Q9: Write a program in PHP to find number of vowels in a string.

Q10: Write a program in PHP to illustrate the use of st_word_count() function.

Lesson - 9

ARRAYS

Structure

9.0 Objective:

This chapter will cover following topics:

- Introduction to Arrays
- Index Arrays
- Associative arrays
- Count() Function
- Range() Function
- Forms in PHP
- Difference between GET and POST

9.1 Introduction to Arrays

Array is a collection of homogeneous type of data. An Array can store more than one value at a time. An Array is a group of data items that share a common name. For example, we can define an Array name marks to represent a set of marks of a group of students. Each member of the array is known as element. Elements of the array share certain common characteristics. In PHP arrays can be of two types

- Index Arrays
- Associative Arrays

9.2 Index Arrays- Index arrays are the arrays in which elements of arrays can be accessed through numerical index. In index arrays, keys are the numbers

There are two ways to create indexed arrays:

We can create arrays as variables are created. We have to use square brackets around them for denoting the index.

$state[1] = 'Punjab"

$state[2] = 'Haryand"

$state[3] = 'Delhi"

The array() function can also be used to create an array in PHP:

array();

The index can be assigned automatically (index always starts at 0), like this example:

Example:

$day = array("sunday", "monday", "tuesday", "wednesday","thrusday","Friday", "Saturday");

In this example, We have declared day as an index array with seven elements. PHP's index arrays begin with subscript 0 not 1. Indexes can also be assigned numerically as

$day[0] = “Sunday”

$day[1] = “Monday”

$day[2] = “Tuesday”

$day[3] = “Wednesday”

$day[4] = “Thursday”

$day[5] = “Friday”

$day[6] = “Saturday” and so on

9.2.1 Initialization of Arrays

Initialization is a process of setting the initial values of array elements. You need not worry about the indexing. PHP do it automatically for us. You can create one element n the array and then another with the same name:

```
$Country[] =”India”;
$Country []=”America”;
$Country []=”Britain”;
```

Square brackets in above assignments indicates that values are to be store in an array named $Country PHP automatically store the above values in $Country[0], $Country[1] and $Country[2].

We can also use the second method using explicit index values to initialize an array as:

```
$Country[0] =”India”;
$Country[1] =”America”;
$Country[2] =”Britain”;
```

Program to illustrate the concept of index arrays

```
<?php
$day = array("sunday", "monday", "tuesday", "wednesday","thrusday",”Friday”, “Saturday”);

foreach ($day as $value)
 {
   echo "$value <br>";
}
?>
```

Output:

This program will display all the elements of the array.

9.3 Associative Arrays

Associative arrays are the arrays with having named keys associated with the elements of the array. In associative arrays keys bears a direct relation to its corresponding values.

There are two methods to create an associative array in PHP.

First Method: We can use named keys directly to create associative array as:

```
        $agestudent['Ram'] = "21";
        $agestudent['Mohan'] = "22";
        $agestudent['Sohan'] = "20";
        $agestudent['Geeta'] = "19";
```

Second Method: We can use array() function to create associative array as:

```
$agestudent = array("Ram"=>"21", "Mohan"=>"22, "Sohan"=>"20","Geeta"=>"19");
```

Program to illustrate the concept of associative arrays

```
<?php
$agestudent = array("Ram"=>"21", "Mohan"=>"22, "Sohan"=>"20","Geeta"=>"19"); //Associative
array
echo "Ram is " . $age['Ram'] . " years old.";
        echo "Mohan is " . $age['Mohan'] . " years old.";
        echo "Sohan is " . $age['Ram'] . " years old.";
        echo "Geetais " . $age['Ram'] . " years old.";
        ?>
```

Output:

This Program will print ages of all elements of array agestudent.

We can also use foreach loop to access the elements of associative array as

```
<?php
$agestudent = array("Ram"=>"21", "Mohan"=>"22, "Sohan"=>"20","Geeta"=>"19"); //Associative
array

foreach($agestudent as $n => $n_value) {
   echo "Key=" . $n . ", Value=" . $n_value;
   echo "<br>";
}
?>
```

This Program will also print ages of all elements of array age student.

