

(Deemed University Established Under Section 3 of UGC Act 1956) Coimbatore - 641021. (For the candidates admitted from 2015onwards) DEPARTMENT OF COMPUTER SCIENCE, CA & IT

| SUBJECT : WEB TECHNOLOGY | SEMESTER: VI | LTPC |
|--------------------------|---------------------|---------|
| SUBJECT CODE: 15CSU602 | CLASS : III B.Sc.CS | 4 1 0 5 |
| UNIT-I | | |

HTML: Introduction to HTML- History Of HTML-Structure of HTML-Formatting Text :Font type, Font Size, Big ,Small, bold ,italic, color, superscript , Subscript, striking out, Underlining the text , Predefined fonts , Pre formatted Text, Blinking Text and Block Quotes. Lists: Ordered, Unordered and Definition List. Creating Link - Images. Tables: Creating Table –Dividing Table into Columns- Dividing Table into Rows- Creating Headers- Adding Border –Putting a Background Image- Heading across two or more columns- Changing color of the cell-aligning the content –Display of Tables. Frames.

UNIT-II

Forms: Working with forms-Creating forms-working with menus- working with Radio buttons- check boxes-textboxes-text areas- password boxes-submit button-Resetting the form. DHTML: Heading and Horizontal line-Hidden Message-Message at the center of the page- Moving Boxes- Changeable Box-CSS: Introduction- Creating Style Sheets-Common Tasks with CSS- Colors-The Font Family.

UNIT-III

JavaScript: Introduction-Operators-Starting with JavaScript: Using Quotes-Using Alerts- Functions-Variables-data types- Statements-Comments. Objects: Working with Objects- Date Object-Math Object-String Object--Handling Events in JavaScript-Event Handling attributes-Window Events-Window Object – Document Object-Navigator Object

UNIT-IV

PHP: Introduction-What is PHP?-Why PHP?-Basic PHP syntax-Comments in PHP-PHP5 varaiables-

PHP5 echo and Print Statements- How Online PHP Programs Runs- PHP Datatypes-PHP String Functions UNIT-V

Functions-Arrays-PHP Form Handling-GET,POST Methods-Form Validation-PHP File Handling-PHP Exception Handling. MySQL: Introduction to MySQL -Connecting to MySQL from a PHP application-Inserting and updating records in table- Deleting and retrieving data from table.

TEXT BOOK

 Ramesh Bangia. 2014. Web Technology, 1st Edition, Firewall Media Publications, New Delhi (Chapters : 4,5,6 and 11) Dave W.Mercer, Allan Kent, Steven D.Nowicki, Davd Mercer, Dan Squie, Wankyu Choi.2015. Beginning PHP5. Wiley India (P) Ltd, New Delhi.

REFERENCES

- Deitel H.M., P.J.Deitel and A.B.Goldberg. 2012. Internet & World Wide Web, 3rd Edition, Pearson Education Asia, New Delhi.
- 2. Jeffrey C. Jackson. 2014. Web Technologies, 3rd Edition, Pearson Education Publishers, New Delhi.
- 3. Rohit Khurana. 2015. JavaScript, 1st Edition, A.P.H Publishing, New Delhi.
- Xavier C. 2013. Web Technology and Design, 1st Edition, New Age International Publishers, New Delhi.
- 5. Luke welling, Laura Thomson, 2011. PHP and MySQL Web Development, 4th Edition, Pearson education.
- Tim Converse & Joyce Park with Clark Morgan . 2015. PHP5 & MySQL Bible, 1st Edition, John Wily, India.

WEB SITES

- 1. www.w3schools.com/
- 2. alexle.net/archives/category/web-technolgy
- 3. jmarshall.com/easy/
- 4. www.php.net/
- 5. en.wikipedia.org/wiki/PHP
- 6. <u>www.w3schools.com/PHP/DEfaULT.asP</u>

ESE MARKS ALLOCATION

| | Section A | |
|---|--|----|
| 1 | 20 X 1 = 20 Online Examination | 20 |
| | Section B | |
| 2 | 5 X 8 = 40 Either ' A ' OR ' B ' Choice | 40 |
| 3 | Total | 60 |



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

<u>Unit –I</u>

<u>Syllabus</u>

HTML: Introduction to HTML- History Of HTML-Structure of HTML-Formatting Text :Font type, Font Size, Big ,Small, bold ,italic, color, superscript , Subscript, striking out, Underlining the text , Predefined fonts , Pre formatted Text, Blinking Text and Block Quotes. Lists: Ordered, Unordered and Definition List. Creating Link - Images. Tables: Creating Table –Dividing Table into Columns- Dividing Table into Rows- Creating Headers- Adding Border –Putting a Background Image- Heading across two or more columns- Changing color of the cell-aligning the content –Display of Tables. Frames.

Introduction to HTML:

- HTML stands for Hyper Text Markup Language.
- Hyper text is the text displayed on a computer display or other electronic devices with references to other text.
- The references can be otherwise called as hyperlinks which links the other document with current document.
- The document / file that contain a hypertext can be called as Hyper Document. It is of two types.
 - ✓ **Static:** Document is already prepared and stored in advance.
 - ✓ **Dynamic:** Changes according to the user's input.
- Using hypertext complex and dynamic systems of linking and cross referencing can be developed.
- HTML is a markup language which contains a set of markup tags is used to describe the web pages.
- HTML documents / web pages mainly contain HTML tags and plain text.
- The HTML tags are the keywords that are surrounded by angle brackets E.G:-"<keyword>".
- Each HTML tags will be in pairs called as start and end tags. With the end tag having a slash "/" in it.

History of HTML:

• 1960s-GML (Generalized Markup Language) was developed by Charles Goldfarb, Edward Musher and Raymond Lorie at IBM for organizing vast amount of documentation.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

- 1978- ANSI took basics of GML and fashioned nationwide standard called GCA.
- Six years later ISO began to work on new global version called SGML (Standard Generalized Markup Language).
- 1989-Tim Berners Lee a physicist at CERN(Centre European pour la Recherche Nucleaire) created a method for the scientists to share papers. In the same year a memo proposing hypertext system was released.
- Late 1990- Tim Berners Lee specified HTML and wrote a browser and server software.
- Late 1991-public description of HTML was released containing 18 elements.
- 1995-HTML 2.0.
- Jan 1997-HTML 3.2 (W3C Recommendation).
- Dec 1997- HTML 4.0 (W3C).
- April 1998 HTML 4.0 got minor edits.
- Dec 1999-HTML 4.0.1.
- Jan 2008-HTML 5.
- Mean while two languages related to hyper text where developed.
 - ✓ DHTML→combination of HTML 4, Cascading Style Sheets and JavaScript to create dynamic Web pages. These work on basis of DOM(Document Object Model) to change the content of the web page even if it's loaded into the browser.
 - ✓ XML→XML stands for Extensible Markup Language using which we can create our own markup languages. It is developed by W3C.

Structure of HTML:

The basic structure of a HTML document and the description of the tags present in it are as below.

- **<!DOCTYPE>-** Defines the document type.
- <HTML>- indicates the browser that it is a HTML document and </HTML> tells the browser that HTML document is completed.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

• **<HEAD>-** container for all the head elements.

<!DOCTYPE html> <html> <head><title>HTML</title></head> <body>

<h1>MY FIRST PROGRAM</h1>

SAMPLE

</body> </html> **OUTPUT:**

MY FIRST PROGRAM

SAMPLE

- ✓ <TITLE>- defines title of the document, title in the browser toolbar, title for the page when it is added to favorites, title for the page in search-engine results. (E.g):- <title>Example
 Title</title>
- ✓ **<STYLE>-** style information for an HTML document is defined here (i.e):- how HTML elements should render in a browser is defined.

(E.G):-

<style>

h1{color:red;}

p{color:blue;}

</style>

**As per the above example wherever the h1 or p element is used then their color will be in blue and red only, because the style is specified in the STYLE element. The same sort of style specification will be used in the cascading style sheets which are maintained as separate files and those files will be linked into the current document using the link tag in the HEAD itself.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

✓ <BASE>- specifies the base URL/target for all relative URLs in a document. Contains either a href attribute or target attribute or sometimes both. All the elements at the body of the html document will be referring the base tag's.

(E.G):-

<base href="www.w3schools.com/images">

</base>



In the above example if the url in img tag is not able to give the image then the base url will be referred if present then the image will be displayed.

 \checkmark <LINK>- defines the relationship between a document and an external resource.

(E.G):-

k rel="stylesheet" type="text/css" href="sample.css">

In the above example a document which is external called sample.css is linked. It is a cascading style sheet file linked using href attribute. Apart from that there are two more



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

attributes called type-which specifies the type of file linked and rel-specifies the type of link between the current document and the external file. Most probably the style sheet files will be linked in to a HTML document.

✓ <META>- provides metadata about the HTML document. Metadata will not be displayed on the page. Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata. The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

(E.G):-

```
<meta charset="UTF-8">
<meta name="description" content="Example">
<meta name="keywords" content="HTML">
<meta name="author" content="BSCCS">
```

Here in the example the character set details, description details, details about keyword in document and the author name are specified for the use of browsers, search engines etc.

 ✓ <SCRIPT>- used to define a client-side script, such as a JavaScript or points to external source using src attribute. The external source may be the java script file. In the HTML document the script may be written without a function or with function.

The advantage of using function is whenever a particular event occurs in the HTML document then we can easily call the function for the script to be executed on the client side itself.

(E.G):-

<html>

<head>



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

```
<script type="text/javascript">
function b_click()
{
    alert("hello");
}
</script>
</script>
</head>
<body>
<input type="button" onclick="b_click()" value="Hello button"/>
</body>
</html>
Output:
```





CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

In the above example there is command button in the html body. A java script is also written in the head tag. When the button is pressed the coding is written in a way that the alert of hello will be displayed in a dialog box. Here the script ran in the client side.

- ✓ <NOSCRIPT>- defines an alternate content for users that have disabled scripts in their browser.
- **<BODY>-** contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc. This is the place where the actual content of the web page required for display will be given.

The attributes of the body tag are

- ✓ alink- The alink attribute specifies the color of an active link in a document. The default color of the active link in a document will be red.
- ✓ background- specifies a background image for a document using the image's URL. Proper image that is image with correct resolution should be chosen to display and if not then the display image will be inconvenient to the user/client.
- ✓ bgcolor- specifies the background color of a document. For specifying the color of the document the either the name of the particular color can be used or the hexadecimal value for a color can be used or rgb value can be used.
- ✓ link- specifies the default color of unvisited links in a document. The default color of an unvisited in a HTML document will be blue.
- ✓ text- used to specify the color of text displayed in a document. Default color of the body text is black or it may differ based on the web browser of the client.
- ✓ vlink- used to specify the color of visited links displayed in a document. All the visited links in a HTML document will be pink in color.

Formatting Text:

In general a text in any kind of document is subject to some formatting. Formatting is a process where the properties of text in document will be altered. Here mainly the text's color, size and style will be changed using the tag . The changes are made using the attributes in the font tag.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

Font type:

- ↓ Font type is the type of font that can be used to display a text in a web page or some other media.
- The font type for a text is specified in a web page, the client using that website can experience that font.
- One of the tags used for text formatting in HTML is FONT tag. The attribute used to specify the font type is FACE.
- **4** Single font type or multiple font types separated by commas can be specified in a font tag.
- If the specified font is not in client's web browser then the other font after comma will be taken in multiple font specification. If all the fonts are not in client side then client's default font will be chosen.

4

4 (E.G):- This is some text!

The output for above example code will be:

EXAMPLE

This is some text!

In the above example the font face given is verdana and if the client browser is not supporting that font then a search will be done if there is another font specified or not. If not then the default font of the client browser will be chosen.

Font size:

4 The size attribute specifies the size of the text inside a element.

(E.G):- This is some text!

The output for above example will be



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

In the example the font size is given as a numeral and that particular size will be applied. The size range for fonts in web browsers is 1 to 7. Apart from numerical value the values BIG and SMALL can be given for font size which results in relative large and small size of the text displayed.

Font color:

- ↓ The color attribute specifies the color of the text inside a element.
- Syntax:
- This is some text!

Like the bgcolor attribute the color for font can be specified by giving the color's name as it is or giving the hexa decimal value or giving the rgb component value. When hexadecimal values are used more number of colors can be viewed by the developer of that web page but the final color will be only one.

Small:

- ↓ The <small> tag defines smaller text
- ↓ (E.G):- <small>Copyright 1999-2050</small>





CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

<u>Big:</u>

- \downarrow The <big> tag defines bigger text.

EXAMPLE Bigger text

Bold:

- ↓ The tag specifies bold text
- \downarrow (E.G):- This is normal text and this is bold text.
- ↓ The output for above example will be

EXAMPLE This is normal text This is bold text

Italic:

The <i> tag defines a part of text in an alternate voice or mood. The content of the <i> tag is usually displayed in italic.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

- The <i> tag can be used to indicate a technical term, a phrase from another language, a thought, or a ship name, etc.
- ↓ (E.G):- He named his car <i>The lightning</i>, because it was very fast.
- 4 The output for above example will be.



Super Script:

- The <sup> tag defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW.
- ↓ (E.G):- This text contains ^{superscript} text.
- ↓ The output for above sample code

EXAMPLE

This text contains superscripttext

Sub script:

- ♣ The <sub> tag defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H₂O.
- ↓ (E.G):- This text contains _{subscript} text.
- **4** The output for above sample code will be

| | EXAMPLE | | |
|------------|----------------------------------|--|-------|
| Prepared E | | Of Computer Science, CA & IT, KAHE, Page | 11/38 |
| | This text contains subscripttext | | |



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

Striking Out:

- **4** The <strike> tag defines strikethrough text.
- (E.G):- Version 2.0 is <strike>not yet available!</strike> now available!
- ↓ Output for above code will be

EXAMPLE

Version 2.0 is not yet available!

now available!

- text that should be stylistically different from normal text, such as misspelled words or proper nouns in Chinese.
- ↓ (E.G):- This is a <u>paragraph</u>.
- ↓ Output for above code will be





CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

Blinking text:

The HTML <bink> tag is used to enclose a text to make it blink. This tag was supported by Netscape and now this is obsolete.

Block quotes:

- ↓ The <blockquote> tag specifies a section that is quoted from another source.
- ♣ (E.G):-

<blockquote cite="http://www.worldwildlife.org/who/index.html">

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

</blockquote>

EXAMPLE

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

Predefined fonts:



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

HTML has some tags that can be used to apply the properties of predefined fonts for the text in web page they are as follows.

1) Teletype:

The <tt> tag is used to define a teletype text.

(E.G):-

<tt>Teletype text</tt>

2) Keyboard text:

The <kbd> tag is used to give a keyboard input type font appearance to the web page text.

(E.G):- <kbd>Keyboard input</kbd>

3) Sample text:

The SAMP tag for sample text indicates the font type of sample output from a computer program.

(E.G):- <samp>Sample output from a computer program</samp>

4) Code:

The CODE tag is used to define a piece of computer code.

(E.G):- <code>A piece of computer code</code>

Preformatted text:

Any text between the opening tag and the closing tag will preserve the formatting of the source document.

(E.G):-

function testFunction(strText){





CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

alert (strText)

}

This will produce following result:

EXAMPLE function testFunction(strText){ alert (strText)

}

Lists:

Listing of items, subjects or menu in the form of a list. In HTML there are three different types of lists.

- An unordered list. This will list items using bullets
- A ordered list. This will use different schemes of numbers to list your items
- <dl>- A definition list. This is arrange your items in the same way as they are arranged in a dictionary.

Unordered Lists:

An unordered list is a collection of related items that have no special order or sequence. The most common unordered list on the Web is a collection of hyperlinks to other documents.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

(E.G):-

The following example list is created by using tag. Each item in the list is marked with a butllet. The bullet itself comes in three flavors: squares, discs, and circles. The default bullet displayed by most web browsers is the traditional full disc.

One Movie list is given below:

| <center></center> |
|----------------------------|
| <h2>Movie List</h2> |
| |
| |
| Independence Day |
| Horror |
| Titanic |
| Ghost in the ship |
| |
| |

This will produce following result:

Movie List

- Independence Day
- Horror
- Titanic
- Ghost in the ship

Type attribute can be used to specify the type of bullet you like. By default it is a disc. Following are the possible way:



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

| <ul type="square"> | <ul type="disc"> | <ul type="circle"> |
|--|---|--|
| Hindi English Maths Physics | HindiEnglishMathsPhysics | Hindi English Maths Physics |

Ordered Lists:

The typical browser formats the contents of an ordered list just like an unordered list, except that the items are numbered instead of bulleted. The numbering starts at one and is incremented by one for each successive ordered list element tagged with

This list is created by using *<*ol> tag. Each item in the list is marked with a number.

One Movie list is given below:

<center> <h2>Movie List</h2> </center> <ub> Independence Day Horror Titanic Ghost in the ship

This will produce following result:



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

Movie List

- 1. Independence Day
- 2. Horror
- 3. Titanic
- 4. Ghost in the ship

Definition Lists:

HTML and XHTML also support a list style entirely different from the ordered and unordered lists we have discussed so far - definition lists . Like the entries in a dictionary or encyclopedia, complete with text, pictures, and other multimedia elements, the Definition List is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

- <dl> Defines the start of the list
- $\langle dt \rangle$ A term
- <dd>- Term definition
- </dl> Defines the end of the list

Example:

<dl> <dt>HTML</dt> <dd>This stands for Hyper Text Markup Language</dd> <dt>HTTP</dt> <dd>This stands for Hyper Text Transfer Protocol</dd> </dl>

This will produce following result:



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

HTML

This stands for Hyper Text Markup Language

HTTP

This stands for Hyper Text Transfer Protocol

LINKS:

- Web pages can contain links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks.
- Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images.
- A link is specified using the <a> element. This element is called **anchor tag** as well. Anything between the opening <a> tag and the closing tag becomes part of the link and a user can click that part to reach to the linked document.
- Following is the simple syntax to use this tag.

Attributes:

Following are most frequently used attributes for <a> tag.

- **href:** specifies the URL of the target of a hyperlink. Its value is any valid document URL, absolute or relative, including a fragment identifier or a JavaScript code fragment.
- **target:** specify where to display the contents of a selected hyperlink. If set to "_blank" then a new window will be opened to display the loaded page, if set to "_top" or "_parent" then same window



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

will be used to display the loaded document, if set to "_self" then loads the new page in current window. By default its "_self".

- **name & id:** attributes places a label within a document. When that label is used in a link to that document, it is the equivalent of telling the browser to go to that label.
- **title:** attribute lets you specify a title for the document to which you are linking. The value of the attribute is any string, enclosed in quotation marks. The browser might use it when displaying the link, perhaps flashing the title when the mouse passes over the link.

(E.G):-

YAHOOOOOOOOOOOOO

The above html code will yield the following output



**In the above program the link has an address to a web page when its clicked. Here the address will lead the user to yahoo website. The title attribute has a value of Yahoo link and it will be displayed when the user points the link in screen. As the target was set to self the link's web page will open in the user's current page.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

If there is no internet connectivity then another page will be displayed showing the message of inability to open the required web page successfully.



Images:

Any image can be inserted in web page by using tag. Following is the simple syntax to use this tag.

Attributes:

Following are most frequently used attributes for tag.

- width: sets width of the image. This will have a value like 10 or 20% etc.
- height: sets height of the image. This will have a value like 10 or 20% etc.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

- **border:** sets a border around the image. This will have a value like 1 or 2 etc.
- **src:** specifies URL of the image file.
- **alt:** this is an alternate text which will be displayed if image is missing.
- **align:** this sets horizontal alignment of the image and takes value left, right or center.
- valign: this sets vertical alignment of the image and takes value top, bottom or center.
- hspace: horizontal space around the image. This will have a value like 10 or 20% etc.
- **vspace:** vertical space around the image. This will have a value like 10 or 20% etc.
- **name:** name of the image with in the document.
- **id:** id of the image with in the document.
- **style:** this will be used if you are using CSS.
- **title:** specifies a text title. The browser, perhaps flashing the title when the mouse passes over the link.

Example:

Output:

HTML IMAGE

TABLES:

Creating a Table:



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

- Tables are very useful to arrange in HTML and they are used very frequently by almost all web developers.
- **4** Tables are just like spreadsheets and they are made up of rows and columns.
- ♣ A table is created by using tag. Inside element the table is written out row by row.
- 4 A row is contained inside a $\langle tr \rangle$ tag . which stands for table row.
- \blacksquare Each cell is then written inside the row element using a $\langle td \rangle$ tag . which stands for table data.

```
(E.G):-

Cell1

Cell2

Cell2

Cell3

Cell4

Cell4
```

This will produce following result:

| Cell1 | Cell2 |
|-------|-------|
| Cell3 | Cell4 |

Dividing table into columns:

The table can be divided onto columns in two ways one is COLGROUP (structured) and another one is COL (unstructured).

• The <colgroup> tag specifies a group of one or more columns in a table for formatting.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

- The <colgroup> tag is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row.
- The <col> tag specifies column properties for each column within a <<u>colgroup></u> element.
- The <col> tag is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row.
- There are common attributes that can be used in these two tags, they are as follows.
 - Align: The align attribute specifies the horizontal alignment of the content. It is supported only in opera browser.
 - ✓ Span: The span attribute defines the number of columns. Its supported by a variety of browsers.
 - ✓ **Valign:** The valign attribute specifies the vertical alignment of the content.
 - ✓ **Width:** The width attribute specifies the width.
 - ✓ **Style:** The style attribute can be used to define the overall style for column group or col tag.

(E.G):-

```
<colgroup>
```

```
<col span="2" style="background-color:red">
```

```
<col style="background-color:yellow">
```

</colgroup>

```
ISBN
```

```
Title
```

```
Price
```

```
3476896
```

```
My first HTML
```



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

\$53

Here in the example the colgroup tag is used to group 3 columns in a table. Inside the colgroup tag the col tag is used to specify the styles for a number of columns. When the attributes are used in the colgroup tag and col tag is avoided then the columns spanned will be having the formatting in the colgroup tag structurally.

The output for the above html coding will be



Dividing table into horizontal sections:

- The previous topic says about how to divide the table into vertical groups or sections, here the table will be grouped into horizontal sections or groups using three tags.
 - **<thead>-**indicates the heading section of table where the contents given will be displayed first in the table. The contents can be given as usual with the tr, td and th tags. To be simple the thead tag groups the header content for the table.
 - **-**tbody indicates the body section of the table where the real values under the thead contents will be given. To be simple the tbody tag groups the body content for the table.
 - <tfoot>-denotes the footer part of the table where the last content will be given for a table. To be simple the tfoot tag groups the footer content for the table.



CLASS: III BSC CSCOURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

- These tags are given in order to apply common styles to the table row wise instead of applying the styles for each cell.
- The common attributes for these tags are Align, Valign and Style.
- **The values for align attribute in above two titles are left, center, right and justify.
- ** The values for valign attribute in above two titles are top, middle, bottom and baseline.

```
(E.G):-
```

```
<thead>
Month
 Savings
</thead>
<tfoot>
Sum
 $180
</tfoot>
January
 $100
February
 $80
```



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

Output



**As explained the contents in thead, tbody and tfoot will be displayed in the appropriate places and also the formatting to the respective rows can be applied in common.

Creating Headers:

- Headers in a table will be placed on top of the text generally. The other properties of the headers in table are
 - \checkmark They are in bold and centered.
- 4 The appropriate use of the header tag \langle th \rangle in proper place in a table gives the table a good look.
- \downarrow The cells defined using tag will be called as HEADER CELLS.
- \downarrow The other cells defined using tag will be called as STANDARD CELLS.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

Month

Savings

January

\$100

The above example will yield the following output.

| C:Wocuments and Settings\SK\Desktop\th.html - Microsoft Internet Explorer | | |
|---|-------|----------|
| Elle Edit Yiew Favorites Iools Help | | . |
| 🛇 Back 👻 🕑 👻 📓 🏠 🔎 Search 👷 Favorites 🚱 🔗 - چ 🖅 - 🛄 🏭 🎎 🦓 | | |
| Address 🖉 C:\Documents and Settings\SK\Desktop\th.html | Go Li | nks " |
| Month (Savings) January \$100 | | |
| | | ~ |
| B | | |

Adding Border:

- Overall a table can be highlighted using a border.
- To provide a border for a table the attribute border can be used in table tag.
- A numerical value can be assigned to the border attribute, based on the numerical value of the border the thickness of the table value varies.
- The attributes related to the colors of table border are



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

- ✓ BORDERCOLOR- applies a single color to the whole table border and the color can be applied through color name, hexadecimal value and rgb component value.
- ✓ BORDERCOLOR-DARK- applies a color to the darker parts of the table border.
- ✓ BORDERCOLOR-LIGHT- applies a color to the lighter parts of the table border.

(E.g):-

```
Month

Savings

Savings

January

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$100

$10
```

The output will be a table with a border thickness of one point.

Putting a Background Image:

- A background image can be added to either to the whole table or individual cells.
- The attribute for displaying a background image in a table or table cell is BACKGROUND and the URL of the image will be assigned to this attribute.
- Here the individual cells may be the standard or header cells.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

(E.G):-

```
<html>
<body>
<table
       background="C:\Documents
                                  Settings\All
                                             Users\Documents\My
                             and
Pictures\Water lilies.jpg" border=5>
<font color="White">
 <font color="White">
ISBN</font>
 <font color="White">Title</font>
 <font color="White">Price</font>
3476896
 My first HTML
 $53
</body>
```

</html>

Output:

Pictures\Sample



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

| C:\Documents and Settings\SK\Desktop\js.html - Microsoft Internet Explorer | | |
|--|-------------|--------------|
| Elle Edit View Favorites Iools Help | | 2 |
| 🕞 Back 👻 🕥 👻 📓 🏠 🔎 Search 🬟 Favorites 🤣 🎯 - 🛬 I | e 🔹 🔜 🕼 🏙 🦀 | |
| Address 🕘 C:\Documents and Settings\SK\Desktop\js.html | ~ | 🔁 Go Links » |
| ISBN Trie Price | | 8 |
| | | ~ |
| 🔊 Done | 😼 My C | Computer |

Heading across two or more columns:

- Sometimes if there is a need to display a single heading over two columns then it's possible by making a single cell to span two or more columns.
- That's made possible by the use of COLSPAN attribute. When it is used with thor td tag then we can span a number of columns assigned to the colspan attribute.
- (E.G):-

```
Month
Savings
January
January
```





CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

February

\$100

Sum: \$180

Output





CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

In the above example the last row content sum will be displayed in one cell spanning the space of two cells. But the contents above this sum will be displayed in separate cells since there is no usage of colspan attribute in them.

Changing color of the cell:

- The color of a cell in table can be changed by the use of bgcolor attribute which is also used in the other elements for background color.
- A cell's background color can be changed by using the above specified attribute in td or th element.
- Like all the other elements the cell color can be changed in three ways. That is by using color name, hexa value and rgb component value.
- (E.G):-

<html>

<body>

```
<font color="White">
```

```
<font color="White">ISBN</font>
```

```
<font color="White">Title</font>
```

```
<font color="White">Price</font>
```

```
<font color="White">3476896</font>
```

```
<font color="White">My first HTML</font>
```

```
<font color="White">$53</font>
```



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

</body>

</html>

• In the above example the background color of the table cell is set to black and the font is set to white. The output for the example program above will be

| C:\Documents and Settings\SK\Desktop\js.html - Microsoft Internet Explorer | | |
|--|--------|----------|
| Elle Edit <u>V</u> iew F <u>e</u> vorites <u>T</u> ools <u>H</u> elp | | . |
| 🕞 Back 🔹 🐑 👻 😰 🏠 🔎 Search 🤺 Favorites 🤣 🔗 - چ 📼 🔹 🧾 🏭 🎎 🦄 | | |
| Address 🕘 C:\Documents and Settings\SK\Desktop\js.html | 🔁 Go | Links » |
| ISBN Title Price 3476896 My first HTIML \$53 | | |
| | | |
| | | |
| | | ~ |
| 街 Done 🤤 😨 My Co | mputer | |

Aligning the contents of cells:

- Alignment generally can be defined as the direction of the object displayed.
- Here in the term of tables the alignment can be said as the direction of the content in the cells.
- That is set using the attribute ALIGN in td or th element.
- The most used alignment values are left, right, center and justify.