9.4 count() Function

In PHP, the count() function is used to find the number of elements in an given array.

Example

```
<?php
```

```
//Program to illustrate count function
$days = array("sunday", "monday", "tuesday", "wednesday","thrusday",”Friday”, “Saturday”);
echo count($days);  //Print the length of array days
}
?>
```

9.5 Range() Function

In PHP, the range() function is used to create an array consisting of range of high and low values.

The syntax is

```
0Range(starting value, End value, [step]);
```

Where step is optional.

Examples:

```
$a=range(1,10);
//same as $a=array(1,2,3,4,5,6,7,8,9,10)
$b=range(1,10,2);
//same as $b=array(1,3,5,7,9)
```

The range function can also be used for creating characters array as

```
$alphabets=range(“a”, “d”);
//same as $alphabets=array(“a”, “b”, “c”, “d”);
```

9.6 Forms in PHP

A form on a web page allows the user to input the data. We can create forms using HTML. We can only enter the data through forms in HTML. We can process user data using PHP. The two common methods GET and POST can be used to pass data in PHP. The super global variable $_POST and $_GET can be used for this purpose. The following example will illustratate the concept of accesing data using POST.

Example

```
<html>
<head>
<title>Student Form</title>
</head>
<body>
<form action="input.php" method="post">
Name: <input type="text" name="name"><br>
```

```
    FName: <input type="text" fname="fname"><br>
    Class: <input type="text" name="class"><br>
    <input type="submit">
    </form>
    </body>
    </html>
```

The user will fill the data and press the submit button, the data will go to PHP file named "input.php". Suppose user has filled mohan as name, sohan as father’s name and PGDCA as class. In this example, we have used HTTP POST method to send the data.

We will write the code of "input.php" as:

```
<html>
<body>


Your name is  <?php echo $_POST["name"]; ?><br>
Your father’s name is  <?php echo $_POST["fname"]; ?><br>
Your class is: <?php echo $_POST["class"]; ?>

</body>
</html>
```

The output could be something like this:

```
Your name is Mohan
Your father’s name is Sohan

Your class is PGDCA
```

The following example will illustrate concept of accessing data using GET.

Example

```
<html>
<head>
<title>Student Form</title>
</head>
<body>
<form action="input.php" method="get">
Name: <input type="text" name="name"><br>
```

```
FName: <input type="text" fname="fname"><br>
Class: <input type="text" name="class"><br>
<input type="submit">
</form>
</body>
</html>
```

The user will fill the data and press the submit button, the data will go to PHP file named "input.php". Suppose user has filled mohan as name, sohan as father’s name and PGDCA as class. In this example, we have used HTTP GET method to send the data.

We will write the code of "input.php" as:

```
<html>
<body>

Your name is  <?php echo $_GET["name"]; ?><br>
Your father’s name is  <?php echo $_GET["fname"]; ?><br>
Your class is: <?php echo $_GET["class"]; ?>

</body>
</html>
```

The output of above code will be:

```
Your name is Mohan
Your father’s name is Sohan

Your class is PGDCA
```

We can see that output of GET and POST is same

9.7 Difference between GET and POST

Both $_GET and $_POST are super global arrays, which means that they are always accessible from any function. GET is the default method of submitting the data whereas POST is not default method. Data will be visible in the page address field in case of GET method. We should not use GET method while sending sensitive information. Get is used for short and non-sensitive data. GET should not be used for sending passwords. Data will be not visible in the page address field in case of POST method. We can use POST method for sending personal and sensitive data. Get is used for short and non-sensitive data. POST has not any size limitations. We can use POST method for sending large amount of data. GET is passed to the script via the URL parameters whereas . POST is passed to the script via the HTTP POST method. Generally, Programmers prefers POST method to send data.