Display of tables:

- Tables may sometimes load into the browser very slowly. For increasing the speed of loading there are some tips as follows.
 - **4** Use absolute values in pixels or percentages for cell width.


CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

- Use only proportional width for cells, columns and horizontal sections when the width for table is been already set.
- **4** Table should be divided into column groups.

FRAMES:

- HTML frames allow authors to present documents in multiple views, which may be independent windows or sub windows.
- In other words if the web page developer want to display the contents in multiple sections under one window then this frames are useful.
- Also the frames can be used to navigate between pages without leaving the current page.
- The process of dividing the window into multiple sections involves the use of two tags / elements.
- They are <frameset> and <frame> elements.
- The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.
- The attributes inside the frameset element are cols and rows. Both of them specify the number of columns and rows that a frameset should contain. The values for these attributes will be in pixels, % and *.
- The contents inside divided columns and rows in a frameset are given using the frame element.
- The attributes of frame tag are as follows.
 - The frameborder attribute specifies whether or not to display a border around a frame. The input value for this attribute is numerical.
 - **4** The src attribute specifies the URL of the document to show in a frame.



CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

- The longdesc attribute specifies a URL to a page that contains a long description of the content of a frame. It is not supported today. As an alternative a link can be introduced in the particular frame which takes the user to another oage that contains the long text.
- The marginheight attribute specifies the height between the content and the top and bottom of the frame, in pixels.
- The marginwidth attribute specifies the width between the content and the left and right sides of the frame, in pixels.
- The name attribute is used to reference the element in a JavaScript, or to act as a target for a link.
- The noresize attribute specifies that a <frame> element cannot be resized by the user. By default, each <frame> in a <frameset> can be resized by dragging the border between the frames. However, this attribute locks the size of a frame.
- 4 The scrolling attribute specifies whether or not to display scrollbars in a <frame>.

** The <noframes> element can be used to link to a non-frameset version of the web site or to display a message to users that frames are required.

(E.g):-

<frameset cols="25%,*,25%">

<frame src="frame_a.htm">

<frame src="frame_b.htm">

<frame src="frame_c.htm">

</frameset>

Output:





CLASS: III BSC CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: I(INTRODUCTION TO HTML) BATCH-2015-2018

| Frame A Note: The frameset, frame, and noframes | Frame B | Frame C | |
|--|---------|---------|--|
| elements are not supported in HTML5. | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |





COURSE CODE: 15CSU602 UNIT: I (INTRODUCTION TO HTML) BATCH-2015-2018

POSSIBLE QUESTIONS

(Section B)

5*8=40

1. Give a brief introduction to the following tags.

i)
 ii) iii) <a href> iv)

- 2. Discuss Table Handling in HTML.
- 3. Brief about the components of header section in HTML.
- 4. Write about the attributes to design colorful web page in HTML.
- 5. Write about Link tag in HTML.
- 6. Discuss about image maps in HTML.
- 7. What is meant by columspan and rowspan?
- 8. Discuss about creation of unordered lists in HTML with examples.
- 9. Write about formatting of text in HTML
- 10. Explain the usage of <frame> and <frameset> tags in HTML with examples.



DEPARTMENT OF COMPUTER SCIENCE, CA & IT

III B.Sc CS (Batch 2015-2018)

WEB TECHNOLOGY

PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS

ONLINE EXAMINATIONS UNIT-1

| S.No | Questions | opt1 | opt2 | opt3 | opt4 | Answer |
|------|---|--|--|---|---|-----------------------|
| | | | Hyper Text | Hyperlinks and | Hyper Tool | |
| | | Home Tool | Markup | Text Markup | Mark | Hyper Text Markup |
| 1 | What does HTML stand for? | Markup Language | Language | Language | language | Language |
| | | | | | | |
| | Choose the correct HTML tag for the largest | | | | | |
| 2 | heading | <h6></h6> | <head></head> | <h1></h1> | <heading></heading> | <h1></h1> |
| | | | | | | |
| | Select the HTML tag for inserting a line | | | | | |
| 3 | break? | | <lb></lb> | <break></break> | <brk></brk> | |
| | | | <background>y</background> | <body< th=""><th></th><th></th></body<> | | |
| | What is the HTML tag for adding a | <body< th=""><th>ellow<th>bgcolor="yello</th><th><body=yellow< th=""><th><body< th=""></body<></th></body=yellow<></th></th></body<> | ellow <th>bgcolor="yello</th> <th><body=yellow< th=""><th><body< th=""></body<></th></body=yellow<></th> | bgcolor="yello | <body=yellow< th=""><th><body< th=""></body<></th></body=yellow<> | <body< th=""></body<> |
| 4 | background color? | color="yellow"> | und> | w"> | color> | color="yellow"> |
| | | | | | | |
| | | | | | | |
| 5 | What tag is used to make a text bold | <bld></bld> | <bb></bb> | | <bold></bold> | |
| | | | | | | |
| | | | | | | |
| 6 | Choose the HTML tag to make a text italic | <ii></ii> | <italics></italics> | <i></i> | <it></it> | <i></i> |

| | | <a< th=""><th><a< th=""><th></th><th><mail< th=""><th><a< th=""></a<></th></mail<></th></a<></th></a<> | <a< th=""><th></th><th><mail< th=""><th><a< th=""></a<></th></mail<></th></a<> | | <mail< th=""><th><a< th=""></a<></th></mail<> | <a< th=""></a<> |
|----|--|--|--|--|---|-------------------|
| | | href="mailto:xxx@ | href="xxx@yy | <mail>xxx@y</mail> | href="xxx@y | href="mailto:xxx@ |
| 7 | How can you make an e-mail link? | yyy"> | y"> | yy | yy"> | yyy"> |
| | | | | | | |
| | | <head><tfo< th=""><th><thead><body></body></thead></th><th><td< th=""><th><t< th=""><th></th></t<></th></td<></th></tfo<></head> | <thead><body></body></thead> | <td< th=""><th><t< th=""><th></th></t<></th></td<> | <t< th=""><th></th></t<> | |
| 8 | Which of these tags are all tags? | ot> | | > | t> | |
| | | | | | | |
| | Choose the correct HTML to left-align the | | | <td< th=""><th></th><th></th></td<> | | |
| 9 | content inside a tablecell | | | valign="left"> | <tdleft></tdleft> | |
| | | | | | | |
| | How can you make a list that lists the items | | | | | |
| 10 | with numbers? | | | <dl></dl> | <list></list> | |
| | | | | | | |
| | How can you make a list that lists the items | | | | | |
| 11 | with bullets? | <list></list> | | | <dl></dl> | |
| | HTML tags are surrounded by the two | | | | | |
| | characters | | | | | |
| 12 | | < and > | << and >> | < and | ?? And ?? | < and > |
| | | | | | | |
| | is used to insert a comment in the | | | The comment | | |
| 13 | HTML source code | Head Tag | Title Tag | tag | body tag | The comment tag |
| | is the syntax for the comment tag | | | | | |
| | used in HTML | This is a</td <td>< This is a</td> <td><!-- This is a</td--><td><? This is a</td><td><!-- This is a</td--></td></td></td> | < This is a | This is a</td <td><? This is a</td><td><!-- This is a</td--></td></td> | This is a</td <td><!-- This is a</td--></td> | This is a</td |
| 14 | | comment> | comment> | comment !> | comment ?> | comment> |
| | is the HTML Attribute which defines | | | | | |
| | the body of an HTML document | | | | | |
| 15 | | <head></head> | <body></body> | <title></title> | <bd></bd> | <body></body> |

| | is the HTML Attribute which defines | | | | | |
|----|-------------------------------------|-------------------------|-------------------|-----------------|-------------------|-----------------|
| | an HTML table | | | | | |
| 16 | | | | > | <tb></tb> | |
| | is the tag used to emphasized the | | | | | |
| | text | | | | | |
| 17 | | <emphasize></emphasize> | <emp></emp> | | <emph></emph> | |
| | is the tag used to defines small | | | | | |
| | text | | | | | |
| 18 | | <big></big> | | <small></small> | <low></low> | <small></small> |
| | | | | | | |
| | | | | | | |
| 19 | is the tag used to define a row | | > | | <trow></trow> | |
| Γ | is the tag used define a cell | | | \Box | | |
| | | | | | | |
| 20 | | | | > | <tcell></tcell> | |
| | | | | | | |
| | The attribute is used to define an | | | | | |
| 21 | "alternate text" for an image. | alt | align | src | alter | alt |
| | | | | | | |
| | | | | | | |
| 22 | is the tag used to define a image. | | <image/> | <im></im> | | |
| | | | | | | |
| | is the tag used to defines a | | | | | |
| 23 | clickable area inside an image map | <imgmap></imgmap> | <area/> | <image/> | | <area/> |
| | | | | | | |
| | The attribute specifies a | | | | | |
| 24 | background-color for an HTML page. | backcolor | bgcolor | Bgcolor() | bodycolor | bgcolor |

| | The attribute specifies a background- | | | | | |
|----|---|-------------|------------|------------|-------------|-------------|
| 25 | image for an HTML page. | background | Bground | backimg | bgimg | background |
| | can be defined as a hexadecimal | | | | | |
| | notation for the combination of Red, Green, | | | | | |
| 26 | and Blue color values (RGB). | HTML colors | HTML Table | Html Lists | HTML frames | HTML colors |
| | | | | | | |
| | is the attribute where the linked | | | | | |
| 27 | document will be opened. | target | Source | src | url | target |
| | | | | | | |
| | is the tag used to define the | | | | | |
| 28 | definition list | dl | dt | dm | dlist | dl |
| | | | | | | |
| | is the tag used to define the | | | | | |
| 29 | definition terms | dl | dt | dd | dterms | dt |
| | | | | | | |
| | | | | | | |
| 30 | is the tag used to define the paragraph | р | ра | paragraph | para | р |
| | | | | | | |
| | is the tag used to define horizontal | | | | | |
| 31 | line | hr | hrr | br | brr | hr |
| | | | | | | |
| | | | | | | |
| 32 | defines the largest heading. | <h2></h2> | <h1></h1> | <h6></h6> | <h5></h5> | <h1></h1> |
| | | | | | | |
| | | | | | | |
| 33 | defines the smallest heading. | <h6></h6> | <h1></h1> | <h2></h2> | <h3></h3> | <h6></h6> |

| 34 | Defines the document's body | <body></body> | <bg></bg> | <bodydoc></bodydoc> | <body statement></body | <body></body> |
|----|--|-----------------|--------------|---------------------|----------------------------------|-----------------|
| | | Standard | Standard | General | some Geenral | Standard |
| | | Generalized | General Mark | Markup | Mark | Generalized |
| 35 | SGML Stands for | Markup Language | Up Languae | Languae | Lanaguage | Markup Language |
| | is the attribute of the body tag | | | | | |
| | which can be used to specify the url of an | | | | | |
| 36 | image to tile as the document background. | <style></style> | | | | |

| 43 | Tag for using graphic is IMG followed by | SRC | RCS | CRS | SCR | SRC |
|----|--|--------------|----------------|-----------------|----------------|----------------|
| 44 | HTML definition is given in the file as | Last line | First line | Not given | Optional | First Line |
| | | | | | optional | |
| 15 | Colors can be given to fonts using | bacolor | olr | color | backgroundco | color |
| 45 | | Ugeoloi | | 0101 | 101 | |
| 10 | | | | 1. C | | 1' 1'(|
| 40 | which is not a type of list in html | ordered list | unordered list | definition list | graphics list | graphics list |
| | | | | | | |
| 47 | HTML file is saved with extension | .ht | .html | .txt | .doc | .html |
| | The output of a HTML file can be viewed | | | | | |
| 48 | using | document | slide show | browser | interpreter | browser |
| | How many spaces should be provided in | | | | | |
| 10 | writing a HTML tag | | 2 | 1 - | No restriction | No restriction |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

UNIT-II

Syllabus

Forms: Working with forms-Creating forms-working with menus- working with Radio buttonscheck boxes-textboxes-text areas- password boxes-submit button-Resetting the form. DHTML: Heading and Horizontal line-Hidden Message-Message at the center of the page- Moving Boxes-Changeable Box- CSS: Introduction- Creating Style Sheets-Common Tasks with CSS- Colors-The Font Family.

FORMS:

- \checkmark Concept that is used to make the website or webpage to be interactive is called as forms.
- ✓ The main idea here is to get the input data from the users of the website/page and respond with a result.
- ✓ The controls that are used to get input may be textboxes, password boxes, checkboxes, radio buttons etc.

Creating a Form:

- ✓ Basically the form tag/element contains 2 attributes, they are action and method.
- ✓ The action attribute specifies the url of the CGI script in the server. The input data given are processed using this CGI script.
- ✓ This Common Gateway Interface (CGI) is a common method used to generate dynamic content in the web pages. This CGI when implemented provides an interface between the web page that generates contents and the web server.
- \checkmark The method attribute has two values they are GET and POST.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

- GET- in get method the form data is appended into the URL. The length of the URL is limited to 3000 characters. This method is not suitable for sensitive data. When the result page of form submitted is to be stored using bookmarks then it is possible in this method. Query strings are good example for this GET.
- POST- appends form data inside the body of HTTP request and it's not shown in the URL. There is no size limitations for the data submitted in this method. These kind of pages cannot be bookmarked. The new email id creation is a good example for this POST method.

(E.G):-

<html>

<body>

<form action="demo_form.asp">

First name: <input type="text" name="FirstName" value="Mickey">

Last name: <input type="text" name="LastName" value="Mouse">

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

</form>

Click the "Submit" button and the form-data will be sent to a page on the server called "demo_form.asp".

</body>

</html>

** In the Below example the default form method is get where the data is carried using the URL value. The page where the CGI script runs is also specified in the action attribute. Two text boxes with a



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

submit button is displayed. When the submit button is clicked the data is posted into the URL and it's processed in the action URL.

Output:

| First name: Mickey Last name: Mouse Submit | |
|--|--|
| Click the "Submit" button and the form-data will be sent to a page on the server called "demo_form.asp". | |
| | |
| | |
| | |
| | |

Working with menus:

- Menus in html are used to display a list of items in a webpage. The menu can be otherwise called as drop down list.
- ↓ The <select> element is used to create a drop-down list.
- **4** Mainly used to display list of items in commercial sites, job sites etc.
- **4** The attributes under this select are as follows
 - ✓ Disabled-The disabled attribute is a Boolean attribute. When present, it specifies that the drop-down list should be disabled. A disabled drop-down list is unusable and un-clickable.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

The disabled attribute can be set to keep a user from using the drop-down list until some other condition has been met (like selecting a checkbox, etc.). Then, a JavaScript can remove the disabled value, and make the drop-down list usable.

- ✓ Multiple-The multiple attribute is a boolean attribute. When present, it specifies that multiple options can be selected at once. Selecting multiple options vary in different operating systems and browsers:
 - > For windows: Hold down the control (ctrl) button to select multiple options
 - > For Mac: Hold down the command button to select multiple options
- ✓ Size-The size attribute specifies the number of visible options in a drop-down list. If the value of the size attribute is greater than 1, but lower than the total number of options in the list, the browser will add a scroll bar to indicate that there are more options to view.
- ✓ Name- The name attribute specifies the name for a drop-down list. The name attribute is used to reference elements in a JavaScript, or to reference form data after a form is submitted.
- **4** The items for a menu can be displayed using the OPTION tag. The attributes for option tag are
 - ✓ Disabled- The disabled attribute is a boolean attribute. When present, it specifies that an option should be disabled. A disabled option is unusable and un-clickable.
 - ✓ Label- The label attribute specifies a shorter version of an option. The shorter version will be displayed in the drop-down list.
 - ✓ Selected-The selected attribute is a boolean attribute. When present, it specifies that an option should be pre-selected when the page loads. The pre-selected option will be displayed first in the drop-down list.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

✓ Value- The value attribute specifies the value to be sent to a server when a form is submitted. The content between the opening <option> and closing </option> tags is what the browsers will display in a drop-down list. However, the value of the value attribute is what will be sent to the server when a form is submitted.

<select>

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="mercedes">Mercedes</option>

<option value="audi">Audi</option>

</select>

**displays a drop down list with the names of various cars.

Working with Radio Buttons:

- As an alternate to menus the radio buttons can be used to display options to the user and he/she can select only one option among them.
- ✓ The tag/element that is used for this purpose is input tag where the type should be given as radio in order to display the radio button.
- ✓ The other attributes in this input tag are name-specifies the name of the radio button for programming usage and value-specifies the value that should be returned to the server when the form is submitted.
- \checkmark Another one attribute is disables which denotes that whether the radio button is clickable or not.

(E.G):-

<form action="/cgi-bin/radiobutton.cgi" method="post">

<input type="radio" name="subject" value="maths" /> Maths

<input type="radio" name="subject" value="physics" /> Physics

<input type="submit" value="Select Subject" />



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

</form>

**displays a form with two radio buttons maths and physics out of which one can be selected.

Output:



Check Boxes:

✓ For the selection of multiple options these check boxes are used. The implementation details are same as the radio buttons with only one change. That is the type value will be checkbox.

(E.G):-

```
<form action="/cgi-bin/checkbox.cgi" method="post">
```

```
<input type="checkbox" name="subject" value="maths" /> Maths
```

<input type="checkbox" name="subject" value="physics" /> Physics

<input type="submit" value="Select Subject" />

</form>

Output:



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

| Elle Edle Yiew Favorites Favorites Back Image: Second Section (Section | | |
|--|------------|----------|
| Back Image: Second Section Sectin Section Section Section Sectin Section Section Section | | |
| Address 🛃 C:\Documents and Settings\SK\Desktop\test.html | [2] [2] 48 | |
| Maths Physics Select Subject | 🔽 🄁 Go Li | nks » |
| | | |
| | | |
| | | <u> </u> |

Text Box:

- \checkmark Used to get text input from the user.
- ✓ The tag used here is input tag. The type value will be "text" in order to display a text box. And a name will be given to it.
- ✓ (E.G):- <input type="text" name="ftext"/>

Text areas:

- \checkmark The <textarea> tag defines a multi-line text input control.
- ✓ A text area can hold an unlimited number of characters, and the text renders in a fixed-width font (usually Courier).
- \checkmark The size of a text area can be specified by the cols and rows attributes.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

- ✓ The other attributes related to this text area tag are name and readonly- The readonly attribute is a boolean attribute. When present, it specifies that a text area should be read-only. In a read-only text area, the content cannot be changed, but a user can tab to it, highlight it and copy content from it.
- ✓ (E.G):-

<textarea rows="4" cols="50">

At Karpagam University various courses are offered.

</textarea>

Password Boxes:

- ✓ If the input given by the user is to be hidden to his/her view then its possible through the password box.
- \checkmark It is also displayed using the input tag with 2 changes.
 - > The type of the input element is password.
 - > The size is also mentioned for password limitation.

(E.G):- <input type="Password" name="pwdbox" size="10"/>

Submit Button:

- ✓ If the form data is entered in a web page is ready for processing then it should be submitted to the server. To initiate that action the submit button is used in html.
- ✓ This is also displayed using the input tag where the type is given as submit. A value should be given inorder to display a text in the button face.

(E.g):-



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

<input type="Submit" value="Submit"/>

Resetting the form:

- ✓ While entering the data into a form in a web page if the user makes a mistake and finds it then he/she will wish to start with a new and fresh form.
- \checkmark That is done with the help of reset button in the html.
- \checkmark That is also displayed using the input tag but the type value will be reset.

(E.G):-

<input type="Reset" value="Reset"/>

When the user clicks the reset button the web page just reloads giving the user again a new and fresh page.

** For submitting and resetting the <button> tag can also be used in HTML 4.0.

DHTML:

Dynamic HTML, or DHTML, is an <u>umbrella term</u> for a collection of technologies used together to create interactive and animated <u>web sites</u> by using a combination of a static <u>markup language</u> (such as <u>HTML</u>), a <u>client-side scripting</u> language (such as <u>JavaScript</u>), a presentation definition language (such as <u>CSS</u>), and the <u>Document Object Model</u>.

DHTML allows scripting languages to change <u>variables</u> in a web page's definition language, which in turn affects the look and function of otherwise "static" HTML page content, after the page has been fully loaded and during the viewing process. Thus the dynamic characteristic of DHTML is the way it functions while a page is viewed, not in its ability to generate a unique page with each page load.

Prepared By K.Kathirvel & K.Yuvaraj, Department Of Computer Science, CA & IT, KAHE



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

By contrast, a <u>dynamic web page</u> is a broader concept, covering any web page generated differently for each user, load occurrence, or specific variable values. This includes pages created by client-side scripting, and ones created by <u>server-side scripting</u> (such as <u>PHP</u>, <u>Perl</u>, <u>JSP</u> or <u>ASP.NET</u>) where the web server generates content before sending it to the client.

Heading and Horizontal Line:

The <hr> tag defines a thematic break in an HTML page. The <hr> element is used to separate content (or define a change) in an HTML page.

In HTML5, the <hr> tag defines a thematic break. In HTML 4.01, the <hr> tag represents a horizontal rule.

However, the <hr> tag may still be displayed as a horizontal rule in visual browsers, but is now defined in semantic terms, rather than presentational terms. All the layout attributes are removed in HTML5. Use CSS instead.

An example for the Horizontal Rule is given below.

| DQCIYPE html <html> <body></body></html> | HTML |
|--|--|
| <h1>HTML</h1> HTML is a language for describing web pages. | HTML is a language for describing web pages. |
| <hr/> <hl><\$\$\$</hl> \$\$\$ defines how to display HTML elements. | CSS |
| | CSS defines how to display HTML elements. |
| | |
| | |
| | |
| | |
| | |
| | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

Add some text for the TITLE tag. Add an H1 heading, with some text between the two H1 tags. Now add a paragraph of text below the H1 heading. To create an Inline Style, click after the first H1. Tap the space bar on your keyboard. Type the following:

<H1 style=text-align:center>

Make sure there is no space after the colon and before the word 'center'.

Your HTML code should look something like ours below:

| newpage.html - Notepad | | |
|--|-----|--|
| File Edit Format View Help | | |
| HTML | ~ | |
| <html> <head> <title>Styles</title></head></html> | | |
| | | |
| <body> <h1 style="text-align:center">Centred Heading</h1></body> | | |
| <p>Style sheets can be used to format text on a page.</p> | web | |
| | | |
| | * | |

(One thing British English users should note is the two different spelling of the word 'centre'. American English is used in HTML, so it's 'er' rather than 're'.)

When you have finished typing the code, view the results in your browser. You should see that the heading is centered on your page:



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018



When a hr tag is used after this aligned heading then the output will be as follows.



HIDDEN MESSAGE:

Prepared By K.Kathirvel & K.Yuvaraj, Department Of Computer Science, CA & IT, KAHE



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

To display a hidden message in a Web page then the visibility property can be set to hidden in style attribute. This can be done in any element available in html. Later that hidden message can be retrieved using a java script coding with the help of that element's name.

```
(E.G):-
```

<!DOCTYPE html>

<html>

<body>

```
This is a p element.
```

<button type="button" onclick="myFunction()">Hide content of p</button>

<script>

```
function myFunction()
```

```
{
```

```
document.getElementById("myP").style.visibility="hidden";
```

```
}
```

</script>

</body>

</html>

The above coding yields the following output.

Reparts

This is a p element.

Hide content of p

Prepa



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

When the button is clicked the text will be disappearing above.

MOVING BOXES:

To create an effect of moving boxes in a web page we have to create a set of boxes using div tag and then change their positions accordingly. Such that the end position of one box will be the starting position of another box.

Apart from that the size and color of all the boxes should be same to create such an effect.

(E.G):-

<div id="d1" style="position:Absolute;left:150;top:180;width:120;height:60"></div>

<div id="d1" style="position:Absolute;left:270;top:180;width:120;height:60"></div>

Here by the above coding two boxes will be created but they will be looking like moving.

CHANGEABLE BOX:

Changeable box is just a gimmicks used to create a feeling to the user that he/she views that different boxes are interchangeably appearing in a web page.

That is done by creating two div sections in a web page with same contents but different color, size properties and using a java script the visibility of both the boxes are changed alternatively when the mouse is hovered over the boxes.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

<u>CSS</u>

- CSS is the acronym for: 'Cascading Style Sheets'. CSS is an extension to basic HTML that allows you to style your web pages.
- An example of a style change would be to make words bold. In standard HTML you would use the tag like so:

<**b**>make me bold</**b**>

**This works fine and there is nothing wrong with it per se, except that now if you wanted to say change all your text that you initially made bold to underlined, you would have to go to every spot in the page and change the tag.

**Another disadvantage can be found in this example: say you wanted to make the above text bold, make the font style Verdana and change its color to red; you would need a lot of code wrapped around the text:

Prepared By K.Kathirvel & K.Yuvaraj, Department Of Computer Science, CA & IT, KAHE



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

This is text This is verbose and contributes to making your HTML messy. With CSS, you can create a custom style elsewhere and set all its properties, give it a unique name and then 'tag' your HTML to apply these stylistic properties:

My CSS styled text

And in between the tags at the top of your web page you would insert this CSS code that defines the style we just applied:

```
<style type="text/css">
.myNewStyle {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-weight: bold;
color: #FF0000;
```

```
}
```

</style>

In the above example we **embed the css code** directly into the page itself. This is fine for smaller projects or in situations where the styles you're defining will only be used in a single page. There are many times when you will be applying your styles to many pages and it would be a hassle to have to copy and paste your CSS code into each page.

Besides the fact that you will be cluttering up your pages with the same CSS code, you also find yourself having to edit each of these pages if you want to make a style change. Like with JavaScript, you can define/create your CSS styles in a separate file and then link it to the page you want to apply the code to:

```
k href="myFirstStyleSheet.css" rel="stylesheet" type="text/css">
```



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

The above line of code links your external style sheet called 'myFirstStyleSheet.css' to the HTML document. You place this code in between the <head> </head> tags in your web page.

HOW TO CREATE AND LINK A STYLESHEET:

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing just one file.

Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:

<head>

k rel="stylesheet" type="text/css" href="mystyle.css"> </head>

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file is shown below:

"myStyle.css":



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

hr{color:sienna;}
p{margin-left:20px;}
body {background-image: url("images/background.gif");}

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the <style> tag, like this:

<head> <style> hr{color:sienna;} p{margin-left:20px;} body{background-image:url("images/background.gif");} </style> </head>

Inline Styles

An inline style loses many of the advantages of a style sheet (by mixing content with presentation).

To use inline styles, add the style attribute to the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

This is a paragraph.

Multiple Style Sheets



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, assume that an external style sheet has the following properties for the h3 selector:

h3{

color:red;

text-align:left;

font-size:8pt;

}

Then, assume that an internal style sheet also has the following properties for the h3 selector:

h3{

```
text-align:right;
font-size:20pt;
```

}

If the page with the internal style sheet also links to the external style sheet the properties for the h3 element will be:

color:red;

text-align:right;

font-size: 20pt;

The color is inherited from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.

Multiple Styles Will Cascade into One

Prepared By K.Kathirvel & K.Yuvaraj, Department Of Computer Science, CA & IT, KAHE



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

Styles can be specified:

- inside an HTML element
- inside the head section of an HTML page
- in an external CSS file

Cascading order

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

- 1. Browser default
- 2. External style sheet
- 3. Internal style sheet (in the head section)
- 4. Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).

COMMON TASKS WITH CSS:

Fonts:

The common tasks done on text displayed in a web page is explained with an example below.

<!DOCTYPE html>

<html>

<head>

<style>

p.ex1 {

font: 15px arial, sans-serif;



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

}

p.ex2 {

font:italic bold 12px/30px Georgia, serif;

}

</style>

</head>

<body>

This is a paragraph. This paragraph. T

This is a paragraph. This paragraph. T

</body>

</html>

In the above example two types of text formatting are used for the paragraph element with different class names. When those class names are used with two types of paragraph elements then we can see the exact change as below.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

This is a paragraph. This is a paragraph.

This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph.

Margins:

Using css we can also edit the margins of several elements as by the following example.

<!DOCTYPE html>

<html>

<head>

<style>

p {

background-color: yellow;

}

p.ex {

margin-top: 100px; margin-bottom: 100px;

margin-right: 150px;

margin-left: 50px;

}



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

|--|

</head>

<body>

This is a paragraph with no specified margins.

This is a paragraph with specified margins.

</body>

</html>

| This is a paragraph with no specified margins. | |
|--|--|
| | |
| | |
| | |
| This is a paragraph with specified margins. | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

The margin property can have from one to four values.

4 margin: 25px 50px 75px 100px;

- ✓ top margin is 25px
- ✓ right margin is 50px
- ✓ bottom margin is 75px
- ✓ left margin is 100px

4 margin: 25px 50px 75px;



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

- ✓ top margin is 25px
- ✓ right and left margins are 50px
- ✓ bottom margin is 75px

margin: 25px 50px;

- \checkmark top and bottom margins are 25px
- \checkmark right and left margins are 50px

✓ all four margins are 25px

Links:

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

In addition, links can be styled differently depending on what **state** they are in.

The four links states are:

- a:link a normal, unvisited link
- a:visited a link the user has visited
- a:hover a link when the user mouses over it
- a:active a link the moment it is clicked

(E.G):-

<!DOCTYPE html>

<html>

<head>

⁴ margin: 25px;

Lote (Lington) (Leve Case) (Lington) (Leve CASE) (Lington) (Lingto

KARPAGAM ACADEMY OF HIGHER EDUCATION

CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

```
<style>
/* unvisited link */
a:link {
  color: #FF0000;
}
/* visited link */
a:visited {
  color: #00FF00;
}
/* mouse over link */
a:hover {
  color: #FF00FF;
}
/* selected link */
a:active {
  color: #0000FF;
}
</style>
</head>
<body>
<b><a href="default.asp" target="_blank">This is a link</a></b>
<b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be
effective.
<b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be
```

effective.</body>

Е

| This is a link |
|---|
| Note: a hover MUST come after a link and a visited in the CSS definition in order to be effective. |
| Note: a active MUST come after a hover in the CSS definition in order to be effective. |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

</html>

COLORS:

Border Color

The border-color property is used to set the color of the border. The color can be set by:

- name specify a color name, like "red"
- RGB specify a RGB value, like "rgb(255,0,0)"
- Hex specify a hex value, like "#ff0000"

You can also set the border color to "transparent".

<!DOCTYPE html>

<html>

<head>




<style>

CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

```
p.one {
    border-style: solid;
    border-color: red;
}
p.two {
    border-style: solid;
    border-color: #98bf21;
}
</style>
</head>
<body>
A solid red border
A solid red border
A solid green border
>>Note:</b> The "border-color" property does not work if it is used alone. Use the "border-style"
property to set the borders first.
```

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

```
Result:
```

A solid red border

```
A solid green border
```

Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

Text Color

The color property is used to set the color of the text.

With CSS, a color is most often specified by:

- a HEX value like "#ff0000"
- an RGB value like "rgb(255,0,0)"
- a color name like "red"

The default color for a page is defined in the body selector.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  color: red;
}
h1 {
  color: #00ff00:
}
p.ex {
  color: rgb(0,0,255);
}
</style>
</head>
<body>
```

<h1>This is heading 1</h1>

This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.

This is a paragraph with class="ex". This text is blue.

</body>



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

</html>

This is heading 1

This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.

This is a paragraph with class="ex". This text is blue.

CSS Background

CSS background properties are used to define the background effects of an element.

CSS properties used for background effects:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Background Color

The background-color property specifies the background color of an element.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

The background color of a page is defined in the body selector:

body{

```
background-color:#b0c4de;
```

}

With CSS, a color is most often specified by:

- a HEX value like "#ff0000"
- an RGB value like "rgb(255,0,0)"
- a color name like "red"

Example

h1{

```
background-color:#6495ed;
```

}

```
p{
```

background-color:#e0ffff;

}

```
div{
```

background-color:#b0c4de;

}

Background Image

The background-image property specifies an image to use as the background of an element.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

Example

body{

```
background-image:url("paper.gif");
```

```
}
```

Background Image - Repeat Horizontally or Vertically

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

Example

body{

```
background-image:url("gradient.png");
```

}

If the image is repeated only horizontally (repeat-x), the background will look better:

Example

body{

```
background-image:url("gradient.png");
```

background-repeat:repeat-x;

}



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

Background Image - Set position and no-repeat

Showing the image only once is specified by the background-repeat property:

Example

body{

```
background-image:url("img_tree.png");
```

```
background-repeat:no-repeat;
```

}

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

The position of the image is specified by the background-position property:

Example

body{
background-image:url("img_tree.png");
background-repeat:no-repeat;
background-position:righttop;
}

Background - Shorthand property



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

As you can see from the examples above, there are many properties to consider when dealing with backgrounds.

To shorten the code, it is also possible to specify all the properties in one single property. This is called a shorthand property.

The shorthand property for background is simply "background":

Example

body{

background:#ffffffurl("img_tree.png")no-repeatrighttop;

}

THE FONT FAMILY:

Introduction

Setting font properties will be among the most common uses of style sheets. Unfortunately, there exists no well-defined and universally accepted taxonomy for classifying fonts, and terms that apply to one font family may not be appropriate for others. E.g., 'italic' is commonly used to label slanted text, but slanted text may also be labeled as being Oblique, Slanted, Incline, Cursive or Kursiv. Therefore it is not a simple problem to map typical font selection properties to a specific font.

Font family: the 'font-family' property

'font-family'

Value:

[[<family-name> | <generic-family>] [, <family-name>| <generic-family>]*



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

] | inherit

Initial: depends on user agent

Applies to: all elements

Inherited: yes

Percentages: N/A

Media: visual

Computed value: as specified

The property value is a prioritized list of font family names and/or <u>generic family names</u>. Unlike most other CSS properties, component values are separated by a comma to indicate that they are alternatives:

body { font-family: Gill, Helvetica, sans-serif }

Although many fonts provide the "missing character" glyph, typically an open box, as its name implies this should not be considered a match for characters that cannot be found in the font. (It should, however, be considered a match for U+FFFD, the "missing character" character's code point).

There are two types of font family names:

<family-name>

The name of a font family of choice. In the last example, "Gill" and "Helvetica" are font families.

<generic-family>



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

In the example above, the last value is a generic family name. The following generic families are defined:

- 'serif' (e.g., Times)
- 'sans-serif' (e.g., Helvetica)
- 'cursive' (e.g., Zapf-Chancery)
- 'fantasy' (e.g., Western)
- 'monospace' (e.g., Courier)

Style sheet designers are encouraged to offer a generic font family as a last alternative. Generic font family names are keywords and must NOT be quoted.

Font family names must either be given quoted as <u>strings</u>, or unquoted as a sequence of one or more <u>identifiers</u>. This means most punctuation characters and digits at the start of each token must be escaped in unquoted font family names.

For example, the following declarations are invalid:

font-family: Red/Black, sans-serif; font-family: "Lucida" Grande, sans-serif; font-family: Ahem!, sans-serif; font-family: test@foo, sans-serif; font-family: #POUND, sans-serif; font-family: Hawaii 5-0, sans-serif;

If a sequence of identifiers is given as a font family name, the computed value is the name converted to a string by joining all the identifiers in the sequence by single spaces.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

To avoid mistakes in escaping, it is recommended to quote font family names that contain white space, digits, or punctuation characters other than hyphens:

body { font-family: "New Century Schoolbook", serif }

<BODY STYLE="font-family: '21st Century', fantasy">

Font family names that happen to be the same as a keyword value ('inherit', 'serif', 'sans-serif', 'monospace', 'fantasy', and 'cursive') must be quoted to prevent confusion with the keywords with the same names. The keywords 'initial' and 'default' are reserved for future use and must also be quoted when used as font names. UAs must not consider these keywords as matching the '<family-name>' type.

Generic font families

Generic font families are a fallback mechanism, a means of preserving some of the style sheet author's intent in the worst case when none of the specified fonts can be selected. For optimum typographic control, particular named fonts should be used in style sheets.

All five generic font families are defined to exist in all CSS implementations (they need not necessarily map to five distinct actual fonts). User agents should provide reasonable default choices for the generic font families, which express the characteristics of each family as well as possible within the limits allowed by the underlying technology. User agents are encouraged to allow users to select alternative choices for the generic fonts.

serif

Glyphs of serif fonts, as the term is used in CSS, tend to have finishing strokes, flared or tapering ends, or have actual serifed endings (including slab serifs). Serif fonts are typically



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

proportionately-spaced. They often display a greater variation between thick and thin strokes than fonts from the 'sans-serif' generic font family.

CSS uses the term 'serif' to apply to a font for any script, although other names may be more familiar for particular scripts, such as Mincho (Japanese), Sung or Song (Chinese), Totum or Kodig (Korean). Any font that is so described may be used to represent the generic 'serif' family.

Examples of fonts that fit this description include:

| Latin fonts | Times New Roman, Bodoni, Garamond, Minion Web, ITC Stone Serif, MS Georgia, Bitstream Cyberbit |
|-------------------|---|
| Greek fonts | Bitstream Cyberbit |
| Cyrillic fonts | Adobe Minion Cyrillic, Excelsior Cyrillic Upright, Monotype Albion 70, Bitstream Cyberbit, ER Bukinist |
| Hebrew fonts | New Peninim, Raanana, Bitstream Cyberbit |
| Japanese fonts | Ryumin Light-KL, Kyokasho ICA, Futo Min A101 |
| Arabic fonts | Bitstream Cyberbit |
| Cherokee fonts | Lo Cicero Cherokee |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

sans-serif

Glyphs in sans-serif fonts, as the term is used in CSS, tend to have stroke endings that are plain -- with little or no flaring, cross stroke, or other ornamentation. Sans-serif fonts are typically proportionately-spaced. They often have little variation between thick and thin strokes, compared to fonts from the 'serif' family. CSS uses the term 'sans-serif' to apply to a font for any script, although other names may be more familiar for particular scripts, such as Gothic (Japanese), Kai (Chinese), or Pathang (Korean). Any font that is so described may be used to represent the generic 'sans-serif' family.

Examples of fonts that fit this description include:

| Latin fonts | MS Trebuchet, ITC Avant Garde Gothic, MS Arial, MS Verdana, Univers, Futura, ITC |
|-------------|--|
| | Stone Sans, Gill Sans, Akzidenz Grotesk, Helvetica |
| Greek fonts | Attika, Typiko New Era, MS Tahoma, Monotype Gill Sans 571, Helvetica Greek |
| Cyrillic | Helvetice Cyrillie, EP, Univers, Lucida Sans Unicode, Pastion |
| fonts | nervenca Cyrinic, EK Onivers, Euclida Sans Onicode, Bastion |
| Hebrew | Arial Hebrew MS Tahoma |
| fonts | |
| Japanese | Shin Go, Heisei Kaku Gothic W5 |
| fonts | Shin Go, Heiser Kaku Gothie W.J |
| Arabic | MS Tahoma |
| fonts | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

cursive

Glyphs in cursive fonts, as the term is used in CSS, generally have either joining strokes or other cursive characteristics beyond those of italic typefaces. The glyphs are partially or completely connected, and the result looks more like handwritten pen or brush writing than printed letterwork. Fonts for some scripts, such as Arabic, are almost always cursive. CSS uses the term 'cursive' to apply to a font for any script, although other names such as Chancery, Brush, Swing and Script are also used in font names.

Examples of fonts that fit this description include:

| Latin fonts | Caflisch Script, Adobe Poetica, Sanvito, Ex Ponto, Snell Roundhand, Zapf-Chancery |
|----------------|---|
| Cyrillic fonts | ER Architekt |
| Hebrew fonts | Corsiva |
| Arabic fonts | DecoType Naskh, Monotype Urdu 507 |

Font boldness: the 'font-weight' property

'font-weight'

| Value: | normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit |
|----------|---|
| Initial: | normal |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

| Applies to: | all elements |
|-----------------|--------------|
| Inherited: | yes |
| Percentages: | N/A |
| Media: | visual |
| Computed value: | see text |

The 'font-weight' property selects the weight of the font. The values '100' to '900' form an ordered sequence, where each number indicates a weight that is at least as dark as its predecessor. The keyword 'normal' is synonymous with '400', and 'bold' is synonymous with '700'. Keywords other than 'normal' and 'bold' have been shown to be often confused with font names and a numerical scale was therefore chosen for the 9-value list.

p { font-weight: normal } /* 400 */
h1 { font-weight: 700 } /* bold */

The 'bolder' and 'lighter' values select font weights that are relative to the weight inherited from the parent:

strong { font-weight: bolder }

The following two examples show typical mappings.

Assume four weights in the "Rattlesnake" family, from lightest to darkest: Regular, Medium, Bold, Heavy.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

| First example of font-weight mapping | | |
|--------------------------------------|-------------|-------------------|
| Available faces | Assignments | Filling the holes |
| "Rattlesnake Regular" | 400 | 100, 200, 300 |
| "Rattlesnake Medium" | 500 | |
| "Rattlesnake Bold" | 700 | 600 |
| "Rattlesnake Heavy" | 800 | 900 |

Assume six weights in the "Ice Prawn" family: Book, Medium, Bold, Heavy, Black, ExtraBlack. Note that in this instance the user agent has decided not to assign a numeric value to "Ice Prawn ExtraBlack".

| Second example of font-weight mapping | | |
|---------------------------------------|-------------|-------------------|
| Available faces | Assignments | Filling the holes |
| "Ice Prawn Book" | 400 | 100, 200, 300 |
| "Ice Prawn Medium" | 500 | |
| "Ice Prawn Bold" | 700 | 600 |
| "Ice Prawn Heavy" | 800 | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

| Second example of font-weight mapping | | |
|---------------------------------------|-------------|-------------------|
| Available faces | Assignments | Filling the holes |
| "Ice Prawn Black" | 900 | |
| "Ice Prawn ExtraBlack" | (none) | |

Values of 'bolder' and 'lighter' indicate values relative to the weight of the parent element. Based on the inherited weight value, the weight used is calculated using the chart below. Child elements inherit the calculated weight, not a value of 'bolder' or 'lighter'.

Font size: the 'font-size' property

'font-size'

| Value: | <absolute-size> <relative-size> <length> <percentage> inherit</percentage></length></relative-size></absolute-size> |
|--------------|---|
| Initial: | medium |
| Applies to: | all elements |
| Inherited: | yes |
| Percentages: | refer to inherited font size |
| Media: | visual |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

| Computed value: abs | solute length |
|---------------------|---------------|
|---------------------|---------------|

The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Values have the following meanings:

<absolute-size>

An <absolute-size> keyword is an index to a table of font sizes computed and kept by the UA. Possible values are:

[xx-small | x-small | small | medium | large | x-large | xx-large]

<relative-size>

A <relative-size> keyword is interpreted relative to the table of font sizes and the font size of the parent element. Possible values are: [larger | smaller]. For example, if the parent element has a font size of 'medium', a value of 'larger' will make the font size of the current element be 'large'. If the parent element's size is not close to a table entry, the UA is free to interpolate between table entries or round off to the closest one. The UA may have to extrapolate table values if the numerical value goes beyond the keywords.

Length and percentage values should not take the font size table into account when calculating the font size of the element.

Negative values are not allowed.

On all other properties, 'em' and 'ex' length values refer to the computed font size of the current element. On the 'font-size' property, these length units refer to the computed font size of the parent element.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

Note that an application may reinterpret an explicit size, depending on the context. E.g., inside a VR scene a font may get a different size because of perspective distortion.

Examples:

Shorthand font property: the 'font' property

'font'

| Value: | [[<'font-style'> <'font-variant'> <'font-weight'>]? <'font-size'> [/ <'line- |
|--------------|--|
| | height'>]? <'font-family'>] caption icon menu message-box small- |
| | caption status-bar inherit |
| | |
| Initial: | see individual properties |
| | |
| Applies to: | all elements |
| | |
| Inherited: | yes |
| | |
| Percentages: | see individual properties |
| | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

| Media: | visual |
|-----------------|---------------------------|
| Computed value: | see individual properties |

The 'font' property is, except as described <u>below</u>, a shorthand property for setting 'font-style', 'fontvariant', 'font-weight', 'font-size', 'line-height' and 'font-family' at the same place in the style sheet. The syntax of this property is based on a traditional typographical shorthand notation to set multiple properties related to fonts.

All font-related properties are first reset to their initial values, including those listed in the preceding paragraph. Then, those properties that are given explicit values in the 'font' shorthand are set to those values. For a definition of allowed and initial values, see the previously defined properties.

- p { font: 12px/14px sans-serif }
- p { font: 80% sans-serif }
- p { font: x-large/110% "New Century Schoolbook", serif }
- p { font: bold italic large Palatino, serif }
- p { font: normal small-caps 120%/120% fantasy }

In the second rule, the font size percentage value ('80%') refers to the font size of the parent element. In the third rule, the line height percentage refers to the font size of the element itself.

In the first three rules above, the 'font-style', 'font-variant' and 'font-weight' are not explicitly mentioned, which means they are all three set to their initial value ('normal'). The fourth rule sets the 'font-weight' to 'bold', the 'font-style' to 'italic' and implicitly sets 'font-variant' to 'normal'.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

The fifth rule sets the 'font-variant' ('small-caps'), the 'font-size' (120% of the parent's font), the 'line-height' (120% times the font size) and the 'font-family' ('fantasy'). It follows that the keyword 'normal' applies to the two remaining properties: 'font-style' and 'font-weight'.

System fonts may only be set as a whole; that is, the font family, size, weight, style, etc. are all set at the same time. These values may then be altered individually if desired. If no font with the indicated characteristics exists on a given platform, the user agent should either intelligently substitute (e.g., a smaller version of the 'caption' font might be used for the 'small-caption' font), or substitute a user agent default font. As for regular fonts, if, for a system font, any of the individual properties are not part of the operating system's available user preferences, those properties should be set to their initial values.

That is why this property is "almost" a shorthand property: system fonts can only be specified with this property, not with 'font-family' itself, so 'font' allows authors to do more than the sum of its subproperties. However, the individual properties such as 'font-weight' are still given values taken from the system font, which can be independently varied.

button { font: 300 italic 1.3em/1.7em "FB Armada", sans-serif }
button p { font: menu }
button p em { font-weight: bolder }

If the font used for dropdown menus on a particular system happened to be, for example, 9-point Charcoal, with a weight of 600, then P elements that were descendants of BUTTON would be displayed as if this rule were in effect:

button p { font: 600 9px Charcoal }

Because the 'font' shorthand property resets any property not explicitly given a value to its initial value, this has the same effect as this declaration:

button p {



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: II (FORMS, DHTML & CSS) BATCH-2015-2018

font-family: Charcoal; font-style: normal;

font-variant: normal;

font-weight: 600;

font-size: 9px;

line-height: normal; }

POSSIBLE QUESTIONS

(Section B)

(5*8=40)

- 1. List out the fields which can be created with <input> tag and Explain.
- 2. Write a note on colors & fonts in CSS.
- 3. Explain creation of Hidden and Text area elements in DHTML.
- 4. Explain how you will design a style sheet.
- 5. Discuss in HTML about elements to get input from the users in form.
- 6. Discuss on conditional statements in JavaScript?
- 7. Write about the usage of menu control in HTML.
- 8. How to create a moving box using DHTML? Explain.
- 9. Write about the lexical structure of JavaScript.
- 10. What are the three types of style sheet? Explain.



KARPAGAM ACADEMY OF HIGHER EDUCATION DEPARTMENT OF COMPUTER SCIENCE, CA & IT III B.Sc CS (Batch 2015-2018) WEB TECHNOLOGY PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS ONLINE EXAMINATIONS

UNIT-2

| S.No | Questions | opt1 | opt2 | opt3 | opt4 | Answer |
|------|---------------------------------------|---|--|-----------------------|---|-------------------------|
| | | | | | <input< td=""><td><input< td=""></input<></td></input<> | <input< td=""></input<> |
| | What is the HTML code for making | | <input< td=""><td></td><td>type="checkb</td><td>type="check"</td></input<> | | type="checkb | type="check" |
| 1 | a checkbox? | <checkbox></checkbox> | type="check"> | <check></check> | ox"> | > |
| | | | <input< th=""><th></th><th></th><th></th></input<> | | | |
| | What is the HTML code for making | <input< td=""><td>type="dropdo</td><td></td><td></td><td></td></input<> | type="dropdo | | | |
| 2 | a drop-down list? | type="list"> | wn"> | <select></select> | <list></list> | <select></select> |
| | | | <input< td=""><td></td><td></td><td></td></input<> | | | |
| | What is the HTML code for creating | <input< td=""><td>type="textbox</td><td></td><td><input area="</td"/><td></td></td></input<> | type="textbox | | <input area="</td"/> <td></td> | |
| 3 | a text area? | type="textarea"> | "> | <textarea></textarea> | text> | <textarea></textarea> |
| | Where in an HTML document will | | | | | In the |
| | you refer to an external style sheet? | At the end of the | At the top of | In the <head></head> | In the <head></head> | <head></head> |
| 4 | | document | the document | section | section | section |
| | Which HTML tag is used to define | | | | | |
| | an internal style sheet? | | | | | |
| 5 | | <style></style> | | | | |

| 7 | are used to select different kinds of user input. | HTML Forms | HTML Table | Html Lists | HTML frames | HTML Forms |
|----|---|-----------------|---------------|-------------------|----------------|------------------|
| 8 | is an area that can contain form elements. | table | form | lists | none | form |
| 9 | are used when you want the user to select one of a limited number of choices. | Check Buttons | buttons | Radio Buttons | list box | Radio Buttons |
| 10 | are used when you want the user to select one or more options of a limited number of choices. | Checkboxes | buttons | Radio Buttons | list box | Checkboxes |
| 11 | is the tag used to define an input. | <input/> | <read></read> | <inputs></inputs> | <get></get> | <input/> |
| 12 | element is used to include style information to the browser. | <style></style> | | | | |

| 16 | element allows the visitor to | texbox | checkbox | radio button | button | texbox |
|----|-----------------------------------|---------------|-------------------|---------------------|--|---------------|
| 10 | | ICAUGA | eneekoox | | outton | ICAUUX |
| | is used to clear the contents of | | | | | |
| 17 | the form | submit | clear | reset | post | reset |
| 17 | | Sublint | | 10501 | post | 10500 |
| | Large text can be given in forms | | | | | |
| 18 | using | text | large | largetext | textarea | textarea |
| | To avoid users from changing the | | | 0.00 | | |
| | size of browser window tag | | | | | |
| 19 | is used | resize | noresize | nochange | change | noresize |
| | | | | 6 | | |
| | A website can be made interactive | | | | | |
| 20 | with the help of | tables | frames | text | forms | forms |
| | | | | | | |
| | | | | | | |
| 21 | There are main parts in a form | 1 | 2 | 3 | 4 | 3 |
| | | | | | | |
| | tag is used initially to create | | | | <createfo< td=""><td></td></createfo<> | |
| 22 | a form | <form></form> | <create></create> | <initial></initial> | RM> | <form></form> |
| | | | | | | |
| | | | | | | |
| 23 | is the end tag of a form | | | | | |
| | | | | | | |
| | method is used to submit | | | | | |
| 24 | user information to the server | GET | SUBMIT | POST | RESET | POST |

| | method is used to get | | | | | |
|----|--|--------------|----------|---------------|---------|---------------|
| 25 | information from the server | GET | RETRIEVE | POST | RESET | GET |
| | What tag is used to give the visitors | | | | | |
| | the option of selecting one of the few | | | | | |
| 26 | options? | select | option | give | value | option |
| | In <select size="n"> represents</select> | | | | | |
| | the number of options that should be | | | | | |
| 27 | initially visible | size | n | select | no | n |
| | are form elements that | | | | | |
| | provides user to select one of the | | | | | |
| 28 | option | check box | text box | radio buttons | buttons | radio buttons |
| | | | | | | |
| | is used to make the radio | | | | | |
| 29 | button active by default | active | on | value | pressed | pressed |
| | | | | | rounded | rounded |
| | | | | rectangle | hollow | hollow |
| 30 | Radio buttons have | square boxes | lines | boxes | buttons | buttons |
| | | | | | | |
| | are used to make users to | | | | | |
| 31 | select more than one option | check box | text box | radio buttons | buttons | check box |
| | | | | | | |
| | elements allow users to fill | | | | | |
| 32 | up the information | check box | text box | radio buttons | buttons | text box |
| | A is a door within a window | | | | | |
| | where each door shows different | | | | | |
| 33 | information | tables | frames | text | forms | frames |

| 24 | Name the attribute used to provide | for an a | | 6 | 6 | c |
|----|--|-------------|--------------|--------------|----------|--------------|
| | frame spacing. | Irame | space | Tramespacing | Tramespc | Tramespacing |
| 35 | To show the scroll bar for a frame is used | scrolling | scroll | scr | src | scrolling |
| 36 | What tag is used to display an alternate text if the browser does not support frames | frameset | scrolling=no | noresize | noframes | noframes |
| 37 | A is a set of one or more rules that apply to an HTML document | tables | frames | stylesheet | forms | stylesheet |
| 57 | | | inumes | stylesheet | 1011115 | stylesheet |
| 38 | A rule of a stylesheet contains parts | 2 | 3 | 4 | 1 | 2 |
| 39 | part of a rule in stylesheet links the document and the style | declaration | selector | attribute | property | selector |
| 40 | part of a rule in stylesheet represents the effect | declaration | selector | attribute | property | declaration |
| 41 | The <style></style> | | | | | |

| 43 | Which among the following is not a value of font-size property | length | percentage | absolute | width | width |
|----|--|-----------------|-----------------|---------------------|-----------------|-----------------|
| 44 | The property of text transform is used to capitalize first letter of each word | lowercase | uppercase | capitalize | none | capitalize |
| 45 | tag of class is used to surround letters and words within these tags | <div></div> | <class></class> | | <row></row> | |
| 46 | tag of class is used to surround paragraphs and other block- level elements within these tags | <div></div> | <class></class> | | <row></row> | <div></div> |
| 47 | tag is unique to Netscape Navigator | <div></div> | <class></class> | | <layer></layer> | <layer></layer> |
| 48 | What are the two attributes used to change position of text or image using <layer> tag</layer> | top and bottom | top and left | bottom and right | top and right | top and left |
| 49 | property of layer tag is used to make a content visible or hidden | hide | appear | visibility | show | visibility |
| 50 | Which HTML tag is used to define an internal style sheet? | <style></style> | | | | |

| | The property describes how much space to insert between an | | | | | |
|----|---|-------------|------------------|----------------------|-----------|----------------|
| 52 | element and its margin | span | div | padding | merge | padding |
| 53 | Fonts are often measured in | inches | pixels | cm | points | points |
| 54 | The most common alignments are left,right,center and | vertical | top | horizontal | justify | justify |
| 55 | CSS has defined colors | 15 | 12 | 16 | 14 | 16 |
| 56 | Spacing can be provided by | space | spacer | borders | margins | margins |
| 57 | spacing is used to adjust the amount of space between letters | wordspacing | letterspacing | spacing | spacer | letterspacing |
| 58 | Which among the following is not a value of font-stretch property | normal | condensed | extended | left | left |
| 59 | DHTML is | Dyamic HTML | Detailed HTML | Discontinued HTML | Dead HTML | Dyamic HTML |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

UNIT-III

Syllabus:

JavaScript: Introduction-Operators-Starting with JavaScript:- Using Quotes-Using Alerts- Functions-Variables-data types- Statements-Comments. Objects: Working with Objects- Date Object-Math Object-String Object--Handling Events in JavaScript-Event Handling attributes-Window Events-Window Object – Document Object-Navigator Object

JAVA SCRIPT:

JavaScript is widely used scripting language for mainly validating the user input on the client side (browser). It is also used with the DHTML for creating interactive user interface design.

JavaScript is scripting language used for client side scripting. JavaScript developed by Netscape in 1995 as a method for validating forms and providing interactive content to web site. Microsoft and Netscape introduced JavaScript support in their browsers.

Benefits of JavaScript

Following are the benefits of JavaScript.

- arrays
- loosely typed variables
- regular expressions
- objects and classes
- highly evolved date, math, and string libraries
- W3C DOM support in the JavaScript

Disadvantages of JavaScript

- Developer depends on the browser support for the JavaScript
- There is no way to hide the JavaScript code in case of commercial application

JavaScript is arguably the most important programming language on earth. Once thought of as a toy, JavaScript is now the most widely deployed programming language in history. Almost everyone with a computer or a smart phone has all the tools they need to execute JavaScript programs



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

and to create their own. All you need is a browser and a text editor. JavaScript, HTML, and CSS have become so prevalent that many operating systems have adopted the open web standards as the presentation layer for native apps, including Windows 8, Firefox OS, Gnome, and Google's Chrome OS. Additionally, the iPhone and Android mobile devices support web views that allow them to incorporate Java Script and HTML5 functionality into native applications.

JavaScript is also moving into the hardware world. Projects like Arduino, Tessel, Espruino, and NodeBots foreshadow a time in the near future where JavaScript could be a common language for embedded systems and robotics.

Creating a JavaScript program is as simple as editing a text file and opening it in the browser. There are no complex development environments to download and install, and no complex IDE to learn. JavaScript is easy to learn, too. The basic syntax is immediately familiar to any programmer who has been exposed to the C family syntax.

No other language can boast a barrier to entry as low as JavaScript's. That low barrier to entry is probably the main reason that JavaScript was once widely (perhaps rightly) shunned as a toy. It was mainly used to create UI effects in the browser. That situation has changed. For a long time, there was no way to save data with JavaScript. If you wanted data to persist, you had to submit a form to a web server and wait for a page refresh. That hindered the process of creating responsive and dynamic web applications. However, in 2000, Microsoft started shipping Ajax technology in Internet Explorer. Soon after, other browsers added support for the XMLHttpRequest object.

Advantages of JavaScript

JavaScript didn't just luck into its position as the dominant client-side language on the Web. It is actually very well suited to be the language that took over the world. It is one of the most advanced and expressive programming languages developed to date. The following sections outline some of the features you may or may not be familiar with.

Operators

The operator itself is a keyword or symbol that does something to a value when used in an expression.

For example, the arithmetic operator + adds two values together.

The symbol is used in an expression with either one or two values and performs a calculation on the values to generate a result. For example, here is an expression that uses the multiplication operator: area = (width * height)



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

An expression is just like a mathematical expression. The values are known as operands. Operators that require only one operand (or value) are sometimes referred to as unary operators, while those that require two values are sometimes called binary operators. The different types of operators you will see in this section are:

- > Arithmetic operators
- > Assignment operators
- Comparison operators
- Logical operators
- > String operators

The explanation of above listed operators are as follows.

Arithmetic Operators

Arithmetic operators perform arithmetic operations upon operands. (Note that in the examples in the following table, x = 10.)

| Symb | ol Description | Example | Result |
|------|----------------|---------|--------|
| + | Addition | x+5 | 15 |
| | | | |
| - | Subtraction | x – 2 | 8 |
| | | | |
| * | Multiplication | x*3 | 30 |
| | | | |
| / | Division | x/2 | 5 |
| | | | |
| % | Modulus | x%3 | 1 |
| ++ | Increment | x++ | 11 |
| | | | |
| | Decrement | X | 9 |





CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Assignment Operators

The basic assignment operator is the equal sign, but do not take this to mean that it checks whether two values are equal. Rather, it's used to assign a value to the variable on the left of the equal sign, as you have seen in the previous section, which introduced variables.

The basic assignment operator can be combined with several other operators to allow you to assign a value to a variable and perform an operation in one step. For example, take a look at the following statement where there is an assignment operator and an arithmetic operator:

total = total - profit

This can be reduced to the following statement:

total -= profit

While it might not look like much, this kind of shorthand can save a lot of code if you have a lot of calculations like this (see table that follows) to perform.

**Symbol Example Using Shorthand Equivalent without Shorthand

+= x+=y x=x+y - = x - =y x=x - y

= x=y x=x*y

$$= x = v x = x/v$$

%= x%=y x=x%y

Comparison Operators

As you can see in the table that follows, comparison operators compare two operands and then return either true or false based on whether the comparison is true or not. Note that the comparison for checking whether two operands are equal is two equal signs (a single equal sign would be an assignment operator).

| Operator | Description | Example |
|----------|--------------------------|-----------------------|
| == | Equal to | 1==2 returns false |
| != | Not equal to | 1!=2 returns true |
| > | Greater than | 1 > 2 returns false |
| < | Less than | 1 < 2 returns true |
| >= | Greater than or equal to | 1 > = 2 returns false |
| < = | Less than or equal to | 1 < =2 returns true |





CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Logical or Boolean Operators

Logical or Boolean operators return one of two values: true or false. They are particularly helpful when you want to evaluate more than one expression at a time.

| Operator | Name | Description | Example (where x =1 and y =2) |
|----------|------|--------------------------------|--------------------------------------|
| & & | And | Allows you to check if both of | of |
| | | two conditions are met | (x < 2 & & y > 1) Returns true |
| ?? | Or | Allows you to check if one o | f |
| | | two conditions are met | (x < 2 ?? y < 2) Returns true |
| ! | Not | Allows you to check if | |
| | | something is not the case | ! $(x > y)$ Returns true |

String Operator (Using + with Strings)

You can also add text to strings using the + operator. For example, here the + operator is being used to add two variables that are strings together:

firstName = "Bob"

lastName = "Stewart"

name = firstName + lastName

The value of the name variable would now be Bob Stewart. The process of adding two strings together is known as concatenation.

You can also compare strings using the comparison operators you just met. For example, you could check whether a user has entered a specific value into a text box.

STARTING WITH JAVASCRIPT

Writing First Javascript Program