9.8 Summary

Array is a collection of homogeneous type of data. An Array can store more than one value at a time. An Array is a group of data items that share a common name. For example, we can define an

Array name marks to represent a set of marks of a group of students. Each member of the array is known as element. Elements of the array share certain common characteristics. Arrays can be of two types: Index Arrays and Associative Arrays. Index arrays are the arrays in which elements of arrays can be accessed through numerical index. In index arrays, keys are the numbers. Associative arrays are the arrays with having named keys associated with the elements of the array. In associative arrays keys bears a direct relation to its corresponding values. Count () function is used to find the number of elements in an given array. Range () function is used to create an array consisting of range of high and low values. A form on a web page allows the user to input the data. We can create forms using HTML. We can only enter the data through forms in HTML. We can process user data using PHP. The two common methods GET and POST can be used to pass data in PHP. The super global variable $_POST and $_GET can be used for this purpose. Both $_GET and $_POST are super global arrays, which means that they are always accessible from any function. GET is the default method of submitting the data whereas POST is not default method. Data will be visible in the page address field in case of GET method. We should not use GET method while sending sensitive information. Get is used for short and non-sensitive data. GET should not be used for sending passwords. Data will be not visible in the page address field in case of POST method. Programmers prefers POST method to send data.

9.9 Objective types question/answers to check your progress

Q1: Array is a collection of:

Ans: homogeneous type of data

Q2: Name the various types of array n PHP

Ans: index array and associative array

Q3: which function is used to create an array in PHP?

Ans: array()

Q4: which function is used to find the number of element in array

Ans: count()

Q5: Write down the syntax of range() function

Ans: range(starting_value,ending_value,[size]);

Q6: In PHP which two common method are used to pass data by using form?

Ans: GET and POST.

Q7: Which default method is used for submitting the data in form?

Ans: GET

9.10 References:

9.10.1 Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff
9.10.2 Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress
9.10.3 http://php.net/manual
9.10.4 https://www.w3schools.com/php
9.10.5 https://www.tutorialspoint.com/php
9.10.6 http://www.w3programmers.com

9.11 Suggested Readings:

9.11.1 PHP: The Complete Reference by Steven Holzner Publisher: Tata McGraw Hill

9.11.2 Programming PHP by Kelvin Tetroi
9.11.3 PHP 6 and MYSQL Bible by Steve Suehring, Wiley India edition
9.11.4 Beginning PHP by Wankyu Choi, Allan Kent, Chris Lea, Ganesh Pradash and Chris Uliman Publisher: Shroff
9.11.5 Beginning PHP and Oracle by W. Jason Gilmore and Bob Bryla; Publisher: Apress

9.12 Model Questions

Q1: What is an array? How can you create array in PHP? Explain with examples.

Q2: What is Index array? Explain with examples in PHP.

Q3: Explain the purpose and use of associative arrays.

Q4: Explain the utility of range() and count() functions.

Q5: What do you mean by form? How can you process data in PHP?

Q6: Draw the difference between GET and POST method.

****

FILE AND DIRECTORY HANDLING

Structure

10.0 Objective:

This chapter will cover following topics:

- Introduction to files and directories
- Opening the files with fopen()
- Closing the files with fclose()
- Reading the files with fread()
- Writing to files with fwrite()
- Copy the files with copy()
- Deleting the files with unlink()
- Renaming a file with rename()
- Opening a directory using opendir() function
- Closing a directory using closedir() function
- Reading directory entries
- Other directory functions

10.1 Introduction to files and directories

A file is used to store data and information. A File is a collection of bytes stored on hard disk , floppy disk , pen drive, CD, DVD or some another storage media. A directory is a special type of file that contains files and sub directories and pointers to their storage area on the storage medium.

10.2 Opening the files with fopen()

Fopen() function in PHP is used to open a file. It returns a file handle associated with the file. fopen() can take either two arguments: file name and mode.