```
<script>
.....
</script>
```

document.write(" this is first program")

This is the first line of the program execution which has been taken up by the browser.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Starting to Program with JavaScript

Now that you have seen how JavaScript is able to access a document in the web browser using the DOM, it is time to look at how you use these properties and methods in scripts.

As I mentioned earlier, a programming language mainly performs calculations. So here are the key concepts you need to learn in order to perform different types of calculations:

A variable is used to store some information; it 's like a little bit of the computer 's memory where you can store numbers, strings (which are a series of characters), or references to objects. You can then perform calculations to alter the data held in variables within your code.

Operators perform functions on variables. There are different types of operators for example:

□ Arithmetic operators enable you to do things such as add (+) numbers together, or subtract

(-) one from another (providing they are numbers).

 \Box Comparison operators enable you to compare two strings and see if one is the same as the other, or different (for example, whether x is equal to y or whether a is greater than b).

Functions are parts of a script that are grouped together to perform a specific task. For example, you could have a function that calculates loan repayments, and when you tell the loan calculator function the information it needs (the amount of money to be borrowed, the number of years the loan will last, and the interest rate) the function will be able to return the monthly payment.

Functions are objects in their own right and are very similar to things called methods; one of the key differences is that methods often belong to an object already, whereas functions are customized.

Conditional statements allow you to perform different actions based upon a condition. For example, a condition might be whether a variable holding the current time is greater than 12. If the condition is true, code to write "Good Afternoon" might be run. Whereas, if it is less than 12, a different block of code saying "Good Morning" could be shown.

Loops can be set up so that a block of code runs a specified number of times or until a condition is met. For example, you can use a loop to get a document to write your name 100 times. There are also several built - in JavaScript objects that have methods that are of practical use.

For example, in the same way that the document object of the DOM has methods that allowed you to write to the document, the built - in JavaScript date object can tell you the date, time, or day of the week.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Using quotes

You can insert special characters such as ", ', ;, and &, which are otherwise reserved (because they have a special meaning in JavaScript), by using a backslash before them like so:

document.write("I want to use a \"quote\" mark \ & an ampersand.")

This writes out the following line to the browser:

I want to use a "quote" mark & an ampersand.

USING ALERTS

- If you want to be absolutely sure they see a message before doing anything on the website.
- You would like to warn the user about something. For example "the following page contains humor not suitable for those under the age of 14."
- An error has occurred and you want to inform the user of the problem.
- When asking users for confirmation of some action. For example, if they have just agreed to sign over the deed to their house and you want to ask them again if they are absolutely positive they want to go through with this decision!

Even though the above situations would all be valid times to use the alert function, you could also skip the alert popup and just have the error message, confirmation, etc displayed in plain HTML. More and more bigger sites are opting to lose JavaScript alerts and instead keep everything in HTML.

Coding a simple JavaScript alert

Just for fun, let's suppose that we are making an alert for some website that asks people to hand over the deed to their house. We need to add an alert to be sure these people are in agreement. The following code will add an alert by using an HTML button and the *onClick* event.

HTML & JavaScript Code:

```
<form>
<input type="button" onclick=
"alert('Are you sure you want to give us the deed to your house?')"
value="Confirmation Alert">
</form>
```

The alert() method displays an alert box with a specified message and an OK button.





CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

An alert box is often used if you want to make sure information comes through to the user.

<!DOCTYPE html> <html>

<body>

Click the button to demonstrate line-breaks in an alert box.

<button onclick="myFunction()">Try it</button>

<script>

```
function myFunction() {
```

alert("Hello\nHow are you?");

}

```
</script>
```

</body>

</html>




COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| DOCIYPE html <html> <body> Click the button to demonstrate line-breaks box.</body></html> | Click the button to demonstrate line-breaks in an alert box. Try it | |
|--|--|--|
| <button onclick="myEunction()">Try it <script></script></button> | | |

A function is a group of reusable code which can be called anywhere in your programme. This eliminates the need of writing same code again and again. This will help programmers to write modular code. You can divide your big programme in a number of small and manageable functions.

Like any other advance programming language, JavaScript also supports all the features necessary to write modular code using functions.

You must have seen functions like alert() and write() in previous chapters. We are using these function again and again but they have been written in core JavaScript only once.

JavaScript allows us to write our own functions as well. This section will explain you how to write your own functions in JavaScript.

JavaScript Functions

What is JavaScript Function?

Java script Function is nothing but it is a reusable code-block that is execute when the function is



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

called. Function is defined in the head section of the code. The syntax of JavaScript function is as follows:

```
function fname(prameter1,parameter2, ...)
{
JavaScript code 1
```

```
JavaScript code 2
```

While defining JavaScript function its important to remember that the keyword function should be in lowercase other wise JavaScript won't understand it as function. If you write function in uppercase the code will generate error. In the JavaScript semicolon is optional but its better to put semicolon as part of best practices. All the java script code are written inside the curly braces

Example:

```
<html>
<html>
<head>
<script language="javascript">
function showmessage()
{
alert("How are you");
}
</script>
</head>
<body>
<form>
<input type="button" value="Click Here!"
onclick="showmessage()" >
</form>
</body>
</html>
```

Function starts with the function keyword and code of the function is enclosed in {..} brackets. Functions are written inside the head section of the html document, because function does not execute when the page loads. The alert function displays the message "How are you", when you clicked on the button ("Click Here")..



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018



What is the use of JavaScript Functions:

The java Script Function is very useful in writing the JavaScript code. It is used to group many JavaScript codes under one name. For example you can write a function to validate email address.

Built-in functions

JavaScript provides many built in functions that ease the development of JavaScript programs. Here is the list of the built-in JavaScript functions:

| JavaScript Build in Function | Function Description |
|------------------------------|--|
| alert() | The alert() built-in function displays the alert dialog box. |
| confirm() | The confirm() built-in function display the confirmation dialog box. and ask the user to determine from the two option . |
| focus() | The focus() built -in function built the pointed object active and put the curser on the text field. |
| prompt() | The prompt() built -in function display the prompt dialog box. Inquiring the user for input. |



COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| select() | The select() built -in function used to select the pointed object. |
|----------|--|
| write() | The write() built in function used to write something on the document. |
| | |

Function arguments: Through variable you can pass argument to function. The out put of the function looks on the arguments given by you . Example:

```
<html>
<head>
<script language="javascript">
function myfunction(text)
{
confirm(text)
}
</script>
</head>
<body>
<form>
<input type="button"
onclick="myfunction('Do you want to delete it!')"
value="Delete">
<input type="button"
onclick="myfunction('Do you want to save it!')"
value="Save">
</form>
</body></html>
```

Before we use a function we need to define that function. The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces. The basic syntax is shown here:

```
<script type="text/javascript">
<!--
```





COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| function functionname(parameter-list) |
|---------------------------------------|
| { |
| statements |
| } |
| //> |
| |

Example:

A simple function that takes no parameters called sayHello is defined here:

```
<script type="text/javascript">
<!--
function sayHello()
{
    alert("Hello there");
}
//-->
</script>
```

Calling a Function:

To invoke a function somewhere later in the script, you would simple need to write the name of that function as follows:

```
<script type="text/javascript">
<!--
sayHello();
//-->
</script>
```



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Function Parameters:

Till now we have seen function without a parameters. But there is a facility to pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters.

A function can take multiple parameters separated by comma.

Example:

Let us do a bit modification in our sayHello function. This time it will take two parameters:

```
<script type="text/javascript">
<!--
function sayHello(name, age)
{
    alert( name + " is " + age + " years old.");
}
//-->
</script>
```

Now we can call this function as follows:

```
<script type="text/javascript">
<!--
sayHello('Zara', 7 );
//-->
</script>
```



The return Statement:

A JavaScript function can have an optional return statement. This is required if you want to return a value from a function. This statement should be the last statement in a function.

For example you can pass two numbers in a function and then you can expect from the function to return their multiplication in your calling program.

Now we can call this function as follows:

```
<script type="text/javascript">
<!--
var result;
result = concatenate('Zara', 'Ali');
alert(result );
//-->
</script>
```

JavaScript Variables:

Like many other programming languages, JavaScript has variables. Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container.

Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the **var** keyword as follows:

```
<script type="text/javascript">
<!--
var money;
var name;
//-->
```





COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

</script>

You can also declare multiple variables with the same **var** keyword as follows:

| <script type="text/javascript"></th></tr><tr><td><!</td></tr><tr><td>var money, name;</td></tr><tr><td>//></td></tr><tr><td></script> |
|---|
|---|

Storing a value in a variable is called variable initialization. You can do variable initialization at the time of variable creation or later point in time when you need that variable as follows:

For instance, you might create variable named money and assign the value 2000.50 to it later. For another variable you can assign a value the time of initialization as follows:

```
<script type="text/javascript">
<!--
var name = "Ali";
var money;
money = 2000.50;
//-->
</script>
```

JavaScript is un-typed language. This means that a JavaScript variable can hold a value of any data type. Unlike many other languages, you don't have to tell JavaScript during variable declaration what type of value the variable will hold. The value type of a variable can change during the execution of a program and JavaScript takes care of it automatically.

JavaScript Variable Scope:

The scope of a variable is the region of your program in which it is defined. JavaScript variable will have only two scopes.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

- **Global Variables:** A global variable has global scope which means it is defined everywhere in your JavaScript code.
- Local Variables: A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

Within the body of a function, a local variable takes precedence over a global variable with the same name. If you declare a local variable or function parameter with the same name as a global variable, you effectively hide the global variable.

Following example explains it:

```
<script type="text/javascript">
<!--
var myVar = "global"; // Declare a global variable
function checkscope() {
var myVar = "local"; // Declare a local variable
document.write(myVar);
}
//-->
</script>
```

This produces the following result:

local

JavaScript Variable Names:

While naming your variables in JavaScript keep following rules in mind.

- You should not use any of the JavaScript reserved keyword as variable name. These keywords are mentioned in the next section. For example, break or boolean variable names are not valid.
- JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or the underscore character. For example, 123test is an invalid variable name but _123test is a valid one.



COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

• JavaScript variable names are case sensitive. For example, Name and name are two different variables.

JavaScript Reserved Words:

The following are reserved words in JavaScript. They cannot be used as JavaScript variables, functions, methods, loop labels, or any object names.

| abstract | else | instanceof | switch | |
|----------|------------|------------|--------------|--|
| boolean | enum | int | synchronized | |
| break | export | interface | this | |
| byte | extends | long | throw | |
| case | false | native | throws | |
| catch | final | new | transient | |
| char | finally | null | true | |
| class | float | package | try | |
| const | for | private | typeof | |
| continue | function | protected | var | |
| debugger | goto | public | void | |
| default | if | return | volatile | |
| delete | implements | short | while | |
| do | import | static | with | |
| double | in | super | | |
| | | | | |

JavaScript DataTypes:

One of the most fundamental characteristics of a programming language is the set of data types it supports. These are the type of values that can be represented and manipulated in a programming language.

JavaScript allows you to work with three primitive data types:

- Numbers eg. 123, 120.50 etc.
- Strings of text e.g. "This text string" etc.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

• Boolean e.g. true or false.

JavaScript also defines two trivial data types, null and undefined, each of which defines only a single value.

Statements in JavaScript

A statement is an instruction that instructs a computer (JavaScript interpreter) to carry a specific action. For example,

```
document.write ("This is a JavaScript statement.");
```

instructs the computer to print "This is a JavaScript statement." In JavaScript, statements are terminated with a semicolon or return statement. Because statements in JavaScript are ended with a semicolon, multiple statements can be grouped on one line, for example:

var x; x = 60; document.write (x);

In JavaScript, statements can also be terminated with a line break character, for example,

var x x = 60 document.write (x)

The above three lines are treated as three separate statements as:

var x;

x = 60;

document.write (x);

To avoid ambiguity and any unintended results, use a semicolon to mark end of a statement instead of relying on an implicit semicolon insertion (or a linebreak character).

Working with conditional statements

A conditional statement uses conditional logic to determine what programming statements to execute. By using some conditional logic (a process of checking conditions), we determine if a certain



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

condition is true or false. If the condition is true, we execute some part of the program. Otherwise, if the condition is false, we use some other part of the program.

To create a condition in JavaScript, we need to write a conditional statement with a comparison operator. (JavaScript also supports **logical operators** to create complex conditions.) A comparison operator compares the value of one element with that of another. An expression that uses a comparison operator is referred to as a **Boolean expression**. A Boolean expression evaluates to either true or false.

Let's consider an example to simplify how to create a conditional statement. Suppose we want to write a small program in JavaScript to determine if 5 is less than 6. To answer that question, we will use an **IF statement**. In JavaScript, the general syntax for creating an IF statement is:

if (condition) {
some JavaScript code

}

An IF statement starts with the word "if"; next, the condition(s) is listed in parenthesis. After the opening curly brace ({ }), you list the code that you want to execute if the condition is true. All statements, before the closing brace (}), will be executed if the condition is true. If the condition is false, all the statements inside the IF statement will be skipped, not executed.

Let's write an IF statement that to evaluate whether 5 is less than 6. See the following code:

if (5 < 6) {

document.write ("
Condition is true : 5 is less than 6");
}

On line 1 above, we start our IF condition that says check if 5 is less than 6. The less than operator is a comparison operator. On line 2, we have a print statement that will be executed if the condition is true. On line 3, we end our if statement with the closing curly brace. In our IF statement above, 5 < 6 is a Boolean expression because it can be evaluated to either true or false. We know this expression is true so line 2 will be executed. Here is the complete example:

<script language="javascript">

document.write ("checking if 5 is less than 6......");



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

if (5 < 6) {
document.write ("
Condition is true : 5 is less than 6");
}
document.write ("
Done checking");
</script>

Here is the output of the above code:

Condition is true : 5 is less than 6...... Done checking

As we mentioned before, any statements inside an if block are executed only when the condition is true or is met. Because our condition (5 < 6) is true, the statement inside the IF block is executed to print "Condition is true: 5 is less than 6". The other print statements (outside of IF block) will execute regardless of the result of the Boolean expression.

In the following example, our Boolean expression evaluates to false (i.e., the condition is false or not met):

```
if (5 > 6) {
document.write ("<br>Condition is true : 5 is less than 6");
```

}

So why would the Boolean expression be false in the above example? Well, look closely at our condition. It says if 5 is greater than 6 then execute the statement inside the IF block. We know 5 is not greater than 6, so our Boolean expression (our condition) will evaluate to false. Thus the statement "Condition is true : 5 is less than 6" won't be printed.

Usually, when a condition is false, we want to execute some other code. For instance, suppose we ask the user to an integer, representing his/her age in years. If we get a valid number, we would use that number in our program. (In other words, our IF condition will be true so statements inside the IF block will be executed.)



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Otherwise, assume our program cannot use an invalid input. So if it is an invalid input we will again ask the user to enter a valid number. (This part of the code will be added with an ELSE block to the IF block.) Note any code outside of an IF block will be executed whether the condition is false or true.

To recap, any statements that you want to execute when the condition is true, place them inside the IF block; any statements that you want to execute when the condition is false, place them in an ELSE block. Look at the following example

if (5 > 6) { document.write ("Condition is true : 5 is **less** than 6");

```
}
```

```
else {
```

document.write ("Condition is false : 5 is **not** greater than 6");

}

So in this example, we are saying check if the 5 is greater than 6. If it is true, then, execute line 2; otherwise, execute line 5: document.write ("
Condition is false : 5 is **not** greater than 6");. Because 5 is not greater than 6, our condition is false and the statement inside the ELSE block will be executed. The following shows the output of the above code:

Condition is true : 5 is less than 6

You could also create two separate IF statements instead of using an ELSE clause, but this is not desirable especially when there are many conditions to check as that will require extra computing power.

Working with loops

Looping or iteration control is used to execute some set of instructions repeatedly. Without loops, we would have to write the instructions as many times we want them to execute. Let's print numbers 1 through 5 without using a loop:



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

<script language="javascript"> document.write (1 + "
"); document.write (2 + "
"); document.write (3 + "
"); document.write (4 + "
"); document.write (5 + "
");

The following shows the output of the above code:

| > 1 | |
|---------------|--|
| 2 | |
| 3 | |
| 4 | |
| 5 | |

Before we do this same example with a loop control structure, let's first point out that there are two types of loops:

- 1. loops that repeat a set number of times before quitting. This type of loop is created with a **for** loop. You can use a for loop when you know exactly how many times you want to execute some code.
- 2. loops that repeat until a certain condition is satisfied. This type of loop is created with a **while** loop. The while is useful for those situation in which it is known how many times a specific code needs to be run.

Please access the following pages for more information about JavaScript loop control structures:

Working with loops

With a for loop, we can run a group of statements a set number of times through the use of a **counter**, a variable that tracks the number of times the group of statements has been run. To create a for loop, we set an initial value for the counter variable and each time our command block is executed, we change the value of the counter. When the value of the counter variable reaches our stopping value, the loop will end. A general syntax for creating a JavaScript loop is:



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

for (startingValue, condition, iterationValue) {

JavaScript statements that you want to execute repeatedly

}

Where startingValue = the initial value of the counter variable

condition = a Boolean expression that must be true for the loop to continue and must become false for the loop to stop

iterationValue = a value to change the value of the counter variable

Let's create a for loop that will print numbers 1 through 5. The following shows the JavaScript code for creating a for loop:

```
<script language="javascript">
```

```
for (counter=1; counter <=5; counter++) {</pre>
```

document.write(counter + "
");

```
}
```

</script>

On line 2, we start our for loop with the JavaScript keyword for. With counter=1;, we declare a variable called counter and it is set to 1. The purpose of this statement is to set a initial value for our loop. The code counter <=5; is the Boolean expression.

The purpose of this code is to stop the for loop after the code inside the for loop has executed 5 times. The counter++ simply adds 1 each time the command inside the loop is executed. The following shows the output of this loop:

1
2
3
4
5



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

So how did we get this output? Let's review how the for loop code works. In our example, we start our counter variable with 1. Next, the Boolean expression, counter ≤ 5 ;, checks if the value of counter is less than or equal to 5. Is it? Yes, because we started our for loop with 1 and it is less than 5.

Then, with counter++ we add 1 to counter; so counter will become 2, after the print statement inside the for loop has executed. That will complete our first iteration. At the end of first iteration, the number 1 would be printed to the screen, the value of the counter variable would be 2.

The for loop does not end executing yet because our Boolean expression is still true. Our counter variable is 2, which is less than 5, so the print statement inside the for loop will print 2. Then, the value of counter will be updated again with 1, changing counter variable to 3. The for loop will continue to run like this until the value of counter reaches 6, at which point, the Boolean expression will be false because 6 is not less than or equal to 5. See table 1; it shows the value for the counter variable for each loop iteration. As the table shows, when the counter reaches 6, the for loop will stop because the Boolean expression would no longer be true.

| Table 1 for loop execution | | |
|----------------------------|---------------|------------------|
| Iteration | Counter value | Is counter <= 5? |
| First | 1 | Yes |
| Second | 2 | Yes |
| Third | 3 | Yes |
| Fourth | 4 | Yes |
| Fifth | 5 | Yes |
| Sixth | 6 | No |

In the previous example we incremented the for loop by 1 but we are not limited to incremented by just 1. We can specify other update expressions, as shown in the following for loops:



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

for (i = 5; i >=1; i--) { document.write (i + " "); } // prints 5 4 3 2 1 for (i = 2; i <= 64; i *= 2) { document.write (i + " "); } // prints 2 4 8 16 32 64 for (i = 1000; i > 10; i /= 5) { document.write (i + " "); } // prints 1000 200 40

In the first for loop, we initialize the variable i to 5 and decrement i by 1, as long as i is greater than or equal to 1. In the second for loop, we use the multiplication operator to update the variable i. Note we could we have rewritten the update expression as i = i * 2 to double the value of i each time for loop is executed while the value of i is less than or equal to 64. In the third loop, we use the division operator in the update expression to divide i by 5, as long as i remains greater than 10.

The while loop

The while loop is different from a for loop in the respect that it is known ahead of time how many times a while loop will execute. Suppose you ask the user for some input, the user may not enter the value you seek in the first try.

So using a while loop, we would keep requesting for the right input. But if you use a for loop, you would be limited to asking only once, twice, or however many times you use. The point is with a for loop, you initially specify how many times the for loop is to be executed, but with a while loop that is not necessarily true.

The general syntax for creating a while loop is:

```
while (condition) {
```

JavaScript statements that you want to execute repeatedly

```
}
```

Where condition = a Boolean expression that can be true or false. While the Boolean expression is true, the JavaScript code inside the while loop is executed.

The following shows an example of a while loop:

```
<script language="javascript">
var i = 1;
while (i <= 5) {
```



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

```
document.write (i + " ");
i++;
}
</script>
```

Note that for a while loop we initialize our counter variable, i, before the loop starts. Note also the counter variable is updated inside the while loop. This while loop prints:

1 2 3 4 5

Table 2 shows more examples of while loops and their corresponding output.

| JavaScript while loop codeOutput <script language="javascript">$\&$ 1var i = 1;3// this while loop prints odd numbers between 1 and 10.5while (i <= 10) {</td>9document.write (i + " ");9i += 2;<math><</math>}<math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><math><</math><t</th><th>Table 2 more examples of while loop in use</th><th></th></tr><tr><th><math><</math>script language="javascript">$\$ 1$var i = 1;3// this while loop prints odd numbers between 1 and 10.5while (i <= 10) {</td>9document.write (i + " ");9i += 2;$\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$</th><th>JavaScript while loop code</th><th>Output</th></tr><tr><td>var i = 1; 3 // this while loop prints odd numbers between 1 and 10. 7 while (i <= 10) {</td> 9 document.write (i + " "); 9 i += 2; <math> } // this while loop squares i as long as i is less than or equal to 1024. 8 while (i <= 1024) {</td> 64 document.write (i + ""); 128 i *= 2; 256 } 512 </math></td><td><script language="javascript"></td><td>》1</td></tr><tr><td>// this while loop prints odd numbers between 1 and 10.5while (i <= 10) {</td>9document.write (i + " ");9i += 2;<math>}<math><script language="javascript">2var i = 2;4// this while loop squares i as long as i is less than or16equal to 1024.16while (i <= 1024) {</td>64document.write (i + "");128i *= 2;256}512</math></math></td><td>var i = 1;</td><td>3</td></tr><tr><td>// this while loop prints out infinitely between 1 and 10. 7 while (i <= 10) {</td> 9 document.write (i + " "); i += 2; } <t</td><td>// this while loop prints odd numbers between 1 and 10</td><td>5</td></tr><tr><td>while (i <= 10) {</td> 9 document.write (i + " "); i += 2; }</td><td>// this while loop prints oud humbers between 1 and 10.</td><td>7</td></tr><tr><td>document.write (i + " "); i += 2; }</td><td>while (i <= 10) {</td><td>9</td></tr><tr><td>i += 2; } </script> <pre> </pre> <td>document.write (i + " ");</td> <td></td> | document.write (i + " "); | |
|--|---|--|
| $\begin{cases} \\ \\ \\ \\ & & & \\ \\ \\ var i = 2; & & 4 \\ \\ \\ // this while loop squares i as long as i is less than or \\ equal to 1024. & & \\ \\ \\ while (i <= 1024) \{ & & \\ \\ \\ document.write (i + ""); & & \\ \\ i *= 2; & & \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ $ | i += 2; | |
| $$>2$$ <script language="javascript"> $\$>2\$ var i = 2; 4 // this while loop squares i as long as i is less than or 8 i = 1024. 16 while (i <= 1024) {</td> 64 document.write (i + " "); 128 i *= 2; 256 } 512</td><td>}</td><td></td></tr><tr><td><math><</math> script language="javascript"> \checkmark 2 $var i = 2;$ 4 $//$ this while loop squares i as long as i is less than or 8 $equal to 1024.$ 16 32 32 while (i <= 1024) {</td> 64 document.write (i + " "); 128 $i *= 2;$ 256 \rbrace 512</td><td></script> | | |
| var i = 2;4// this while loop squares i as long as i is less than or8equal to 1024.163232while (i <= 1024) { | <script language="javascript"></script> | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| | 1024 |
|--|------|
| | |

JavaScript Comments

- Comments will not be executed by JavaScript.
- Comments can be added to explain the JavaScript, or to make the code more readable.
- Single line comments start with //.
- The following example uses single line comments to explain the code:

Example

// Write to a heading:

document.getElementById("myH1").innerHTML="Welcome to my Homepage";

// Write to a paragraph:

document.getElementById("myP").innerHTML="This is my first paragraph.";

JavaScript Multi-Line Comments

Multi line comments start with /* and end with */.

The following example uses a multi line comment to explain the code:

Example

/*

The code below will write to a heading and to a paragraph, and will represent the start of my homepage: */

document.getElementById("myH1").innerHTML="Welcome to my Homepage"; document.getElementById("myP").innerHTML="This is my first paragraph.";

Using Comments to Prevent Execution

In the following example the comment is used to prevent the execution of one of the codelines (can be suitable for debugging):

Example



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

//document.getElementById("myH1").innerHTML="Welcome to my Homepage"; document.getElementById("myP").innerHTML="This is my first paragraph.";

In the following example the comment is used to prevent the execution of a code block (can be suitable for debugging):

Example

/*

document.getElementById("myH1").innerHTML="Welcome to my Homepage"; document.getElementById("myP").innerHTML="This is my first paragraph."; */ Using Comments at the End of a Line

In the following example the comment is placed at the end of a code line: Example

var x=5; // declare x and assign 5 to it var y=x+2; // declare y and assign x+2 to it

JavaScript supports three different types of comments:

Multiple-line C-style comments. Everything between /* and */ is a comment, for example:

/* This is a comment */

/* C-style comments can span

as many lines as you like,

as shown in this example */

One-line comments of C++ style. These comments begin with // and continue up to the next line break:

// This is a one-line comment

One-line comments with the HTML comment-opening sequence (<!--). Note that the JavaScript interpreter ignores the closing characters of HTML comments (-->). Consider this example:

<!-- This is treated as a one-line JS comment

<!-- It works just like a comment beginning with //



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

- <!-- --> This is also a one-line JS comment
- <!-- --> because JS ignores the closing characters
- <!-- --> of HTML-style comments

HTML-style comments are not usually found in the middle of JavaScript code. (The // comments are simpler and easier to read.) However, they are useful for hiding JavaScript code from old browsers.

Working With Objects:

With the combination of prototypal inheritance, dynamic object extension, and closures,

JavaScript has one of the most flexible and expressive object systems available in any popular programming language.

In JavaScript, all types of functions, arrays, key/value pairs, and data structures in general are really objects. Even primitive types get the object treatment when you refer to them with the property access notations. They get automatically wrapped with an object so that you can call their prototype methods. For example:

'tonya@example.com'.split('@')[1]; // => example.com

Primitive types behave like objects when you use the property access notations, but you can't assign new properties to them. Primitives get wrapped with an object temporarily, and then that object is immediately thrown away.

Any attempt to assign values to properties will seem to succeed, but subsequent attempts to access that new property will fail.

JavaScript's object system is so powerful and expressive that most of the complexity in common OO patterns melts away when you reproduce them in JavaScript. You simply don't need all of the common croft to accomplish the stated goals.

For instance, because JavaScript is classless, and it's possible to create an object on demand at the precise moment it's needed (lazy instantiation), the singleton is reduced to an object literal:



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

var highlander = {
name: 'McLeod',
catchphrase: 'There can be only one.'
};

.

As you continue through this chapter, you'll see that much of the overhead associated with several other GoF design patterns melts away when you understand how to take advantage of JavaScript's native object capabilities.

You may be aware that JavaScript is not a classical OO language. It's a prototypal language. However, most JavaScript training materials ignore some of the implications of that paradigm shift.

It's time to get a firmer handle on exactly what prototypal means and how you can take advantage of it to write better, faster, more readable, and more efficient code. It might help to get a better sense of the shortcomings of classical inheritance first.

Classical Inheritance Is Obsolete

Those who are unaware they are walking in darkness will never seek the light-Bruce Lee

In Design Patterns: Elements of Reusable Object Oriented Software, the Gang of Four opened the book with two foundational principles of object-oriented design:

1. Program to an interface, not an implementation.

2. Favor object composition over class inheritance.

In a sense, the second principle could follow from the first, because inheritance exposes the parent class to all child classes. The child classes are all programming to an implementation, not an interface. Classical inheritance breaks the principle of encapsulation and tightly couples the child class to its ancestors.

Think of it this way: classical inheritance is like Ikea furniture. You have a bunch of pieces that are designed to fit together in a very specific way.

If everything goes exactly according to plan, chances are high that you'll come out with a usable piece of furniture; but if anything at all goes wrong or deviates from the preplanned



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

specification, there is little room for adjustment or flexibility. Here's where the analogy (and the furniture and the software) breaks down: the design is in a constant state of change.

Composition is more like Lego blocks. The various pieces aren't designed to fit with any specific piece. Instead, they are all designed to fit together with any other piece, with few exceptions.

When you design for classical inheritance, you design a child class to inherit from a specific parent class. The specific parent class name is usually hardcoded right in the child class, with no mechanism to override it.

Right from the start, you're boxing yourself in limiting the ways that you can reuse your code without rethinking its design at a fundamental level.

When you design for composition, the sky is the limit. As long as you can successfully avoid colliding with properties from other source objects, objects can be composed and reused virtually any way you see fit.

Once you get the hang of it, composition affords a tremendous amount of freedom compared to classical inheritance. For people who have been immersed in classical inheritance for years and learn how to take real advantage of composition (specifically using prototypal techniques), it is like walking out of a dark tunnel into the light and seeing a whole new world of possibilities open up for you.

Back to Design Patterns. Why is the seminal work on object-oriented design so distinctly antiinheritance? Because inheritance causes several problems:

Tight coupling

Inheritance is the tightest coupling available in OO design. Descendant classes have an intimate knowledge of their ancestor classes.

Inflexible hierarchies

Single-parent hierarchies are rarely capable of describing all possible use cases. Eventually, all hierarchies are "wrong" for new uses—a problem that necessitates code duplication.

Multiple inheritances is complicated

It's often desirable to inherit from more than one parent. That process is inordinately complex, and its implementation is inconsistent with the process for single inheritance, which makes it harder to read and understand.



COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Brittle architecture

With tight coupling, it's often difficult to refactor a class with the "wrong" design, because much existing functionality depends on the existing design.

The gorilla/banana problem

There are often parts of the parent that you don't want to inherit. Subclassing allows you to override properties from the parent, but it doesn't allow you to select which properties you want to inherit.

These problems are summed up nicely by Joe Armstrong in Coders at Work by Peter Siebel:

The problem with object-oriented languages is they've got this entire implicit environment that they carry around with them. You wanted a banana but what you got was a gorilla holding the banana and the entire jungle. Inheritance works beautifully for a short time, but eventually the app architecture becomes arthritic. When you've built up your entire app on a foundation of classical inheritance, the dependencies on ancestors run so deep that even reusing or changing trivial amounts of code can turn into a gigantic refactor. Deep inheritance trees are

brittle, inflexible, and difficult to extend.

More often than not, what you wind up with in a mature classical OO application is a range of possible ancestors to inherit from, all with slightly different but often similar configurations.

Figuring out which to use is not straightforward, and you soon have a haphazard collection of similar objects with unexpectedly divergent properties. Around this time, people start throwing around the word "rewrite" as if it's an easier undertaking than refactoring the current mess.

Many of the patterns in the GoF book were designed specifically to address these well known problems. In many ways, the book itself can be read as a critique of the shortcomings of most classical OO languages, along with the accompanying lengthy workarounds.

In short, patterns point out deficiencies in the language. You can reproduce all of the GoF patterns in JavaScript, but before you start using them as blueprints for your JavaScript code, you'll want to get a good handle on JavaScript's prototypal and functional capabilities.

For a long time, many people were confused about whether JavaScript is truly object oriented, because they felt it lacked features from other OO languages. Setting aside the fact that JavaScript handles classical inheritance with less code than most class-based languages, coming to JavaScript and asking how to do classical inheritance is like picking up a touch-screen mobile phone and asking where



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

the rotary dial is. Of course, people will be amused when the next thing out of your mouth is, "If it doesn't have a rotary dial, it's not a telephone!"

JavaScript can do most of the OO things you're accustomed to in other languages, such as inheritance, data privacy, and polymorphism. However, JavaScript has many native capabilities that make some classical OO features and patterns obsolete. It's better to stop asking, "How do I do classical inheritance in JavaScript?" and start asking, "What cool new things does JavaScript enable me to do?"

I wish I could tell you that you'll never have to deal with classical inheritance in JavaScript. Unfortunately, because classical inheritance is easy to mimic in JavaScript, and many people come from class-based programming backgrounds, there are several popular libraries that feature classical inheritance prominently, including Backbone.js, which you'll have a chance to explore soon. When you do encounter situations in which you're forced to subclass by other programmers, keep in mind that inheritance hierarchies should be kept as small as possible. Avoid subclassing subclasses, remember that you can mix and match different code reuse styles, and things will go more smoothly.

Date Object:

The Date object is a datatype built into the JavaScript language. Date objects are created with the **new Date**() as shown below.

Once a Date object is created, a number of methods allow you to operate on it. Most methods simply allow you to get and set the year, month, day, hour, minute, second, and millisecond fields of the object, using either local time or UTC (universal, or GMT) time.

The ECMAScript standard requires the Date object to be able to represent any date and time, to millisecond precision, within 100 million days before or after 1/1/1970. This is a range of plus or minus 273,785 years, so the JavaScript is able to represent date and time till year 275755.

Syntax:

Here are different variant of Date() constructor:

| new Date() | |
|------------|--|
| | |

new Date(milliseconds)





COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

new Date(datestring)

new Date(year,month,date[,hour,minute,second,millisecond])

Here is the description of the parameters:

- No Argument: With no arguments, the Date() constructor creates a Date object set to the current date and time.
- Milli seconds: When one numeric argument is passed, it is taken as the internal numeric representation of the date in milliseconds, as returned by the getTime() method. For example, passing the argument 5000 creates a date that represents five seconds past midnight on 1/1/70.
- **Date string:** When one string argument is passed, it is a string representation of a date, in the format accepted by the Date.parse() method.
- **7 arguments:** To use the last form of constructor given above, Here is the description of each argument:
 - 1. **year:** Integer value representing the year. For compatibility (in order to avoid the Y2K problem), you should always specify the year in full; use 1998, rather than 98.
 - 2. **month:** Integer value representing the month, beginning with 0 for January to 11 for December.
 - 3. date: Integer value representing the day of the month.
 - 4. **hour:** Integer value representing the hour of the day (24-hour scale).
 - 5. **minute:** Integer value representing the minute segment of a time reading.
 - 6. second: Integer value representing the second segment of a time reading.
 - 7. **millisecond:** Integer value representing the millisecond segment of a time reading.

Date Properties:

Here is a list of each property and their description.

| Property | Description |
|-------------|---|
| constructor | Specifies the function that creates an object's prototype. |
| prototype | The prototype property allows you to add properties and methods to an object. |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Date Methods:

Here is a list of each method and its description.

| Method | Description | |
|----------------------|--|--|
| Date() | Returns today's date and time | |
| getDate() | Returns the day of the month for the specified date according to local | |
| | time. | |
| getDay() | Returns the day of the week for the specified date according to local | |
| | time. | |
| getFullYear() | Returns the year of the specified date according to local time. | |
| getHours() | Returns the hour in the specified date according to local time. | |
| getMilliseconds() | Returns the milliseconds in the specified date according to local time. | |
| getMinutes() | Returns the minutes in the specified date according to local time. | |
| getMonth() | Returns the month in the specified date according to local time. | |
| getSeconds() | Returns the seconds in the specified date according to local time. | |
| getTime() | Returns the numeric value of the specified date as the number of | |
| | milliseconds since January 1, 1970, 00:00:00 UTC. | |
| getTimezoneOffset() | Returns the time-zone offset in minutes for the current locale. | |
| getUTCDate() | Returns the day (date) of the month in the specified date according to | |
| | universal time. | |
| getUTCDay() | Returns the day of the week in the specified date according to universal | |
| | time. | |
| getUTCFullYear() | Returns the year in the specified date according to universal time. | |
| getUTCHours() | Returns the hours in the specified date according to universal time. | |
| getUTCMilliseconds() | Returns the milliseconds in the specified date according to universal | |
| | time. | |
| getUTCMinutes() | Returns the minutes in the specified date according to universal time. | |
| getUTCMonth() | Returns the month in the specified date according to universal time. | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| getUTCSeconds() | Returns the seconds in the specified date according to universal time. | | |
|----------------------|---|--|--|
| getYear() | Deprecated - Returns the year in the specified date according to local | | |
| | time. Use getFullYear instead. | | |
| setDate() | Sets the day of the month for a specified date according to local time. | | |
| setFullYear() | Sets the full year for a specified date according to local time. | | |
| setHours() | Sets the hours for a specified date according to local time. | | |
| setMilliseconds() | Sets the milliseconds for a specified date according to local time. | | |
| setMinutes() | Sets the minutes for a specified date according to local time. | | |
| setMonth() | Sets the month for a specified date according to local time. | | |
| setSeconds() | Sets the seconds for a specified date according to local time. | | |
| setTime() | Sets the Date object to the time represented by a number of | | |
| | milliseconds since January 1, 1970, 00:00:00 UTC. | | |
| setUTCDate() | Sets the day of the month for a specified date according to universal | | |
| | time. | | |
| setUTCFullYear() | Sets the full year for a specified date according to universal time. | | |
| setUTCHours() | Sets the hour for a specified date according to universal time. | | |
| setUTCMilliseconds() | Sets the milliseconds for a specified date according to universal time. | | |
| setUTCMinutes() | Sets the minutes for a specified date according to universal time. | | |
| setUTCMonth() | Sets the month for a specified date according to universal time. | | |
| setUTCSeconds() | Sets the seconds for a specified date according to universal time. | | |
| setYear() | Deprecated - Sets the year for a specified date according to local time. | | |
| | Use setFullYear instead. | | |
| toDateString() | Returns the "date" portion of the Date as a human-readable string. | | |
| | | | |
| toGMTString() | Deprecated - Converts a date to a string, using the Internet GMT | | |
| | conventions. Use toUTCString instead. | | |
| toLocaleDateString() | Returns the "date" portion of the Date as a string, using the current | | |
| | locale's conventions. | | |
| toLocaleFormat() | Converts a date to a string, using a format string. | | |
| toLocaleString() | Converts a date to a string, using the current locale's conventions. | | |
| toLocaleTimeString() | Returns the "time" portion of the Date as a string, using the current | | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| | locale's conventions. | |
|----------------|---|--|
| toSource() | Returns a string representing the source for an equivalent Date object; | |
| | you can use this value to create a new object. | |
| toString() | Returns a string representing the specified Date object. | |
| toTimeString() | Returns the "time" portion of the Date as a human-readable string. | |
| toUTCString() | Converts a date to a string, using the universal time convention. | |
| valueOf() | Returns the primitive value of a Date object. | |

Date Static Methods:

In addition to the many instance methods listed previously, the Date object also defines two static methods. These methods are invoked through the Date() constructor itself:

| Method | Description | |
|--------------|--|--|
| Date.parse() | Parses a string representation of a date and time and returns the internal | |
| | millisecond representation of that date. | |
| Date.UTC() | Returns the millisecond representation of the specified UTC date and | |
| | time. | |

Math Object:

The **math** object provides you properties and methods for mathematical constants and functions.

Unlike the other global objects, Math is not a constructor. All properties and methods of Math are static and can be called by using Math as an object without creating it.

Thus, you refer to the constant pi as **Math.PI** and you call the *sine* function as **Math.sin**(x), where x is the method's argument.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Syntax:

Here is the simple syntax to call properties and methods of Math.

var pi_val = Math.PI; var sine_val = Math.sin(30);

Math Properties:

Here is a list of each property and their description.

| Property | Description |
|----------|---|
| Е | Euler's constant and the base of natural logarithms, approximately 2.718. |
| LN2 | Natural logarithm of 2, approximately 0.693. |
| LN10 | Natural logarithm of 10, approximately 2.302. |
| LOG2E | Base 2 logarithm of E, approximately 1.442. |
| LOG10E | Base 10 logarithm of E, approximately 0.434. |
| PI | Ratio of the circumference of a circle to its diameter, approximately 3.14159. |
| SQRT1_2 | Square root of 1/2; equivalently, 1 over the square root of 2, approximately 0.707. |
| SQRT2 | Square root of 2, approximately 1.414. |

Math Methods

Here is a list of each method and its description.

| Method | Description |
|---------|---|
| abs() | Returns the absolute value of a number. |
| acos() | Returns the arccosine (in radians) of a number. |
| asin() | Returns the arcsine (in radians) of a number. |
| atan() | Returns the arctangent (in radians) of a number. |
| atan2() | Returns the arctangent of the quotient of its arguments. |
| ceil() | Returns the smallest integer greater than or equal to a number. |
| cos() | Returns the cosine of a number. |
| exp() | Returns E^N , where N is the argument, and E is Euler's constant, the base of the |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| | natural logarithm. | | |
|------------|---|--|--|
| floor() | Returns the largest integer less than or equal to a number. | | |
| log() | Returns the natural logarithm (base E) of a number. | | |
| max() | Returns the largest of zero or more numbers. | | |
| min() | Returns the smallest of zero or more numbers. | | |
| pow() | Returns base to the exponent power, that is, base exponent. | | |
| random() | Returns a pseudo-random number between 0 and 1. | | |
| round() | Returns the value of a number rounded to the nearest integer. | | |
| sin() | Returns the sine of a number. | | |
| sqrt() | Returns the square root of a number. | | |
| tan() | Returns the tangent of a number. | | |
| toSource() | Returns the string "Math". | | |

Event:

What is an Event?

JavaScript's interaction with HTML is handled through events that occur when the user or browser manipulates a page.

When the page loads, that is an event. When the user clicks a button, that click, too, is an event. Another example of events are like pressing any key, closing window, resizing window etc.

Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable to occur.

Events are a part of the Document Object Model (DOM) Level 3 and every HTML element have a certain set of events which can trigger JavaScript Code.





COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

onclick Event Type:

This is the most frequently used event type which occurs when a user clicks mouse left button. You can put your validation, warning etc against this event type.

Example:

| <html></html> | |
|--|--|
| <head></head> | |
| <script type="text/javascript"></td><td></td></tr><tr><td><!</td><td></td></tr><tr><td><pre>function sayHello() {</pre></td><td></td></tr><tr><td>alert("Hello World")</td><td></td></tr><tr><td>}</td><td></td></tr><tr><td>//></td><td></td></tr><tr><td></script> | |
| | |
| <body></body> | |
| <input onclick="sayHello()" type="button" value="Say Hello"/> | |
| | |
| | |

This will produce following result and when you click Hello button then onclick event will occur which will trigger sayHello() function.

onsubmit event type:

Another most important event type is onsubmit. This event occurs when you try to submit a form. So you can put your form validation against this event type.

Here is simple example showing its usage. Here we are calling a validate() function before submitting a form data to the webserver. If validate() function returns true the form will be submitted otherwise it will not submit the data.



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Example:

| <html></html> | |
|--|--|
| <head></head> | |
| <script type="text/javascript"></td><td></td></tr><tr><td><!</td><td></td></tr><tr><td>function validation() {</td><td></td></tr><tr><td>all validation goes here</td><td></td></tr><tr><td></td><td></td></tr><tr><td>return either true or false</td><td></td></tr><tr><td>}</td><td></td></tr><tr><td>//></td><td></td></tr><tr><td></script> | |
| | |
| <body></body> | |
| <form action="t.cgi" method="POST" onsubmit="return validate()"></form> | |
| | |
| <input type="submit" value="Submit"/> | |
| | |
| | |
| | |
| | |

onmouseover and onmouseout:

These two event types will help you to create nice effects with images or even with text as well. The onmouseover event occurs when you bring your mouse over any element and the onmouseout occurs when you take your mouse out from that element.

Example:

Following example shows how a division reacts when we bring our mouse in that division:

<html>

<head>



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