Fopen() returns false if PHP fails to open file. Otherwise fopen() returns a file pointer which can be used for reading or writing to the file. the general syntax of opening a file is:

File pointer = fopen("name of file", "mode");

The possible modes in PHP are shown in given table below:

Mode	**Purpose**
R	Open file for reading only. The file pointer is placed at the beginning of the file.
r+	Open file for reading and writing both. The file pointer is placed at the beginning of the file.
W	Open file for writing only. Existing data will be lost. If the file does not exist,

	PHP will create it. The file pointer is placed at the beginning of the file.
w+	Open file for reading and writing both. Existing data will be lost. If the file does not exist, PHP will create it. The file p ointer is placed at the beginning of the file.
A	Open file for appending only . If the file does not exist, PHP will create it. The file pointer is placed at the end of the file. Data is written at the end of existing file.
a+	Open file for reading and appending both. If the file does not exist, PHP will create it. The file pointer is placed at the end of the file. Data is written at the end of existing file.

Example:

```
$fp = fopen(“./abc.txt”,“r”);
```

Output:

abc.txt will be opened in read mode.

10.3 Closing the files with fclose()

Fclose() function in PHP is used to close a file. After making changes to a opened file, it must be closed with fclose() function. Fclose function takes file pointer as argument and returns true on successful closure of file. Otherwise, fclose() returns false on failure.

The general syntax of closing a file is:

fclose(file pointer)

Example:

```
fclose ($fp)
```

Output:

The file associated with the file pointer fp will be closed.

10.4 Reading the files with fread()

fread() function in PHP is used to read a contents of a file. The file must be opened before reading. fread() function requires two arguments: the file pointer and length of files in bytes.

The general syntax of reading a file is:

fread(file pointer, size)

Example:

```
$fp = fopen(“./abc.txt”,“r”);
$str = fread ($fp,15)
```

Output:

The first fifteen characters of file “abc.txt” will be assigned to the variable $str.

10.5 Filesize()

Filesize() function in PHP is used to find the length of a file. filesize() function takes the file name as argument. It returns the size of the file in bytes. The following steps are required to read a file in PHP:

- Firstly, Open a file using fopen() function.
- Get the length of file using filesize() function.
- Read the contents of file using fread() function.
- Close the file using fclose() function.

The following example will illustrate the use of fread() function in PHP:

```
<html>

  <head>
    <title>Reading the contents of a file </title>
  </head>

  <body>

    <?php
           //program to illustrate the concept of fread() function
      $filename = "abc.txt";
      $fp = fopen( $filename, "r" );
      if( $fp == false )
          {
        echo ( "File cannot be opened” );
        exit();
      }

      $size = filesize( $filename );
      $filetext = fread( $fp, $size );
      fclose( $fp );
      echo ( "File size : $size bytes" );
      echo ( "<pre>$filetext</pre>" );
    ?>
     </body>
</html>
```

Output:

This program will print

Tthe size and contents of file “abc.txt”.

10.6 Writing to files with fwrite()

fwrite() function in PHP is used to write data to a file. fwrite() function takes two arguments: file pointer and string of data to be written in the file. The general syntax is:

```
fwrite (file pointer, string, [integer value]);
```

Where file pointer is the pointer that is associated with the file where data is to be written. String is the data. Integer value is optional that specify the length of data to be written in the file.

Example:

First of all we will open a file with write mode as-

```
$fp = fopen (“abc.txt”,”w”)
```

Then we will use fwrite() to write data to the file as-

```
fwrite ($fp, “My Name is Mohan”);
```

The string “My Name is Mohan” will be written into the beginning of file abc.txt.

Program:

The following example will illustrate the use of fwrite() function in PHP:

```
<html>
  <head>
    <title>writing the datato a file </title>
  </head>
    <body>
<?php
  //program to illustrate the concept of fwrite() function
  $filename = "abc.txt";
  $fp = fopen( $filename, "w" );
  if( $fp == false )
  {
    echo ( "Error in opening abc.txt file" );
    exit();
  }
  fwrite( $fp, "Hello, How are you" );
  fclose( $fp );
?>
  </body>
</html>
```

Output:

This program will write the string:

"Hello, How are you" to the file “abc.txt”.