```
<script type="text/javascript">
<!--
function over() {
 alert("Mouse Over");
}
function out() {
 alert("Mouse Out");
}
//-->
</script>
</head>
<body>
<div onmouseover="over()" onmouseout="out()">
<h2> This is inside the division </h2>
</div>
</body>
</html>
```

You can change different images using these two event types or you can create help baloon to help your users.

HTML 4 Standard Events

The standard HTML 4 events are listed here for your reference. Here script indicates a Javascript function to be executed against that event.

| Event | Value | Description |
|----------|--------|--|
| onchange | script | Script runs when the element changes |
| onsubmit | script | Script runs when the form is submitted |
| onreset | script | Script runs when the form is reset |
| onselect | script | Script runs when the element is selected |





COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| onblur | script | Script runs when the element loses focus |
|-------------|--------|--|
| onfocus | script | Script runs when the element gets focus |
| onkeydown | script | Script runs when key is pressed |
| onkeypress | script | Script runs when key is pressed and released |
| onkeyup | script | Script runs when key is released |
| onclick | script | Script runs when a mouse click |
| ondblclick | script | Script runs when a mouse double-click |
| onmousedown | script | Script runs when mouse button is pressed |
| onmousemove | script | Script runs when mouse pointer moves |
| onmouseout | script | Script runs when mouse pointer moves out of an element |
| onmouseover | script | Script runs when mouse pointer moves over an element |
| onmouseup | script | Script runs when mouse button is released |

HTML Event Attributes

Global Event Attributes

HTML 4 added the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element.

Below are the global event attributes that can be added to HTML elements to define event actions.




CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Window Event Attributes

Events triggered for the window object (applies to the <body> tag):

| Attribute | Value | Description |
|----------------|--------|--|
| onafterprint | script | Script to be run after the document is printed |
| onbeforeprint | script | Script to be run before the document is printed |
| onbeforeunload | script | Script to be run before the document is unloaded |
| onerror | script | Script to be run when an error occur |
| onhaschange | script | Script to be run when the document has changed |
| onload | script | Fires after the page is finished loading |
| onmessage | script | Script to be run when the message is triggered |
| onoffline | script | Script to be run when the document goes offline |
| ononline | script | Script to be run when the document comes online |
| onpagehide | script | Script to be run when the window is hidden |
| onpageshow | script | Script to be run when the window becomes visible |
| onpopstate | script | Script to be run when the window's history changes |
| onredo | script | Script to be run when the document performs a redo |
| onresize | script | Fires when the browser window is resized |
| onstorage | script | Script to be run when a Web Storage area is updated |
| onundo | script | Script to be run when the document performs an undo |
| onunload | script | Fires once a page has unloaded (or the browser window has been closed) |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

Form Events

Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements):

| Attribute | Value | Description |
|-----------------|--------|--|
| onblur | script | Fires the moment that the element loses focus |
| onchange | script | Fires the moment when the value of the element is changed |
| oncontextmenu | script | Script to be run when a context menu is triggered |
| <u>onfocus</u> | script | Fires the moment when the element gets focus |
| onformchange | script | Script to be run when a form changes |
| onforminput | script | Script to be run when a form gets user input |
| oninput | script | Script to be run when an element gets user input |
| oninvalid | script | Script to be run when an element is invalid |
| onreset | script | Fires when the Reset button in a form is clicked Not supported in HTML5 |
| onselect | script | Fires after some text has been selected in an element |
| <u>onsubmit</u> | script | Fires when a form is submitted |

Keyboard Events

| Attribute | Value | Description |
|-----------|--------|-------------------------------------|
| onkeydown | script | Fires when a user is pressing a key |





CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| onkeypress | script | Fires when a user presses a key |
|------------|--------|----------------------------------|
| onkeyup | script | Fires when a user releases a key |

Mouse Events

Events triggered by a mouse, or similar user actions:

| Attribute | Value | Description |
|-------------|--------|--|
| onclick | script | Fires on a mouse click on the element |
| ondblclick | script | Fires on a mouse double-click on the element |
| ondrag | script | Script to be run when an element is dragged |
| ondragend | script | Script to be run at the end of a drag operation |
| ondragenter | script | Script to be run when an element has been dragged to a valid drop target |
| ondragleave | script | Script to be run when an element leaves a valid drop target |
| ondragover | script | Script to be run when an element is being dragged over a valid drop target |
| ondragstart | script | Script to be run at the start of a drag operation |
| ondrop | script | Script to be run when dragged element is being dropped |
| onmousedown | script | Fires when a mouse button is pressed down on an element |
| onmousemove | script | Fires when the mouse pointer moves over an element |
| onmouseout | script | Fires when the mouse pointer moves out of an element |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| onmouseover | script | Fires when the mouse pointer moves over an element |
|------------------|--------|--|
| <u>onmouseup</u> | script | Fires when a mouse button is released over an element |
| onmousewheel | script | Script to be run when the mouse wheel is being rotated |
| onscroll | script | Script to be run when an element's scrollbar is being scrolled |

Media Events

Events triggered by medias like videos, images and audio (applies to all HTML elements, but is most common in media elements, like <audio>, <embed>, , <object>, and <video>):

| Attribute | Value | Description |
|------------------|--------|---|
| onabort | script | Script to be run on abort |
| oncanplay | script | Script to be run when a file is ready to start playing (when it has buffered enough to begin) |
| oncanplaythrough | script | Script to be run when a file can be played all the way to the end without pausing for buffering |
| ondurationchange | script | Script to be run when the length of the media changes |
| onemptied | script | Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects) |
| onended | script | Script to be run when the media has reach the end (a useful event for messages like "thanks for listening") |
| onerror | script | Script to be run when an error occurs when the file is being loaded |
| onloadeddata | script | Script to be run when media data is loaded |
| onloadedmetadata | script | Script to be run when meta data (like dimensions and duration) are loaded |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| onloadstart | script | Script to be run just as the file begins to load before anything is |
|--------------------|--------|--|
| | | actually loaded |
| | • . | |
| onpause | script | Script to be run when the media is paused either by the user or |
| | | programmatically |
| onplay | script | Script to be run when the media is ready to start playing |
| onplaying | script | Script to be run when the media actually has started playing |
| onprogress | script | Script to be run when the browser is in the process of getting the |
| | | media data |
| | • . | |
| onratechange | script | Script to be run each time the playback rate changes (like when a user |
| | | switches to a slow motion or fast forward mode) |
| onreadystatechange | script | Script to be run each time the ready state changes (the ready state |
| | | tracks the state of the media data) |
| onseeked | script | Script to be run when the seeking attribute is set to false indicating |
| | | that seeking has ended |
| onseeking | script | Script to be run when the seeking attribute is set to true indicating that |
| | | seeking is active |
| onstalled | script | Script to be run when the browser is unable to fetch the media data for |
| | | whatever reason |
| onsuspend | script | Script to be run when fetching the media data is stopped before it is |
| | sinp | completely loaded for whatever reason |
| | | |
| ontimeupdate | script | Script to be run when the playing position has changed (like when the |
| | | user fast forwards to a different point in the media) |
| onvolumechange | script | Script to be run each time the volume is changed which (includes |
| | | setting the volume to "mute") |
| onwaiting | script | Script to be run when the media has paused but is expected to resume |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| | (like when the media pauses to buffer more data) |
|--|--|
|--|--|

The Window Object

Window Object

The window object represents an open window in a browser.

If a document contain frames (<frame> or <iframe> tags), the browser creates one window object for the HTML document, and one additional window object for each frame.

Window Object Properties

| Property | Description |
|------------------|--|
| closed | Returns a Boolean value indicating whether a window has been closed or not |
| defaultStatus | Sets or returns the default text in the statusbar of a window |
| document | Returns the Document object for the window (See Document object) |
| frames | Returns an array of all the frames (including iframes) in the current window |
| history | Returns the History object for the window (See History object) |
| innerHeight | Returns the inner height of a window's content area |
| innerWidth | Returns the inner width of a window's content area |
| length | Returns the number of frames (including iframes) in a window |
| location | Returns the Location object for the window (See Location object) |
| name | Sets or returns the name of a window |
| <u>navigator</u> | Returns the Navigator object for the window (See Navigator object) |
| opener | Returns a reference to the window that created the window |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| outerHeight | Returns the outer height of a window, including toolbars/scrollbars |
|------------------|---|
| | |
| outerWidth | Returns the outer width of a window, including toolbars/scrollbars |
| pageXOffset | Returns the pixels the current document has been scrolled (horizontally) from the |
| | upper left corner of the window |
| pageYOffset | Returns the pixels the current document has been scrolled (vertically) from the |
| | upper left corner of the window |
| parent | Returns the parent window of the current window |
| screen | Returns the Screen object for the window (See Screen object) |
| screenLeft | Returns the x coordinate of the window relative to the screen |
| screenTop | Returns the y coordinate of the window relative to the screen |
| screenX | Returns the x coordinate of the window relative to the screen |
| screenY | Returns the y coordinate of the window relative to the screen |
| self | Returns the current window |
| status | Sets or returns the text in the statusbar of a window |
| top | Returns the topmost browser window |
| Window Object Me | thods |

Window Object Methods

| Method | Description |
|----------------|---|
| <u>alert()</u> | Displays an alert box with a message and an OK button |
| <u>atob()</u> | Decodes a base-64 encoded string |
| <u>blur()</u> | Removes focus from the current window |
| btoa() | Encodes a string in base-64 |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| <u>clearInterval()</u> | Clears a timer set with setInterval() |
|------------------------|--|
| <u>clearTimeout()</u> | Clears a timer set with setTimeout() |
| <u>close()</u> | Closes the current window |
| <u>confirm()</u> | Displays a dialog box with a message and an OK and a Cancel button |
| createPopup() | Creates a pop-up window |
| focus() | Sets focus to the current window |
| <u>moveBy()</u> | Moves a window relative to its current position |
| moveTo() | Moves a window to the specified position |
| open() | Opens a new browser window |
| print() | Prints the content of the current window |
| prompt() | Displays a dialog box that prompts the visitor for input |
| resizeBy() | Resizes the window by the specified pixels |
| resizeTo() | Resizes the window to the specified width and height |
| scroll() | This method has been replaced by the <u>scrollTo()</u> method. |
| scrollBy() | Scrolls the content by the specified number of pixels |
| scrollTo() | Scrolls the content to the specified coordinates |
| setInterval() | Calls a function or evaluates an expression at specified intervals (in milliseconds) |
| setTimeout() | Calls a function or evaluates an expression after a specified number of milliseconds |
| stop() | Stops the window from loading |

Document Object:



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

JavaScript **document object** refers to an html document.

Can be created using html **body** tag.

Any other JavaScript object which refers to an html element, can be accessed using **document.elementName** or **document.elements**[] array.

Javascript Document Objects Property

| Name | Description | Version | |
|----------------|--|----------------|--|
| alinkColor | A string that specifies the ALINK attribute. | Implemented in | |
| | | JavaScript 1.0 | |
| anchors | An array containing an entry for each anchor in the | Implemented in | |
| | document. | JavaScript 1.0 | |
| applets | An array containing an entry for each applet in the | Implemented in | |
| | document. | JavaScript 1.0 | |
| <u>bgColor</u> | A string that specifies the BGCOLOR attribute. | Implemented in | |
| | | JavaScript 1.0 | |
| <u>classes</u> | Creates a Style object that can specify the styles of HTML | Implemented in | |
| | tags with a specific CLASS attribute. | JavaScript 1.0 | |
| cookie | Specifies a cookie. | Implemented in | |
| | | JavaScript 1.0 | |
| <u>domain</u> | Specifies the domain name of the server that served a | Implemented in | |
| | document. | JavaScript 1.0 | |
| embeds | An array containing an entry for each plug-in in the | Implemented in | |
| | document. | JavaScript 1.0 | |
| fgColor | A string that specifies the TEXT attribute. | Implemented in | |
| | | JavaScript 1.0 | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| formName | A separate property for each named form in the document. | Implemented in |
|--------------|---|----------------|
| | | JavaScript 1.0 |
| | | |
| <u>forms</u> | An array a containing an entry for each form in the | Implemented in |
| | document. | JavaScript 1.0 |
| height | The height of the document, in pixels. | Implemented in |
| | | JavaScript 1.0 |
| | | |
| ids | Creates a Style object that can specify the style of individual | Implemented in |
| | HTML tags. | JavaScript 1.0 |
| images | An array containing an entry for each image in the | Implemented in |
| | document. | JavaScript 1.0 |
| lastModified | A string that specifies the date the document was last | Implemented in |
| | modified. | JavaScript 1.0 |
| layers | Array containing an entry for each layer within the | Implemented in |
| | document. | JavaScript 1.0 |
| linkColor | A string that specifies the LINK attribute. | Implemented in |
| | | JavaScript 1.0 |
| links | An array containing an entry for each link in the document. | Implemented in |
| | | JavaScript 1.0 |
| plugins | An array containing an entry for each plug-in in the | Implemented in |
| | document. | JavaScript 1.0 |
| referrer | A string that specifies the URL of the calling document. | Implemented in |
| | | JavaScript 1.0 |
| tags | Creates a Style object that can specify the styles of HTML | Implemented in |
| | tags. | JavaScript 1.0 |
| title | A string that specifies the contents of the TITLE tag. | Implemented in |





CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| | | JavaScript 1.0 |
|--------------|--|----------------|
| | A string that specifies the complete URL of a document | Implemented in |
| | A sumplime specifies the complete ORE of a document. | IavaScript 1.0 |
| | | |
| vlinkColor | A string that specifies the VLINK attribute. | Implemented in |
| | | JavaScript 1.0 |
| | | |
| <u>width</u> | The width of the document, in pixels. | Implemented in |
| | | JavaScript 1.0 |
| | | |

Javascript Document Objects Methods

| Name | Description | Version |
|----------------------|---|----------------|
| <u>captureEvents</u> | Sets the document to capture all events of the specified type. | Implemented in |
| | | JavaScript 1.0 |
| close | Closes an output stream and forces data to display. | Implemented in |
| | | JavaScript 1.0 |
| contextual | Uses contextual selection criteria to specify a Style object that | Implemented in |
| | can set the style of individual HTML tags. | JavaScript 1.0 |
| getSelection | Returns a string containing the text of the current selection. | Implemented in |
| | | JavaScript 1.0 |
| <u>handleEvent</u> | Invokes the handler for the specified event. | Implemented in |
| | | JavaScript 1.0 |
| <u>open</u> | Opens a stream to collect the output of write or writeln methods. | Implemented in |
| | | JavaScript 1.0 |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| | - | |
|----------------------|--|----------------|
| <u>releaseEvents</u> | Sets the window or document to release captured events of the | Implemented in |
| | specified type, sending the event to objects further along the | JavaScript 1.0 |
| | event hierarchy. | |
| routeEvent | Passes a captured event along the normal event hierarchy. | Implemented in |
| | | JavaScript 1.0 |
| write | Writes one or more HTML expressions to a document in the | Implemented in |
| | specified window. | JavaScript 1.0 |
| writeln | Writes one or more HTML expressions to a document in the | Implemented in |
| | specified window and follows them with a newline character. | JavaScript 1.0 |
| | | |

Javascript Navigator Objects

Description

Contains information about the version, mimetype and what plug-ins users have installed of Navigator in use. Cannot be created by user. It is automatically created by the javascript runtime engine.

Javascript Navigator Object: Properties

| Name | Description | Version |
|----------------|--|---------------------------|
| | | |
| appCodeName | Specifies the code name of the browser. | Implemented in JavaScript |
| | | 1.0 |
| <u>appName</u> | Specifies the name of the browser. | Implemented in JavaScript |
| | | 1.0 |
| appVersion | Specifies version information for the Navigator. | Implemented in JavaScript |
| | | 1.0 |
| language | Indicates the translation of the Navigator being used. | Implemented in JavaScript |





CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

| | | 1.2 |
|-----------|---|---------------------------|
| | | |
| mimeTypes | An array of all MIME types supported by the client. | Implemented in JavaScript |
| | | 1.1 |
| platform | Indicates the machine type for which the Nevigetor was | Implemented in IsyaScript |
| plation | indicates the machine type for which the Navigator was | Implemented in JavaScript |
| | compiled. | 1.2 |
| | | |
| plugins | An array of all plug-ins currently installed on the client. | Implemented in JavaScript |
| | | 1.1 |
| userAgent | Specifies the user-agent header. | Implemented in JavaScript |
| | | 1.0 |

Javascript Navigator Object : Methods

| Name | Description | Version |
|--------------|---|-------------------------------|
| | | |
| javaEnabled | Tests whether Java is enabled. | Implemented in JavaScript 1.1 |
| taintEnabled | Specifies whether data tainting is enabled. | Implemented in JavaScript 1.1 |
| | | |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: III (JAVA SCRIPT) BATCH-2015-2018

POSSIBLE QUESTIONS

(Section B)

(5*8=40)

- 1. Write short notes on function in JavaScript.
- 2. Explain the purpose and properties of the window object in JavaScript.
- 3. Write about JavaScript Math Object.
- 4. Discuss about statements, blocks, and comments in JavaScript
- 5. Explain usage of date object in JavaScript with an example program.
- 6. Write a JavaScript program which submits and resets input in a form.
- 7. Write about statements in Javascript with syntax and examples.
- 8. Write a Javascript program for factorial calculation using a function.
- 9. Write short notes on functions in JavaScript.
- 10. Explain the various events and how to handle the events in JavaScript.



DEPARTMENT OF COMPUTER SCIENCE, CA & IT

III B.Sc CS (Batch 2015-2018)

WEB TECHNOLOGY

PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS

ONLINE EXAMINATIONS

UNIT-3

| S.No | Questions | opt1 | opt2 | opt3 | opt4 | Answer |
|------|---|---------------------------|-------------------|------|------|--------|
| | Inside which HTML element do we put the | | | | | |
| | JavaScript? | | | | | |
| 1 | | <javascript></javascript> | <script></script> | | | |

| 1 | | | | | | |
|----|---|-----------------------|--|---------------|------------------|----------------|
| | How do you call a function named | | call | call function | called | |
| 7 | "myFunction"? | myFunction() | myFunction() | myFunction | myFunction() | myFunction() |
| | How do you write a conditional statement for | | | | | |
| | executing some statements only if "i" is equal | | | | | |
| 8 | to 5? | if (i==5) | if i=5 | if i==5 then | if i=5 then | if (i==5) |
| | | | | | | |
| | | 'This is a | This is a</td <td>//This is a</td> <td>??this is a</td> <td>//This is a</td> | //This is a | ??this is a | //This is a |
| 9 | How can you add a comment in a JavaScript? | comment | comment> | comment | comment | comment |
| | | /*This comment | //This comment | comment has | //* this comment | /*This comment |
| | What is the correct JavaScript syntax to insert a | has more than | has more than | more than one | has more than | has more than |
| 10 | comment that has more than one line? | one line*/ | one line// | line> | one line*// | one line*/ |
| | tag is used to specify | | | | | |
| | alternate content for browsers that do not | | | | | |
| 11 | support Javascript | <noscript></noscript> | <script></script> | | | |

| | event occur when user removes focus | | | | | |
|----|--|--------------|----------|--------------------------|--------------|--------------------------|
| 16 | from window of form element. | Abort | blur | Change | Error | blur |
| 10 | | | | | | |
| | event occurs user presses or holds | | | | | |
| 17 | down a key | Key Down | KevPress | KeyUn | Arrays | KevPress |
| 17 | | | | neyep | liiuys | |
| | event occur, when user loads the | | | | | |
| 18 | event occur when user loads the | Load | Click | Abort | Key down | Load |
| 10 | | Loud | | | | Loud |
| | is a collection of properties and | | | | | |
| 10 | is a concertor of properties and methods, all classes under a single name | Objects | Class | Data | Load | Objects |
| 1) | inculous an classes under a single name. | Objects | C1035 | | Load | |
| | is a method used to aloge the | | | | | |
| 20 | Is a method used to close the | Close() | () | Foous | \mathbf{D} | Class |
| 20 | specified willdow. | Close() | open() | rocus() | Diui() | |
| | | | | | | |
| 01 | is a method which gives a focus | Class | | F ₁ () | Dlaser | F ₂ () |
| 21 | to a window. | Close() | open() | Focus() | Blur() | Focus() |
| | function evaluates a string of | | | | | |
| | javascript code without reference to a particular | T C 1 | | | | |
| 22 | object. | Is finite | eval | 1s Nan | 1sNaN | eval |
| | | | | | | |
| | function evaluates an argument to | | | | | |
| 23 | determine whether it is a finite bumber. | isFinite | eval | is Nan | isinfinite | isFinite |
| | | | | | | |
| | function evaluates an argument to | | | | | |
| 24 | determine if it is not a number | isFinite | eval | isNaN | is infinite | isNaN |

| | can only be accessed within that | | | | | |
|----|---|--------------------|----------------|-----------------|------------------|-----------------------|
| 25 | function | variable | local variable | global Variable | private varaible | local variable |
| 26 | are the variables can be accessed outside of all functions. | local variable | variable | global Variable | private varaible | global Variable |
| 27 | is an Object Oriented Programming | JavaScript | c | cobol | fortron | JavaScript |
| 28 | are the values associated with an object. | Attributes | Functions | Properties | methods | Properties |
| 29 | are the actions that can be performed on objects. | Methods | Functions | Attributes | Properties | Methods |
| 30 | is used to manipulate a stored piece of text. | Math Object | String object | Boolean Object | date object | String object |
| 31 | is used to work with dates and times. | Date object | Math Object | String object | Boolean Object | Date object |
| 32 | is used to store a set of values in a single variable name. | Array object | Math Object | String object | Boolean Object | Array object |
| 33 | is used to convert a non-Boolean value to a Boolean value (true or false). | The Boolean object | Math Object | String object | Array object | The Boolean object |

| | allows you to perform common | | The Boolean | | | |
|----|---|--------------|-------------|---------------|---------------|--------------|
| 34 | mathematical tasks. | Math object | object | Date object | String object | Math object |
| | | | | | | |
| | is the method which is used to | | | | | |
| 35 | return a random number between 0 and 1. | random() | max() | any() | rand() | random() |
| | is the method used to return the number | | | | | |
| | with the highest value of two specified | | | | | |
| 36 | numbers. | max() | min() | maximum() | minmium() | max() |
| | is the method used to return the | | | | | |
| | number with the lowest value of two specified | | | | | |
| 37 | numbers. | max() | min() | maximum() | minmium() | min() |
| | | | | | | |
| | is often used if you want to make | | | | | |
| 38 | sure information comes through to the user. | An alert box | confirm Box | Prompt box | command box | An alert box |
| | | | | - | | |
| | is often used if you want the user to verify | | | | | |
| 39 | or accept something. | An alert box | confirm Box | Prompt box | command box | confirm Box |
| | | | | 1 | | |
| | is often used if you want the user to | | | | | |
| 40 | input a value before entering a page. | An alert box | confirm Box | Prompt box | command box | Prompt box |
| | | | | 1 | | 1 |
| | When an alert box pops up, the user will have | | | | | |
| 41 | to click to proceed | ОК | Cancel | Ok and Cancel | reset | ОК |
| | ······································ | | | | | |
| | boolean operator is used to perform | | | | | |
| 42 | logical exclusion on two expressions | NOT | XOR | OR | AND | XOR |

| 43 | is used to divide a number and raise an integer result | \ | ٨ | * | / | / |
|----|---|-------|-------|-------|-------|-------|
| 44 | returns the absolute value of a number | asb | abs | value | ceil | abs |
| 45 | returns a string that has been converted to lowercase | LCASE | LOWER | UPPER | UCASE | LCASE |



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

UNIT-IV

PHP: Introduction-What is PHP?-Why PHP?-Basic PHP syntax-Comments in PHP-PHP5 varaiables-PHP5 echo and Print Statements- How Online PHP Programs Runs- PHP Datatypes-PHP String Functions

Introduction

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Common uses of PHP

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.





CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible -

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

"Hello World" Script in PHP

To get a feel for PHP, first start with simple PHP scripts. Since "Hello, World!" is an essential example, first we will create a friendly little "Hello, World!" script.

As mentioned earlier, PHP is embedded in HTML. That means that in amongst your normal HTML (or XHTML if you're cutting-edge) you'll have PHP statements like this –

<html> <head> <title>Hello World</title> </head> <body> <?php echo "Hello, World!";?> </body>

</html>

It will produce following result -

Hello, World!

If you examine the HTML output of the above example, you'll notice that the PHP code is not present in the file sent from the server to your Web browser. All of the PHP present in the Web page is processed and stripped from the page; the only thing returned to the client from the Web server is pure HTML output.



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

All PHP code must be included inside one of the three special markup tags ATE are recognised by the PHP Parser.

<?php PHP code goes here ?>

<? PHP code goes here ?>

<scriptlanguage="php"> PHP code goes here </script>

Basic PHP syntax

Escaping to PHP

A PHP script can be placed anywhere in the document.

A PHP script starts with **<?php** and ends with **?>**:

```
<?php
// PHP code goes here
?>
```

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

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

Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

```
Example
<!DOCTYPE html>
<html>
<body>
```

<h1>My first PHP page</h1>

```
<?php
echo "Hello World!";
```



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

?>

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

The PHP parsing engine needs a way to differentiate PHP code from other elements in the page. The mechanism for doing so is known as 'escaping to PHP'. There are four ways to do this –

Canonical PHP tags The most universally effective PHP tag style is –

<?php...?>

If you use this style, you can be positive that your tags will always be correctly interpreted.

Short-open (SGML-style) tags

Short or short-open tags look like this -

<?...?>

Short tags are, as one might expect, the shortest option You must do one of two things to enable PHP to recognize the tags –

- Choose the --enable-short-tags configuration option when you're building PHP.
- Set the short_open_tag setting in your php.ini file to on. This option must be disabled to parse XML with PHP because the same syntax is used for XML tags.

ASP-style tags

ASP-style tags mimic the tags used by Active Server Pages to delineate code blocks. ASP-style tags look like this –

<%...%>

To use ASP-style tags, you will need to set the configuration option in your php.ini file.

HTML script tags HTML script tags look like this –

<script language = "PHP">...</script>

Comments in PHP Code

A *comment* is the portion of a program that exists only for the human reader and stripped out before displaying the programs result. There are two commenting formats in PHP -

Prepared By K.Kathirvel & K.Yuvaraj, Department Of Computer Science, CA & IT, KAHE Page 4\25



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

Single-line comments – They are generally used for short explanations or notes relevant to the local code. Here are the examples of single line comments.

```
<?
```

This is a comment, and

This is the second line of the comment

// This is a comment too. Each style comments only

print"An example with single line comments";

?>

Multi-lines printing - Here are the examples to print multiple lines in a single print statement -

<?

First Example

print<<<END

This uses the "here document" syntax to output

multiple lines with \$variable interpolation.Note

that the here document terminator must appear on a

line with just a semicolon no extra whitespace!

END;

Second Example
print"This spans
 multiple lines. The newlines will be
 output as well";
?>

Multi-lines comments – They are generally used to provide pseudocode algorithms and more detailed explanations when necessary. The multiline style of commenting is the same as in C. Here are the example of multi lines comments.



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

```
<?

/* This is a comment with multiline

Author : Mohammad Mohtashim

Purpose: Multiline Comments Demo

Subject: PHP

*/

print"An example with multi line comments";
```

?>

PHP is whitespace insensitive

Whitespace is the stuff you type that is typically invisible on the screen, including spaces, tabs, and carriage returns (end-of-line characters).

PHP whitespace insensitive means that it almost never matters how many whitespace characters you have in a row.one whitespace character is the same as many such characters.

For example, each of the following PHP statements that assigns the sum of 2 + 2 to the variable \$four is equivalent –

\$four =2+2;// single spaces
\$four <tab>=<tab2<tab>+<tab>2;// spaces and tabs

\$four =

2+

2;// multiple lines

PHP is case sensitive

Yeah it is true that PHP is a case sensitive language. Try out following example -

<html>

<body>



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

<?php

\$capital =67;

print("Variable capital is \$capital
");

print("Variable CaPiTaL is \$CaPiTaL
");

?>

</body>

</html>

This will produce the following result -

Variable capital is 67 Variable CaPiTaL is

Statements are expressions terminated by semicolons

A *statement* in PHP is any expression that is followed by a semicolon (;). Any sequence of valid PHP statements that is enclosed by the PHP tags is a valid PHP program. Here is a typical statement in PHP, which in this case assigns a string of characters to a variable called greeting -

\$greeting = "Welcome to PHP!";

Expressions are combinations of tokens

The smallest building blocks of PHP are the indivisible tokens, such as numbers (3.14159), strings (.two.), variables (\$two), constants (TRUE), and the special words that make up the syntax of PHP itself like if, else, while, for and so forth

Braces make blocks

Although statements cannot be combined like expressions, you can always put a sequence of statements anywhere a statement can go by enclosing them in a set of curly braces.

Here both statements are equivalent -

```
if(3==2+1)
print("Good - I haven't totally lost my mind.<br>");
```

if(3==2+1){

print("Good - I haven't totally");



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

print("lost my mind.
");

Running PHP Script from Command Prompt

Yes you can run your PHP script on your command prompt. Assuming you have following content in test.php file

Live Demo

<?php

}

```
echo "Hello PHP!!!!!";
```

?>

Now run this script as command prompt as follows -

\$ php test.php

It will produce the following result -

Hello PHP!!!!!

Hope now you have basic knowledge of PHP Syntax.

PHP 5 Variables

The main way to store information in the middle of a PHP program is by using a variable.

Here are the most important things to know about variables in PHP.

- All variables in PHP are denoted with a leading dollar sign (\$).
- The value of a variable is the value of its most recent assignment.
- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.
- Variables in PHP do not have intrinsic types a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP does a good job of automatically converting types from one to another when necessary.
- PHP variables are Perl-like.

PHP has a total of eight data types which we use to construct our variables -



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

- Integers are whole numbers, without a decimal point, like 4195.
- **Doubles** are floating-point numbers, like 3.14159 or 49.1.
- **Booleans** have only two possible values either true or false.
- NULL is a special type that only has one value: NULL.
- Strings are sequences of characters, like 'PHP supports string operations.'
- Arrays are named and indexed collections of other values.
- **Objects** are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources** are special variables that hold references to resources external to PHP (such as database connections).

The first five are *simple types*, and the next two (arrays and objects) are compound - the compound types can package up other arbitrary values of arbitrary type, whereas the simple types cannot.

We will explain only simple data type in this chapters. Array and Objects will be explained separately.

Integers

They are whole numbers, without a decimal point, like 4195. They are the simplest type .they correspond to simple whole numbers, both positive and negative. Integers can be assigned to variables, or they can be used in expressions, like so -

```
$int_var = 12345;
$another_int = -12345 + 12345;
```

Integer can be in decimal (base 10), octal (base 8), and hexadecimal (base 16) format. Decimal format is the default, octal integers are specified with a leading 0, and hexadecimals have a leading 0x.

For most common platforms, the largest integer is $(2^{**}31 \cdot 1)$ (or 2,147,483,647), and the smallest (most negative) integer is . $(2^{**}31 \cdot 1)$ (or .2,147,483,647).

Doubles

They like 3.14159 or 49.1. By default, doubles print with the minimum number of decimal places needed. For example, the code -

<?php

\$many =2.2888800; \$many_2 =2.2111200; \$few = \$many + \$many_2;



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

print("\$many + \$many_2 = \$few
");

```
?>
```

It produces the following browser output -

2.28888 + 2.21112 = 4.5

Boolean

They have only two possible values either true or false. PHP provides a couple of constants especially for use as Booleans: TRUE and FALSE, which can be used like so –

if(TRUE)
print("This will always print
");

else

print("This will never print
");

Interpreting other types as Booleans

Here are the rules for determine the "truth" of any value not already of the Boolean type -

- If the value is a number, it is false if exactly equal to zero and true otherwise.
- If the value is a string, it is false if the string is empty (has zero characters) or is the string "0", and is true otherwise.
- Values of type NULL are always false.
- If the value is an array, it is false if it contains no other values, and it is true otherwise. For an object, containing a value means having a member variable that has been assigned a value.
- Valid resources are true (although some functions that return resources when they are successful will return FALSE when unsuccessful).
- Don't use double as Booleans.

Each of the following variables has the truth value embedded in its name when it is used in a Boolean context.

\$true_num =3+0.14159;

\$true_str ="Tried and true"

\$true_array[49]="An array element";

Loter (Form) (Loter KARPAGAM ACADEMY OF HIGHER EDUCATION Demonst the University Technological (MEY the 1997)

KARPAGAMACADEMY OF HIGHER EDUCATION

CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

\$false_array = array();

\$false_null = NULL;

\$false_num =999-999;

\$false_str ="";

NULL

NULL is a special type that only has one value: NULL. To give a variable the NULL value, simply assign it like this -

```
$my_var = NULL;
```

The special constant NULL is capitalized by convention, but actually it is case insensitive; you could just as well have typed –

\$my_var =null;

A variable that has been assigned NULL has the following properties -

- It evaluates to FALSE in a Boolean context.
- It returns FALSE when tested with IsSet() function.

Strings

They are sequences of characters, like "PHP supports string operations". Following are valid examples of string

\$string_1 ="This is a string in double quotes";

\$string_2 ='This is a somewhat longer, singly quoted string';

\$string_39 ="This string has thirty-nine characters";

\$string_0 ="";// a string with zero characters

Singly quoted strings are treated almost literally, whereas doubly quoted strings replace variables with their values as well as specially interpreting certain character sequences.

<?php

\$variable ="name";

\$literally ='My \$variable will not print!';





CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

```
print($literally);
print"<br>";
$literally ="My $variable will print!";
print($literally);
?>
```

This will produce following result -

My \$variable will not print! My name will print

There are no artificial limits on string length - within the bounds of available memory, you ought to be able to make arbitrarily long strings.

Strings that are delimited by double quotes (as in "this") are preprocessed in both the following two ways by PHP -

- Certain character sequences beginning with backslash (\) are replaced with special characters
- Variable names (starting with \$) are replaced with string representations of their values.

The escape-sequence replacements are -

- \n is replaced by the newline character
- \r is replaced by the carriage-return character
- \t is replaced by the tab character
- \\$ is replaced by the dollar sign itself (\$)
- \" is replaced by a single double-quote (")
- \\ is replaced by a single backslash (\)

Here Document

You can assign multiple lines to a single string variable using here document -

```
<?php
$channel =<<<_XML_
<channel>
<title>What's For Dinner</title>
<link>http://menu.example.com/ </link>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

<description>Choose what to eat tonight.</description>

</channel>

XML;

echo <<<END

This uses the "here document" syntax to output multiple lines with variable

interpolation. Note that the here document terminator must appear on a line with

just a semicolon. no extra whitespace!