10.7 Copying the files with copy()

Copy() function in PHP is used to copy the contents of a file to another file. This function requires takes two arguments: First argument for source file and second for destination file. The syntax of copy() function is:

Copy(source file name, target file name)

Example:

Copy(“abc.txt”, “def.txt”)

Output:

The contents of abc.txt will be copied to def.txt.

10.8 Deleting the files with unlink()

unlink () function in PHP is used to delete a file. This function requires only one string argument referring to the file name whidch we want to delete.The syntax of unlink() function is:

unlink (file name);

Example:

unlink (“abc.txt”);

Output:

The file abc.txt will be deleted.

10.9 Renaming the files with rename()

rename() function in PHP is used to rename a file. This function requires takes two arguments: First argument for old file name and second for new file name. The syntax of rename()function is:

rename(old file name, new file name);

Example:

rename(“abc.txt”, “def.txt”);

Output:

The name of abc.txt will be changed to def.txt.

10.10 opendir():

A directory is a special type of file that contains files and sub directories and pointers to their storage area on the storage medium. In PHP, You can manipulate directories in the same way as the files. Opendir() function in PHP is used to open a directory. It returns a directory pointer associated with the directory. Opendir () requires name of the directory as argument. opendir() returns false if PHP fails to open directory. Otherwise opendir() returns adirectory pointer which can be used for reading or writing to the directory. The general syntax of opening a directory is:

Directory pointer = opendir(“name of directory”);

Example:

$dp = opendir (“mohan”)

10.11 closedir():

Closedir() function in PHP is used to close a already opened directory. Closedir() function requires directory poingter as the argument. Closedir()) returns false if PHP fails to close the directory. The general syntax of closing a directory is:

closedir(directoy pointer);

Example:

closedir($dp);

10.12 readdir():

Reading directory entries using readdir()

The readdir() function in PHP returns the next entry listed in the opened directory. The general syntax of readdir() function is:

Readdir(directory pointer);

Example:

<?php

```
//Program to illustrate the concept of readdir() function
$dp = opendir (“mohan”);
$file=readdir($dp);
Echo($file);
Closedir($dp)l
?>
```

Output:

This program will print the name of first entry in the directory named mohan.

10.13 Other directory functions

PHP provides a lot of functions to manipulate directories. Some of them are listed below:

Name of function	Working
Chdir()	Change the directoy
Rmdir()	Remove a directory
Mkdir()	Creates a new directory
Chroot()	Change the root directory
opendir()	Opens the directory handle
Getcwd()	Returns the current working directory
Rewinddir()	Rewinds the directory handle

10.14 Summary:

A file is used to store data and information. A File is a collection of bytes stored on hard disk , floppy disk , pen drive, CD, DVD or some another storage media. A directory is a special type of file that contains files and sub directories and pointers to their storage area on the storage medium. Open() function in PHP is used to open a file. It returns a file handle associated with the file. fopen()can take either two arguments: file name and mode. Fclose() function in PHP is used to close a file. After making changes to a opened file, it must be closed with fclose() function. Fclose() function takes file pointer as argument and returns true on successful closure of file. Otherwise, fclose() returns false on failure. fread() function in PHP is used to read a contents of a file. The file must be opened before reading. fread() function requires two arguments: the file pointer and length of files in bytes. Filesize() function in PHP is used to find the length of a file. filesize() function takes the file name as argument. It returns the size of the file in bytes. fwrite() function in PHP is used to write data to a file. fwrite() function takes two arguments: file pointer and string of data to be written in the file. Copy() function in PHP is used to copy the contents of a file to another file. This function requires takes two arguments: First argument for source file and second for destination file. unlink () function in PHP is used to delete a file. This function requires only one string argument referring to the file name which we want to delete. rename() function in PHP is used to rename a file. This