END;

print \$channel;

?>

This will produce following result -

This uses the "here document" syntax to output multiple lines with variable interpolation. Note that the here document terminator must appear on a line with just a semicolon. no extra whitespace!

<channel> <title>What's For Dinner<title> <link>http://menu.example.com/<link> <description>Choose what to eat tonight.</description>

Variable Scope

Scope can be defined as the range of availability a variable has to the program in which it is declared. PHP variables can be one of four scope types -

- Local variables
- Function parameters
- <u>Global variables</u>
- <u>Static variables</u>

Variable Naming



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

Rules for naming a variable is -

- Variable names must begin with a letter or underscore character.
- A variable name can consist of numbers, letters, underscores but you cannot use characters like + , , % , (,) . & , etc

There is no size limit for variables.

PHP5 echo and Print Statements

The PHP echo Statement

The echo statement can output one or more strings. In general terms, the echo statement can display anything that can be displayed to the browser, such as string, numbers, variables values, the results of expressions etc.

Since echo is a language construct not actually a function (like <u>if</u> statement), you can use it without parentheses e.g. <u>echo</u> or <u>echo()</u>. However, if you want to pass more than one parameter to echo, the parameters must not be enclosed within parentheses.

Display Strings of Text

The following example will show you how to display a string of text with the echo statement:

Example

```
<?php
// Displaying string of text
echo"Hello World!";
```

?>

The output of the above PHP code will look something like this: Hello World!

Display HTML Code

The following example will show you how to display HTML code using the echo statement:

Example

```
<?php
// Displaying HTML code
echo"<h4>This is a simple heading.</h4>";
echo"<h4 style='color: red;'>This is heading with style.</h4>"
?>
```

The output of the above PHP code will look something like this: *This is a simple heading.*



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

This is heading with style.

Display Variables The following example will show you how to display variable using the echo statement:

Example

```
<?php
// Defining variables
$txt = "Hello World!";
$num = 123456789;
$colors = array("Red", "Green", "Blue");
```

```
// Displaying variables
echo$txt;
echo"<br>";
echo$num;
echo"<br>";
echo$colors[0];
```

?>

The output of the above PHP code will look something like this: Hello World! 123456789 Red

The PHP print Statement

You can also use the print statement (an alternative to echo) to display output to the browser. Like echo the print is also a language construct not a real function. So you can also use it without parentheses like: print or print().

Both echo and print statement works exactly the same way except that the print statement can only output one string, and always returns 1. That's why the echo statement considered marginally faster than the print statement since it doesn't return any value.

Display Strings of Text

The following example will show you how to display a string of text with the print statement:

Example

<?php // Displaying string of text print"Hello World!"; ?>


CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

The output of the above PHP code will look something like this: Hello World!

Display HTML Code

The following example will show you how to display HTML code using the print statement:

| Example |
|--|
| php</td |
| // Displaying HTML code |
| print" <h4>This is a simple heading.</h4> "; |
| print" <h4 style="color: red;">This is heading with style.</h4> " |
| ?> |
| The output of the above PHP code will look something like this: |
| This is a simple heading. |
| This is heading with style. |
| Display Variables |
| The following example will show you how to display variable using the print statement: |
| Example Run this code » |
| php</th |
| // Defining variables |
| <pre>\$txt = "Hello World!";</pre> |
| \$num = 123456789; |
| <pre>\$colors = array("Red", "Green", "Blue");</pre> |
| |
| // Displaying variables |
| print\$txt; |
| print" "; |
| print\$num; |
| print" "; |
| print\$colors[0]; |

?>

The output of the above PHP code will look something like this: Hello World! 123456789 Red

How Online PHP Programs Runs

If you double click on a HTML file (files with **.html** or **.htm** extension), it would open on your web browser. But same won't happen if you double clicked on a PHP file (probably it would open in an <u>editor</u>).



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

The reason is <u>PHP files first need be processed</u> in a web server before sending their output to the web browser.

Therefore before running PHP files, they should be placed inside the web folder of a web server and then make a request to desired PHP file by typing its URL in the web browser. If you installed a web server in your computer, usually the root of its web folder can be accessed by typing **http://localhost** in the web browser. So, if you placed a file called **hello.php** inside its web folder, you can run that file by calling **http://localhost/hello.php**.

Web folder can be changed based on your web host (if you hosted your web site online) or the method you installed the web server in your computer. <u>If you used XAMPP</u> to install Apache (web server) in your computer then the web folder would be **htdocs** which is under the root directory of XAMPP.

Example

<html> <head> <title>Online PHP Script Execution</title> </head> <body> <?php echo "<h1>Hello, PHP!</h1>\n"; ?> </body> </html> eta Tunos

PHP Data Types

Variables can store data of different types, and different data types can do different things.

PHP supports the following data types:

- String
- Integer
- Float (floating point numbers also called double)
- Boolean
- Array
- Object
- NULL
- Resource



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

The values assigned to a PHP variable may be of different data types including simple string and numeric types to more complex data types like arrays and objects.

PHP supports total eight primitive data types: Integer, Floating point number or Float, String, Booleans, Array, Object, resource and NULL. These data types are used to construct variables. Now let's discuss each one of them in detail.

PHP Integers

Integers are whole numbers, without a decimal point (..., -2, -1, 0, 1, 2, ...). Integers can be specified in decimal (base 10), hexadecimal (base 16 - prefixed with 0x) or octal (base 8 - prefixed with 0) notation, optionally preceded by a sign (- or +).

Example <u>Run this code</u> »

<?php

\$a=123; // decimal number
var_dump(\$a);
echo"
";

\$b=-123; // a negative number
var_dump(\$b);
echo"
";

```
$c=0x1A; // hexadecimal number
var_dump($c);
echo"<br>";
```

\$d=0123; // octal number
var_dump(\$d);
?>

Note:Since PHP 5.4+ you can also specify integers in binary (base 2) notation. To use binary notation precede the number with 0b (e.g. \$var = 0b1111111;).

PHP Strings

Strings are sequences of characters, where every character is the same as a byte.

A string can hold letters, numbers, and special characters and it can be as large as up to 2GB (2147483647 bytes maximum). The simplest way to specify a string is to enclose it in single quotes (e.g. 'Hello world!'), however you can also use double quotes ("Hello world!").



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

| <i>Example</i> <u>Run this code »</u> | |
|--|--|
| php<br \$a='Hello world!'; echo\$a; echo" "; | |
| <pre>\$b="Hello world!"; echo\$b; echo" ";</pre> | |
| <pre>\$c='Stay here, I\'ll be back.'; echo\$c; ?></pre> | |

PHP Floating Point Numbers or Doubles

Floating point numbers (also known as "floats", "doubles", or "real numbers") are decimal or fractional numbers, like demonstrated in the example below.

| Example | |
|---------|--|
| | |

```
<?php
$a=1.234;
var_dump($a);
echo"<br>";
$b=10.2e3;
var_dump($b);
echo"<br>";
$c=4E-10;
var_dump($c);
```

?>

PHP Booleans

Booleans are like a switch it has only two possible values either 1 (true) or 0 (false).

Example

<?php



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

// Assign the value TRUE to a variable
\$show_error=true;
var_dump(\$show_error);
?>

PHP Arrays

An array is a variable that can hold more than one value at a time. It is useful to aggregate a series of related items together, for example a set of country or city names.

An array is formally defined as an indexed collection of data values. Each index (also known as the key) of an array is unique and references a corresponding value.

Example

```
<?php
$colors=array("Red", "Green", "Blue");
var_dump($colors);
echo"<br>";
$color_codes=array(
"Red"=>"#ff0000",
"Green"=>"#00ff00",
"Blue"=>"#0000ff"
);
var_dump($color_codes);
?>
```

PHP Objects

An object is a data type that not only allows storing data but also information on, how to process that data. An object is a specific instance of a class which serve as templates for objects. Objects are created based on this template via the new keyword.

Every object has properties and methods corresponding to those of its parent class. Every object instance is completely independent, with its own properties and methods, and can thus be manipulated independently of other objects of the same class.

Here's a simple example of a class definition followed by the object creation.

```
Example
<?php
// Class definition
classgreeting{
// properties
public$str="Hello World!";
```

```
// methods
functionshow_greeting(){
return$this->str;
```



} }

KARPAGAMACADEMY OF HIGHER EDUCATION

CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) **BATCH-2015-2018**

```
// Create object from class
$message=newgreeting;
var dump($message);
?>
```

Tip: The data elements stored within an object are referred to as its properties and the information, or code which describing how to process the data is called the methods of the object.

PHP NULL

The special NULL value is used to represent empty variables in PHP. A variable of type NULL is a variable without any data. NULL is the only possible value of type null.

| Елитри |
|--------|
|--------|

<?php \$a=NULL; var dump(a); echo"
";

```
$b="Hello World!";
$b=NULL;
var_dump($b);
?>
```

When a variable is created without a value in PHP like *\$var*; it is automatically assigned a value of null. Many novice PHP developers mistakenly considered both var1 = NULL; and var2 = ""; are same, but this is not true. Both variables are different — the *\$var1* has null value while *\$var2* indicates no value assigned to it.

PHP Resources

A resource is a special variable, holding a reference to an external resource. Resource variables typically hold special handlers to opened files and database connections.

Example Run this code »

<?php // Open a file for reading \$handle=fopen("note.txt", "r");

> Prepared By K.Kathirvel & K.Yuvaraj, Department Of Computer Science, CA & IT, KAHE Page 21\25



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

var_dump(\$handle); echo"
";

// Connect to MySQL database server with default setting
\$link=mysql_connect("localhost", "root", "");
var_dump(\$link);
?>

PHP string functions are the part of the core. There is no installation required to use this function

PHP String Functions

PHP – indicates the earliest version of PHP that supports the function.

| Sr.No | Function & Description | PHP |
|-------|---|-----|
| 1 | addcslashes It returns the string with blackslashes | 4 |
| 2 | addslashes It returns the string with blackslashes in front of predefined characters | 4 |
| 3 | bin2hex It is used to convert primary data to hexadecimal representation | 4 |
| 4 | chop It is used to removes whitespace | 4 |
| 5 | chr It returns the specific characters | 4 |
| 6 | chunk split It is used to split a string into chunks. | 5 |
| 7 | convert cyr string It is used to convert from one Cyrillic character set to another | 4 |
| 8 | convert uudecode It is used to decode a encoded string | 5 |
| 9 | count chars It is used to returns the information about character used in a string | 4 |
| 10 | Crc32 It is used to calculates 32-bit CRC | 4 |



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

| 11 | crypt It is used to hashing the string | 4 |
|----|---|-------|
| 12 | Echo It is give the output as one or more string | 4 |
| 13 | explode It is used to split a string by string | 4 |
| 14 | fprintf It is used to write a formatted string to a stream | 5 |
| 15 | get html translation table It returns the translation table used by htmlspecialchars() and htmlentities() | 4 |
| 16 | hebrev It is used to convert logical Hebrew text to visual text | 4 |
| 17 | Hebrevc It is used to convert logical Hebrew text to visual text with newline conversion | 4 |
| 18 | hex2bin It is used to convert a string of hexadecimal to ASCII character | 4 |
| 19 | html entity decode It is used to convert HTML entities to their application characters | 4 |
| 20 | htmlentities It is used to convert all applicable characters to Html entities | 5.4 |
| 21 | html special chars decode It is used to convert convert special HTML entities back to characters. | 5.1.0 |
| 22 | htmlspecialchars It is used to convert special characters to HTML entities | 4 |
| 23 | implode It is used to Join array elements with a string. | 5 |
| 24 | join It is alias of implode(), it returns string from the elements of an array | 4 |
| 25 | lcfirst It is used to make a string's first character should be lowercase. | 5.3.0 |



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

| 26 | levenshtein It is used calculate Levenshtein distance between two strings | 4.0.1 |
|----|--|-------|
| 27 | localeconv It is used to get numeric formatting information | 4 |
| 28 | ltrim It used to strip whitespace or other characters from the beginning of a string | 4 |
| 29 | md5_file It is used to calculates the md5 hash of a given file | 4 |
| 30 | md5 It is used to calculates the md5 hash of a string | 4 |
| 31 | metaphone It is used to calculates the metaphone key of a string | 4 |
| 32 | money format It is used to formats a number as a currency string | 4.3.0 |
| 33 | nl langinfo It has contained information about language and locale | 4.3.0 |



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: IV (PHP INTRODUCTION) BATCH-2015-2018

POSSIBLE QUESTIONS

(Section B)

(5*8=40)

- 1. Explain the need for PHP and its basic syntax.
- 2. Write a session application to count the number of times a user has accessed a web page.
- 3. Illustrate loops and arrays in PHP with an example.
- 4. Write a program in PHP to reverse a string.
- 5. Explain branching statements with example.
- 6. Enumerate at least 8 string functions in PHP and explain their usage.
- 7. Explain the use of operators in PHP.
- 8. How Online PHP Programs Runs? Explain the steps involved.
- 9. Discuss in detail about declaring variables and data types in PHP?
- 10. Write a detail account on echo and print statements in PHP.

| (Deemed to be University) (Established Under Section 3 of UGC Act, 1956) |
|--|

KARPAGAM ACADEMY OF HIGHER EDUCATION DEPARTMENT OF COMPUTER SCIENCE, CA & IT

III B.Sc CS (Batch 2015-2018)

WEB TECHNOLOGY

PART - A OBJECTIVE TYPE/MULTIPLE CHOICE QUESTIONS

ONLINE EXAMINATIONS

UNIT-4

| Sno | Questions | opt1 | opt2 | opt3 | opt4 | Answer | | |
|-----|--|---------------|-----------------|-----------------|---------------|-----------------|--|--|
| | | Php hypertext | Php html | Php hypertext | Php html | Php hypertext | | |
| 1 | Php stands for | | | | | | | |
| | | processor | preprocessor | preprocessor | processor | preprocessor | | |
| | | | | | | | | |
| 2 | Php is scripting language | Server-side | Client –side | Middle-side | Outside | Server-side | | |
| | | | | | | | | |
| | | | | | It depends on | | | |
| 3 | Php script are executed in | ISP computer | Client computer | Server computer | | Server computer | | |
| | | | | | php script | | | |
| | | <php></php> | | | | | | |
| 4 | Php script starts with | | php? | ?php?php | | php? | | |
| | | | | | | | | |
| | Which of the following statements prints | | | | | | | |
| 5 | | Out | Write | echo | display | echo | | |
| | in php? | | | | | | | |
| | | | | | | | | |
| 6 | in php each statement should end with | .(dot) | ;(semicolon) | /(slash) | : (colon) | ;(semicolon) | | |
| | | | | | | | | |

| | In php language the variable name starts | | | | | |
|----|---|----------------|---------------------|------------------|--------------------|------------------------------|
| 7 | | ! | & | * | \$ | \$ |
| | with | | | | | |
| | In php the variable names are case | | | Depends on the | Depends on the | |
| 8 | | TRUE | FALSE | | | TRUE |
| | sensitive | | | server | website | |
| | In php variable should be declared | | | Depends on the | Depends on the | |
| 9 | | TRUE | FALSE | | | FALSE |
| | before assigning value | | | server | website | |
| | Which of the following is not a scope for | | | | | |
| 10 | | Local | Global | Static | Extern | Extern |
| | a variable in php? | | | | | |
| | what is the use of the function strlen() in | It returns the | It returns the type | It returns the | It returns the | It returns the length of the |
| 11 | | type of the | | length of the | subset of the | |
| | php? | string | of the string | string | string | string |
| | Which of the following operator | 8 | | ~8 | | 8 |
| 12 | | +(plus) | .(dot) | &(ampersand) | %(percentage) | .(dot) |
| | concatenates the string in php? | - | | | | |
| | Which of the following is not a php | | | | | |
| 13 | | While | Do-while | For | Do-until | Do-until |
| | loop? | | | | | |
| | what is the use of the function strpos() in | Search for a | Search for space | Search for a | Search for text in | Search for a character in |
| 14 | | number in the | | character in the | | |
| | php? | string | in the string | string | the string | the string |
| | | Lines are | Variable | Function | <i>Q</i> | 6 |
| 15 | What is the use of # symbol in php? | | | | No uses in php. | Lines are commented |
| | | commented | declaration | declaration | | |

| | | <pre>\$variable_name</pre> | <pre>\$variable_name=</pre> | <pre>\$variable_name=</pre> | <pre>\$variable_name</pre> | |
|----|--|----------------------------|-----------------------------|-----------------------------|----------------------------|-----------------------------------|
| 16 | How to define a variable in php? | | | | | <pre>\$variable_name=value;</pre> |
| | | | walwar | | | |
| | | =value | value; | =value; | as value; | |
| | | Compares | Compares strings | Compares strings | Compares strings | Compares strings |
| 17 | Use of strcmp() function in php? | strings | | | | |
| | | including case | aveluding case | in uppercase | lowercase | including case |
| | When string with combined with other | including case | excluding case | | lowerease | including case |
| | when string with combined with other | | | | | |
| 18 | | Int | Float | String | Double | String |
| | datatype in php the result is | | | | | |
| | | | | | | |
| 10 | | TT | τ1 | C | C (| τ1 |
| 19 | Php is typed language. | User | Loosely | Server | Strongly | Loosely |
| | | | | | | |
| | Which of the following functions do not | | | | | |
| 20 | | time() | date() | strtotime() | localtime() | localtime() |
| 20 | | time() | dute() | striounie() | ioeutime() | locultine() |
| | return a timestamp? | | | | | |
| | | | A floating-point | | | |
| 21 | The getdate() function returns | An integer | | An array | A string | An array |
| | | - | numbor | - | - | |
| | Returns the time of suprise for a | | nullioei | | | |
| | | | | | | |
| 22 | | datesunrise() | date_sunrise() | date-sunrise() | date.sunrise() | date_sunrise() |
| | given day / location. | | | | | |
| | What will the following script | | | | | |
| 22 | output? php \$time = strtotime</td <td>00.00.00</td> <td>12.00.00</td> <td>00.00</td> <td>12::00</td> <td>00.00</td> | 00.00.00 | 12.00.00 | 00.00 | 12::00 | 00.00 |
| 23 | ('2004/01/01'); echo date'H:\i:s', \$time); | 00:00:00 | 12:00:00 | 00:1:00 | 12:1:00 | 00:1:00 |
| | ?> | | | | | |
| | Checks a date for numeric | | | | | |
| 24 | | check date | verifydate | Verify date | Checkdate | Checkdate |
| | | | · erri j dute | · emp_auto | Choondate | Checkado |
| | validity. | | | | | |

| | What is the difference, in seconds, | It depends on | There is no | The two will only | The two will | |
|----|--|-----------------|-------------------|-----------------------|-------------------|---------------------------|
| 25 | between the current timestamp in the | the number of | | match if the local | | There is no difference |
| 23 | GMT time zone and the current | hours between | | materi il tile local | | |
| | timestamp in vour local time zone? | the local time | difference | time zone is GMT | never match | |
| | You must make a call to to | | | | | |
| 26 | specify what time zone you want | date default ti | datedefault_timez | date defaulttimez | date default tim | date default timezone set |
| 20 | calculations to take place in before | dute_defuuit_f | datederauit_timez | dute_defuulttimez | date_defautt_till | dute_defautt_timezone_set |
| | calling any date functions. | mezone_set() | one_set() | one_set() | ezoneset() | 0 |
| | What would happen if the following | It would output | It would output | It would output | It would raise an | It would output the |
| 27 | script were run on a Windows server set | | | | | |
| 27 | to Moscow, Russia's time zone? php</td <td></td> <td></td> <td></td> <td></td> <td></td> | | | | | |
| | echo gmmktime(0. 0. 0. 1. 1. 1970): ?> | the number 0 | the number -1 | the number 1 | error | number -1 |
| | The function parses an English | | | | | |
| 28 | textual date or time into a Unix | strtodate() | stroftime() | strtotime() | str_to_time() | strtotime() |
| | timestamp | | | | | |
| | Formats a local time or date | | | | | |
| 20 | | Strftimo | Stratimo | Strhtimo | Stritimo | Strftime |
| 29 | | Sutuite | Suguine | Summe | Sultime | Sultime |
| | according to locale settings. | | | | | |
| | | | | | | |
| 30 | Trace the odd data type | floats | integer | doubles | Real number | integer |
| 50 | Thee the out data type | nouts | integer | doubles | iteur number | integer |
| | | | | | | |
| | Which of the following are valid float | | | | | |
| 31 | | 4.5678 | 4.0 | 7e4 | all the answers | all the answers |
| | walwas? | | | | | |
| | values? | 4-1:: to d here | 1 - 1:: (1 h | dell'anite d'han a ca | | |
| | | definited by | definited by | definited by <<< | | |
| 32 | In php string data are | | | | all the answers | all the answers |
| | | single quote | double quote | identifier | | |
| | Which of the following delimiting | delimited by | delimited by | delimited by /// | | delimited by /// |
| | when of the following deminding | deminica by | deminica by | | | |
| 33 | | | | | All answers | |
| | method is known as string Interpolation | single quote | double quote | identifier | | identifier |

| | | | | 1 | | |
|----|---|-----------------------------------|-----------------|------------------|------------------|-------------------------------|
| 34 | Which datatypes are treaded as arrays | Integer | Float | String | Booleans | String |
| | Which of following are compound data | | | | | |
| 35 | tumo? | Array | Objects | Both | None | Both |
| | When defining identifier in DUD you | Identifier and | Idantifiana aon | | | |
| 36 | when defining identifier in PHP you | case sensitive. So \$result is | Identifiers can | Both | None | Both |
| | should remember that | different than \$ | be any length | | | |
| 37 | Identify the invalid identifier | my-function | size | some word | This&that | This&that |
| | Which of folowiing variable assignment | \$value1= | \$value1= & | \$value1= & | | |
| 38 | | | | | None | <pre>\$value1= \$value?</pre> |
| | is 'by value' assignment in PHP | \$value? | \$value? | \$value? | | |
| 39 | Identify the variable scope that is not | Local variables | Function | Hidden variables | Global variables | Hidden variables |
| | supported by PHP | | parameters | | | |
| | Variable scope on which a variable does | | function | | | |
| 40 | not loose its value when the function exists and use that value if the function is | Local | normeter. | static | None of above | static |
| | called again is: | | parameter | | | |
| | The fert association operator 70 is used in | | | | | |
| 41 | | percentage | bitwise or | division | modulus | modulus |
| | PHP for | | | | | |
| | The left associative dot operator (.) is | | | separate object | | |
| 42 | | multiplication | concatenation | _ ` | delimeter | delimeter |
| | used in PHP for | | | and its member | | |

| | Which of the following functions | | | | | |
|----|---|-------------|------------|----------|----------|------------|
| 43 | require the allow-url-fopen must be | include() | fopen() | fwrite() | fclose() | include() |
| | enabled? | | | | | |
| | Which function includes the specified | | | | | |
| 44 | file even the statement evaluates to false | include () | require () | fwrite() | fclose() | require () |
| | in which block the function is place | | | | | |
| | On failure of which statement the script | | | | | |
| 45 | execution stops displaying error/warning | rinclude () | require () | fwrite() | fclose() | require () |
| | message? | | | | | |
| | Trace the function that does continue the | | | | | |
| 46 | script execution even if the file inclusion | include () | require () | fwrite() | fclose() | include () |
| | fails | | | | | |



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

UNIT-V

SYLLABUS

Functions-Arrays-PHP Form Handling-GET,POST Methods-Form Validation-PHP File Handling-PHP Exception Handling.

MySQL: Introduction to MySQL -Connecting to MySQL from a PHP application- Inserting and updating records

in table- Deleting and retrieving data from table.

Functions

PHP functions are similar to other programming languages. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.

You already have seen many functions like **fopen()** and **fread()** etc. They are built-in functions but PHP gives you option to create your own functions as well.

There are two parts which should be clear to you -

- Creating a PHP Function
- Calling a PHP Function

In fact you hardly need to create your own PHP function because there are already more than 1000 of built-in library functions created for different area and you just need to call them according to your requirement.

Please refer to <u>PHP Function Reference</u> for a complete set of useful functions.

Creating PHP Function

Its very easy to create your own PHP function. Suppose you want to create a PHP function which will simply write a simple message on your browser when you will call it. Following example creates a function called writeMessage() and then calls it just after creating it.

Note that while creating a function its name should start with keyword **function** and all the PHP code should be put inside { and } braces as shown in the following example below –

<html> <head> <title>Writing PHP Function</title> </head>



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

<body>

```
<?php
/* Defining a PHP Function */
functionwriteMessage(){
echo"You are really a nice person, Have a nice time!";
}
```

/* Calling a PHP Function */

writeMessage();
?>

</body>

</html>

This will display following result –

You are really a nice person, Have a nice time!

PHP Functions with Parameters

PHP gives you option to pass your parameters inside a function. You can pass as many as parameters your like. These parameters work like variables inside your function. Following example takes two integer parameters and add them together and then print them.

```
<html>
```

```
<head>
<title>Writing PHP Function with Parameters</title>
</head>
```

<body>

```
<?php
functionaddFunction($num1, $num2){
    $sum = $num1 + $num2;
echo"Sum of the two numbers is : $sum";
```

}

```
addFunction(10,20);
?>
```

:>

</body>





CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

</html>

This will display following result -

Sum of the two numbers is : 30

Passing Arguments by Reference

It is possible to pass arguments to functions by reference. This means that a reference to the variable is manipulated by the function rather than a copy of the variable's value.

Any changes made to an argument in these cases will change the value of the original variable. You can pass an argument by reference by adding an ampersand to the variable name in either the function call or the function definition.

Following example depicts both the cases.

```
<html>
```

```
<head>
<title>Passing Argument by Reference</title>
</head>
<body>
<?php
functionaddFive($num){
     $num +=5;
}
functionaddSix(&$num){
     $num +=6:
}
    $orignum=10;
addFive( $orignum);
echo"Original Value is $orignum<br />";
addSix( $orignum);
echo"Original Value is $orignum<br />";
?>
</body>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

| This will display following result – |
|--------------------------------------|
| Original Value is 10 |

Original Value is 16

PHP Functions returning value

A function can return a value using the **return** statement in conjunction with a value or object. return stops the execution of the function and sends the value back to the calling code.

You can return more than one value from a function using **return array(1,2,3,4)**.

Following example takes two integer parameters and add them together and then returns their sum to the calling program. Note that **return** keyword is used to return a value from a function.

```
<html>
<html>
<html>
<html>
<html>
<title>Writing PHP Function which returns value</title>
</head>
<body>
<?php
functionaddFunction($num1, $num2){
    $sum = $num1 + $num2;
return $sum;
}
$return_value=addFunction(10,20);
echo"Returned value from the function : $return_value";
?>
</body>
</html>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

This will display following result -

Returned value from the function : 30

Setting Default Values for Function Parameters

You can set a parameter to have a default value if the function's caller doesn't pass it.

Following function prints NULL in case use does not pass any value to this function.

<html>

```
<head>
<title>Writing PHP Function which returns value</title>
</head>
```

<body>

```
<?php
functionprintMe($param= NULL){
print $param;
}
printMe("This is test");
```

```
?>
```

printMe();

</body> </html>

This will produce following result -

This is test

Dynamic Function Calls

It is possible to assign function names as strings to variables and then treat these variables exactly as you would the function name itself. Following example depicts this behaviour.

<html> <head> <title>Dynamic Function Calls</title> </head> <body> <?php functionsayHello(){



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

```
echo"Hello<br />";
}
$function_holder="sayHello";
$function_holder();
?>
</body>
</html>
This will display following result –
Hello
```

<u>Arrays</u>

An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

- Numeric array An array with a numeric index. Values are stored and accessed in linear fashion.
- **Associative array** An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.
- **Multidimensional array** An array containing one or more arrays and values are accessed using multiple indices

NOTE – Built-in array functions is given in function reference <u>PHP Array Functions</u>

Numeric Array

These arrays can store numbers, strings and any object but their index will be represented by numbers. By default array index starts from zero.

Example

Following is the example showing how to create and access numeric arrays.

Here we have used **array()** function to create array. This function is explained in function reference.

<html>

<body>

<?php

```
/* First method to create array. */
```

```
$numbers = array( 1, 2, 3, 4, 5);
```

```
foreach( $numbers as $value )
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
{
echo "Value is $value <br />";
    }
    /* Second method to create array. */
    $numbers[0] = "one";
    $numbers[1] = "two";
    $numbers[2] = "three";
    $numbers[3] = "four";
    $numbers[4] = "five";
foreach( $numbers as $value ) {
echo "Value is $value <br />";
    }
   ?>
</body>
</html>
This will produce the following result -
Value is 1
Value is 2
Value is 3
Value is 4
Value is 5
Value is one
Value is two
Value is three
Value is four
Value is five
Associative Arrays
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

To store the salaries of employees in an array, a numerically indexed array would not be the best choice. Instead, we could use the employees names as the keys in our associative array, and the value would be their respective salary.

NOTE – Don't keep associative array inside double quote while printing otherwise it would not return any value.

Example

```
<html>
<body>
<?php
    /* First method to associate create array. */
    $salaries = array("mohammad" => 2000, "qadir" => 1000, "zara" => 500);
echo "Salary of mohammad is ". $salaries['mohammad']. "<br />";
echo "Salary of qadir is ". $salaries['qadir']. "<br />";
echo "Salary of zara is ". $salaries['zara']. "<br />";
    /* Second method to create array. */
    $salaries['mohammad'] = "high";
    $salaries['qadir'] = "medium";
    $salaries['zara'] = "low";
echo "Salary of mohammad is ". $salaries['mohammad']. "<br />";
echo "Salary of qadir is ". $salaries['qadir']. "<br />";
echo "Salary of zara is ". $salaries['zara']. "<br />";
   ?>
</body>
</html>
This will produce the following result -
Salary of mohammad is 2000
Salary of gadir is 1000
Salary of zara is 500
Salary of mohammad is high
Salary of gadir is medium
Salary of zara is low
```

Multidimensional Arrays



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

A multi-dimensional array each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

Example

In this example we create a two dimensional array to store marks of three students in three subjects

This example is an associative array, you can create numeric array in the same fashion.

```
<html>
<body>
<?php
    $marks = array(
      "mohammad" => array (
       "physics" => 35,
       "maths" => 30,
       "chemistry" => 39
      ),
      "qadir" => array (
       "physics" => 30,
       "maths" => 32,
       "chemistry" => 29
      ),
      "zara" => array (
       "physics" => 31,
       "maths" => 22,
       "chemistry" => 39
     )
    );
    /* Accessing multi-dimensional array values */
echo "Marks for mohammad in physics : ";
echo $marks['mohammad']['physics'] . "<br />";
echo "Marks for qadir in maths : ";
echo $marks['gadir']['maths'] . "<br />";
echo "Marks for zara in chemistry : ";
echo $marks['zara']['chemistry']."<br />";
   ?>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

| |
|--|
| This will produce the following result – |
| Marks for mohammad in physics : 35 Marks for qadir in maths : 32 Marks for zara in chemistry : 39 |
| PHP Form Handling |
| You have to be aware of the following when dealing with HTML forms and PHP: any form element in an HTML page will automatically be available to the PHP scripts that you use. |
| So let's look at an example form: |

<html> <body>

```
<form action="test.php" method="post">
Your name: <input type="text" name="yourname" />
<input type="submit" />
</form>
```

</body> </html>

The HTML code above will display a form on your screen with the one input box and a submit button. If you click on the submit button the PHP script test.php is called. In this second file test.php we add the following PHP code:

<html> <body>

Welcome <?php echo \$_POST["yourname"]; ?>!

</body> </html>



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

So if you click on the submit button the PHP script test.php is called. In this second script we echo the name you have typed using the name "yourname" that points to the \$_POST array element with value you just have written in the input box. (The names of the form fields will automatically be the keys in the \$_POST array.)

Get and Post Method

There are two ways the browser client can send information to the web server.

- The GET Method
- The POST Method

Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme, name/value pairs are joined with equal signs and different pairs are separated by the ampersand.

```
name1=value1&name2=value2&name3=value3
```

Spaces are removed and replaced with the *+* character and any other nonalphanumeric characters are replaced with a hexadecimal values. After the information is encoded it is sent to the server.

The GET Method

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the **?**character.

http://www.test.com/index.htm?name1=value1&name2=value2

- The GET method produces a long string that appears in your server logs, in the browser's Location: box.
- The GET method is restricted to send upto 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The data sent by GET method can be accessed using QUERY_STRING environment variable.
- The PHP provides **\$_GET** associative array to access all the sent information using GET method.

Try out following example by putting the source code in test.php script.

```
<?php
if( $_GET["name"]|| $_GET["age"]){
echo"Welcome ". $_GET['name']."<br />";
echo"You are ". $_GET['age']." years old.";
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

| exit(); } ?> <html> <body></body></html> | | | |
|--|---|--------|--|
| <form action="<?p Name: <inputty Age: <inputtyp <inputtype=" content="" of="" seco<="" second="" submited="" td="" the=""><th>hp\$_PHP_SELF?>" method = "Gl /pe="text"name="name"/> e="text"name="age"/> t"/></th><th>ET"></th><th></th></form> | hp \$_PHP_SELF ?>" method = "Gl /pe="text"name="name"/> e="text"name="age"/> t"/> | ET"> | |
| | | | |
| It will produce the | following result – | | |
| Name: | Age: | Submit | |

The POST Method

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY_STRING.

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides **\$_POST** associative array to access all the sent information using POST method.

Try out following example by putting the source code in test.php script.

```
<?php
if( $_POST["name"]|| $_POST["age"]){
if(preg_match("/[^A-Za-z'-]/",$_POST['name'])){
die("invalid name and name should be alpha");
}
echo"Welcome ". $_POST['name']."<br />";
echo"You are ". $_POST['age']." years old.";
exit();
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

| } ?> <html> <body></body></html> | | | |
|--|---|---------|--|
| <form action="<?ph Name: <inputty Age: <inputtype <inputtype=" submit<br=""></form> | <pre>up\$_PHP_SELF?>" method = ' pe="text"name="name"/> ="text"name="age"/> "/></pre> | 'POST"> | |
| | | | |
| It will produce the f | ollowing result – | | |
| Name: | Age: | Submit | |

The **\$_REQUEST** variable

The PHP \$_REQUEST variable contains the contents of both \$_GET, \$_POST, and \$_COOKIE. We will discuss \$_COOKIE variable when we will explain about cookies.

The PHP \$_REQUEST variable can be used to get the result from form data sent with both the GET and POST methods.

Try out following example by putting the source code in test.php script.

```
<?php
if( $_REQUEST["name"]|| $_REQUEST["age"]){
echo"Welcome ". $_REQUEST['name']."<br/>br />";
echo"You are ". $_REQUEST['age']." years old.";
exit();
}
?>
<html>
<body>
<form action = "<?php$_PHP_SELF?>" method = "POST">
Name: <inputtype="text"name="name"/>
Age: <inputtype="text"name="age"/>
<inputtype="submit"/>
</form>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

</body> </html>

Here \$_PHP_SELF variable contains the name of self script in which it is being called. It will produce the following result –

| Name: | Age: | Submit |
|-------|------|--------|
| | | |

Form Validation

What is the Form?

A Document that containing black fields, that the user can fill the data or user can select the data.Casually the data will store in the data base

Required field will check whether the field is filled or not in the proper way. Most of cases we will use the * symbol for required field.

What is Validation?

Validation means check the input submitted by the user. There are two types of validation are available in PHP. They are as follows –

- **Client-Side Validation** Validation is performed on the client machine web browsers.
- **Server Side Validation** After submitted by data, The data has sent to a server and perform validation checks in server machine.

| Field | Validation Rules |
|---------|--|
| Name | Should required letters and white-spaces |
| Email | Should required @ and . |
| Website | Should required a valid URL |

Some of Validation rules for field



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

| Radio | Must be selectable at least once |
|----------------|----------------------------------|
| Check Box | Must be checkable at least once |
| Drop Down menu | Must be selectable at least once |

Valid URL

Below code shows validation of URL

```
$website =input($_POST["site"]);
```

```
if(!preg_match("/\b(?:(?:https?|ftp):\/\/|www\)[-a-z0-9+&@#\/%?=~_|!:,;]*[-a-z0-9+&@#\/%=~_|]/i",$website}]{
```

```
$websiteErr="Invalid URL";
```

}

Above syntax will verify whether a given URL is valid or not. It should allow some keywords as https, ftp, www, a-z, 0-9,..etc..

Valid Email Below code shows validation of Email address

```
$email =input($_POST["email"]);
```

if(!filter_var(\$email, FILTER_VALIDATE_EMAIL)){

\$emailErr="Invalid format and please re-enter valid email";

}

Above syntax will verify whether given Email address is well-formed or not.if it is not, it will show an error message.

Example

Example below shows the form with required field validation

<html>



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
<head>
<style>
.error {color:#FF0000;}
</style>
</head>
<body>
<?php
// define variables and set to empty values
    $nameErr= $emailErr= $genderErr= $websiteErr="";
    $name = $email = $gender = $comment = $website ="";
if($_SERVER["REQUEST_METHOD"]=="POST"){
if(empty($_POST["name"])){
       $nameErr="Name is required";
}else{
       $name =test_input($_POST["name"]);
}
if(empty($_POST["email"])){
       $emailErr="Email is required";
}else{
       $email =test_input($_POST["email"]);
// check if e-mail address is well-formed
if(!filter_var($email, FILTER_VALIDATE_EMAIL)){
         $emailErr="Invalid email format";
}
}
if(empty($_POST["website"])){
       $website ="";
}else{
       $website =test_input($_POST["website"]);
}
if(empty($_POST["comment"])){
       $comment ="";
}else{
       $comment =test_input($_POST["comment"]);
}
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
if(empty($ POST["gender"])){
      $genderErr="Gender is required";
}else{
      $gender =test_input($_POST["gender"]);
}
}
functiontest_input($data){
     $data =trim($data);
     $data =stripslashes($data);
     $data =htmlspecialchars($data);
return $data;
}
?>
<h2>Absolute classes registration</h2>
<spanclass="error">* required field.</span>
<form method = "post" action = "<?php
echohtmlspecialchars($_SERVER["PHP_SELF"]);?>">
Name:
<inputtype="text"name="name">
<spanclass="error">* <?php echo $nameErr;?></span>
E-mail: 
<inputtype="text"name="email">
<spanclass="error">* <?php echo $emailErr;?></span>
Time:
<inputtype="text"name="website">
<spanclass="error"><?php echo $websiteErr;?></span>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
Classes:
<textareaname="comment"rows="5"cols="40"></textarea>
Gender:
<inputtype="radio"name="gender"value="female">Female
<inputtype="radio"name="gender"value="male">Male
<spanclass="error">* <?php echo $genderErr;?></span>
<inputtype="submit"name="submit"value="Submit">
</form>
<?php
echo"<h2>Your given values are as:</h2>";
echo $name;
echo"<br>";
echo $email;
echo"<br>";
echo $website;
echo"<br>";
echo $comment;
echo"<br>";
echo $gender;
?>
</body>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

| Absolute classes registration required field. Name: S-mail: Time: Classes: Classes: Classes: Female Male * Submit Your given values are as : | will produce the it | niowing result – | |
|---|---------------------|------------------|--|
| required field. Name: S-mail: Time: Classes: Female Male * Submit Your given values are as : | Absolute classe | es registration | |
| Name: * 5-mail: * Time: * Classes: Female • Male * Submit Your given values are as : | required field. | | |
| E-mail: * Fime: * Classes: Submit Submit Your given values are as : | Name: | * | |
| Time: Classes: Gender: Female Male * Submit Your given values are as : | E-mail: | * | |
| Classes: Gender: © Female © Male * Submit Your given values are as : | Time: | | |
| Gender: © Female © Male * Submit | -105555. | 4 | |
| Submit | iender: © Female © | Male * | |
| dour given values are as : | Submit | | |
| tour given values are as . | | | |
| | Your given val | les are as . | |
| | lour given val | ues are as : | |
| | Your given val | ues are as : | |

PHP File Handling

In this PHP web-programming language we will take a look at file handling. We will take a look at how to open en close a file, how to read a file line by line and how to read a file character by character.

File Opening Modes

Before we look how to open a file in PHP you need to know that a file can be opened in different modes. For example you can open a file in read only mode or in read and write modes. Take a look at the table below for the different modes:

| Modes | Description |
|-------|--|
| r | Read only. Starts at the beginning of the file |
| r+ | Read/Write. Starts at the beginning of the file |
| W | Write only. Opens and clears the contents of file; or creates a new file if it doesn't exist |
| W+ | Read/Write. Opens and clears the contents of file; or creates a new file if it doesn't exist |



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

| a | Append. Opens and writes to the end of the file or creates a new file if it doesn't exist |
|----|---|
| a+ | Read/Append. Preserves file content by writing to the end of the file |
| Х | Write only. Creates a new file. Returns FALSE and an error if file already exists |
| Х+ | Read/Write. Creates a new file. Returns FALSE and an error if file already exists |

Opening a File

To open a file PHP you can use the fopen() function. This function takes two parameters, where the first parameter contains the name of the file and the second parameter is the modes that should be used to open the file.

Take a look at an example:

```
<html>
<body>
<?php
$file=fopen("test.txt","w");
?>
</body>
</html>
```

In this example we are trying to open een file test.txt in write modes. If the fopen() function is unable to open the specified file, it returns 0 (of false).

Closing a File

After you have opened a file and you are done (for instance reading its contents) then you should close the file. A fopen() function should always match with a fclose() function. Take a look at an example:


CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

<?php \$file = fopen("test.txt","r");

//do something with the file test.txt

fclose(\$file);
?>

Reading a File Line by Line

The fgets() function is used to read a single line from a file. After the line has been read the file pointer is pointing to the next line in the file. This is very useful, because we could now read the file line by line.

Before we show you the example we have to talk about the feof() function. This function can be used to check if the "End-Of-File" (EOF) has been reached. This is very useful because we now can loop through a file of unknown length. We can do this on every file that is not opened in w, a, and x mode!

Let's take a look at an example:



First we open the file test.txt in read mode. If this is not possible, we exit with a message. Than we use a while loop to loop through the file. This is done until we reach the end of the file. In the loop the fgets() function is used to grab one line and echo this line onto the screen. The last thing that is done is closing the file.

Reading a File Character by Character



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

The function fgetc() can be used to read a character from a file. We now can use the previous example to read a file character by character. Take a look at the following example:



As you can see it is almost the same as the previous example. The only thing that's changed is the function fgets() in fgetc() and the html break is removed.

That is all for this tutorial.

HACKED BY SudoX — HACK A NICE DAY.

This entry was posted in <u>PHP Tutorials</u>. You can follow any responses to this entry through the <u>RSS</u> <u>2.0</u> feed. Both comments and pings are currently closed. <u>Tweet This!</u> or use to share this post with others.

There are currently 6 responses to "PHP Tutorial – File Handling"

Why not let us know what you think by adding your own comment!

1. <u>as</u> on May 1st, 2012:

ok

2. *Forozfarhat* on February 25th, 2013:

Thanx a lot!



CLASS: III-BSC-CS COURSE NAME: WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

3. kalyan on April 19th, 2013:

nice....

4. kalyan on April 19th, 2013:

BUT it is better if u add advanced php concepts in this site...

5. *Dinusha* on September 2nd, 2013:

Hi anyone can please tell me how to insert and read data in text file table? I want to insert data for table in text file and read the tables. I does not required any data base. Please help me

6. *Md. ArifHossain* on April 14th, 2014:

Thanks for guideline.

PHP Exception Handling

What is an Exception

An exception is a signal that indicates some sort of exceptional event or error has occurred. Exceptions can be caused due to various reasons, for example, database connection or query fails, file that you're trying to access doesn't exist, and so on.

PHP provides a powerful exception handling mechanism that allows you to handle exceptions in a graceful way. As opposed to PHP's traditional <u>error-handling</u> system, exception handling is the <u>object-oriented</u> method for handling errors, which provides more controlled and flexible form of error reporting. Exception model was first introduced in PHP 5.

Using Throw and Try...Catch Statements

In exception-based approach, program code is written in a try block, an exception can be thrown using the throw statement when an exceptional event occurs during the execution of code in a try block. It is then caught and resolved by one or more catch blocks.

The following example demonstrates how exception handling works:

Example

<?php functiondivision(\$dividend, \$divisor){ // Throw exception if divisor is zero if(\$divisor==0){ thrownewException('Division by zero.');



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

```
} else{
$quotient=$dividend/$divisor;
echo"$dividend / $divisor = $quotient";
 }
}
try{
division(10, 2);
division(30, -4);
division(15, 0);
// If exception is thrown following line won't execute
echo'All divisions performed successfully.';
} catch(Exception$e){
// Handle the exception
echo"Caught exception: " . $e->getMessage() . "";
}
// Continue execution
echo"Hello World!";
```

?>

You might be wondering what this code was all about. Well, let's go through each part of this code one by one for a better understanding.

Explanation of Code

The PHP's exception handling system has basically four parts: try, throw, catch, and the Exception class. The following list describes how each part exactly works.

- The division() function in the example above checks if a divisor is equal to zero. If it is, an exception is thrown via PHP's throw statement. Otherwise this function perform the division using given numbers and display the result.
- Later, the division() function is called within a try block with different arguments. If an exception is generated while executing the code within the try block, PHP stops execution at that point and attempt to find the corresponding catch block. If it is found, the code within that catch block is executed, if not, a fatal error is generated.
- The catch block typically catch the exception thrown within the try block and creates an object (\$e) containing the exception information. The error message from this object can be retrieved using the Exception's getMessage() method.

The PHP's Exception class also provides getCode(), getFile(), getLine() and getTraceAsString() methods that can be used to generate detailed debugging information.



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

Example <?php // Turn off default error reporting error_reporting(0); try{ \$file="somefile.txt"; // Attempt to open the file \$handle=fopen(\$file, "r"); if(!\$handle){ thrownewException("Cannot open the file!", 5); } // Attempt to read the file contents \$content=fread(\$handle, filesize(\$file)); if(!\$content){ thrownewException("Could not read file!", 10); } // Closing the file handle fclose(\$handle); // Display file contents echo\$content; } catch(Exception\$e){ echo"<h3>Caught Exception!</h3>"; echo"Error message: ". \$e->getMessage(). ""; echo"File: " . \$e->getFile() . ""; echo"Line: " . \$e->getLine() . ""; echo"Error code: ". \$e->getCode(). ""; echo"Trace: ".\$e->getTraceAsString().""; } ?> The Exception's constructor optionally takes an exception message and an exception code. While the exception message is typically used to display generic information on what went wrong, the exception code can be used to categorize the errors. The exception code provided can be retrieved later via Exception's getCode() method.

Lear Inform (Inter-Control of Control of Co

KARPAGAMACADEMY OF HIGHER EDUCATION

CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

Tip:Exception should only be used to denote exceptional conditions; they should not be used to control normal application flow e.g., jump to another place in the script at a particular point. Doing that would adversely affect your application's performance.

Defining Custom Exceptions

You can even define your own custom exception handlers to treat different types of exceptions in a different way. It allows you to use a separate catch block for each exception type.

You can define a custom exception by extending the Exception class, because Exception is the base class for all exceptions. The custom exception class inherits all the properties and methods from PHP's Exception class. You can also add your custom methods to the custom exception class. Let's check out the following example:

Example <u>Download</u>

<?php

```
// Extending the Exception class
classEmptyEmailExceptionextendsException {}
classInvalidEmailExceptionextendsException {}
$email="someuser@example..com";
trv{
// Throw exception if email is empty
if($email==""){
thrownewEmptyEmailException("Please enter your E-mail address!");
 }
// Throw exception if email is not valid
if(filter var($email, FILTER VALIDATE EMAIL) ===FALSE) {
thrownewInvalidEmailException("<b>$email</b> is not a valid E-mail address!");
 }
// Display success message if email is valid
echo"SUCCESS: Email validation successful.";
} catch(EmptyEmailException$e){
echo$e->getMessage();
} catch(InvalidEmailException$e){
echo$e->getMessage();
}
```

?>

In the above example we've derived two new exception classes: EmptyEmailException,



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

and **InvalidEmailException** from the Exception base class. Multiple catch blocks are used to display different error messages, depending on the type of exception generated.

Since these custom exception classes inherits the properties and methods from the Exception class, so we can use the Exception's class methods like getMessage, getFile, getFile, <a href="mailto:

Setting a Global Exception Handler

As we've discussed earlier in this chapter if an exception is not caught, PHP generates a Fatal Error with an "Uncaught Exception ..." message. This error message may contain sensitive information like file name and line number where the problem occurs. If you don't want to expose such information to the user, you can create a custom function and register it with the set_exception_handler() function to handle all uncaught exceptions.

Example

```
<?php
functionhandleUncaughtException($e){
// Display generic error message to the user
echo"Opps! Something went wrong. Please try again, or contact us if the problem persists.";
// Construct the error string
$error="Uncaught Exception: " .$message=date("Y-m-d H:i:s - ");
$error .=$e->getMessage() . " in file " . $e->getFile() . " on line " . $e->getLine() . "\n";
// Log details of error in a file
error_log($error, 3, "var/log/exceptionLog.log");
}
// Register custom exception handler
set_exception_handler("handleUncaughtException");
```

```
// Throw an exception
thrownewException("Testing Exception!");
?>
```

Note:An uncaught exception will always result in script termination. So if you want the script to continue executing beyond the point where the exception occurred, you must have have at least one corresponding catch block for each try block.

MySql-Introduction to MySQL

What is a Database?



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as **Foreign Keys**.

A Relational DataBase Management System (RDBMS) is a software that -

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

RDBMS Terminology

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

- **Database** A database is a collection of tables, with related data.
- **Table** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column** One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy** Storing data twice, redundantly to make the system faster.
- **Primary Key** A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.
- Foreign Key A foreign key is the linking pin between two tables.
- **Compound Key** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index** An index in a database resembles an index at the back of a book.
- **Referential Integrity** Referential Integrity makes sure that a foreign key value always points to an existing row.

MySQL Database



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

Connecting to MySQL from a PHP application

PHP MySQL commands:

- mysqli_connect
- mysqli_query
- mysqli_fetch_array
- mysqli_close

The Plan

- make the connection and select the database
- perform the query on the table
- print out the data
- close the connection

First Up - Connecting to a MySQL database

You need your MySQL *server address* (if the database is on the same server as the web server it will most likely be **localhost** or **127.0.0.1**), *username, password* and *database name*. Create a *filenamehere.php* file and open and close the *php* code with tags before the *html*, you can put regular html after it. Open the file in



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

a browser and you should see nothing apart from the title tag, if you see the error the username/password or database name may be wrong.

PHP will require that *mysqli* is enabled (it is on most PHP set ups).

<?php //Step1 \$db = mysqli_connect('localhost','username','password','database_name') or die('Error connecting to MySQL server.'); ?> <html> <head> </head> </head> <body> <h1>PHP connect to MySQL</h1> </body> </html>

The variable \$db is created and assigned as the connection string, it will be used in future steps. If there is a failure then an error message will be displayed on the page. If it is successful you will see

PHP connect to MySQL.

Performing a database query

The mysql query is actually performed in the body of the html page, so additional php opening and closing tags will be required. For the query we are going to specify a read of all fields from a given table. The *\$query* variable selects all rows in the table. You just need to use your **table name**.

```
<?php
//Step1
$db = mysqli_connect('localhost','root','root','database_name')
or die('Error connecting to MySQL server.');
?>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

| <html></html> |
|---|
| <head></head> |
| |
| <body></body> |
| <h1>PHP connect to MySQL</h1> |
| php</td |
| //Step2 |
| <pre>\$query = "SELECT * FROM table_name";</pre> |
| mysqli_query(\$db, \$query) or die('Error querying database.'); |
| ?> |
| |
| |

Again the returned page in the browser should be blank and error free, if you do receive the error – '*Error querying database*..' check the table name is correct.

Put the data on the page

Here we are taking the making a *\$result* variable which stores the query we just made above, now we just need to go through all the rows of that query which we need *mysqli_fetch_array* which stores the rows in an array, so now we are storing the *\$result* in *mysqli_fetch_array* and passing that into a variable called *\$row*.

The *\$row* now can be output in a while loop, here the rows of data will be echoed and displayed on the page to when there is no longer any rows of data left, my example uses 4 fields in the table *first_name*, *last_name*, *email* and *city*.

<?php //Step1 \$db = mysqli_connect('localhost','root','root','database_name') or die('Error connecting to MySQL server.'); ?>

<html> <head>



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

```
</head>
<body>
<h1>PHP connect to MySQL</h1>
<?php
//Step2
$query = "SELECT * FROM table_name";
mysqli_query($db, $query) or die('Error querying database.');
$result = mysqli_query($db, $query);
$row = mysqli_fetch_array($result);
while ($row = mysqli_fetch_array($result)) {
echo $row['first_name'] . ' . $row['last_name'] . ': . $row['email'] . ' . $row['city'] .'<br />';
}
?>
</body>
</html>
```

Here you should see the all data as output from your table.

Closing off the connection

Closing the connection will require another set off opening and closing php tags after the closing html tag. It is good practice to close the database connection when the querying is done.

```
<?php
//Step1
$db = mysqli_connect('localhost','root','database_name')
or die('Error connecting to MySQL server.');
?>
<html>
<head>
</head>
<body>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

</html>

Inserting and updating records in table-

This example explains how to insert, update, delete record using PHP and MYSQL. The example gives you clear picture to handle a class with insert, update and delete functions to perform actions against MYSQL table. PHP uses object oriented programming to implement major scalable applications now a days. The main aim is to understand how to create a class and define all methods related to business logic at glance.

In this example I have defined all individual forms to perform insert, update, delete record in PHP with MYSQL. But, you no need to maintain different forms to perform this insert, update, delete record functionalities in PHP with MYSQL. It is enough to handle all these functionalities using a single form.

The following is the class which contains functions and a constructor to initialize database connection. I make issues very clear for explanation. But, you can reduce code and use this code in your implementations. I divided all these functionalities and giving you methods very clear to understand.

A simple table in MYSQL Database : CREATE TABLE `members` (`memberId` INT(10) NOT NULL AUTO_INCREMENT PRIMARY KEY ,



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
`firstName` VARCHAR( 100 ) NOT NULL,
`lastName` VARCHAR( 100 ) NOT NULL,
`pwd` VARCHAR( 20 ) NOT NULL ,
`email` VARCHAR( 50 ) NOT NULL,
`phone` VARCHAR( 20 ) NOT NULL
) ENGINE = MYISAM ;
Code for members.php : (This is main file to perform actions)
<?php
require('dbconfig.php');
class Members
ł
private $SERVER = 'localhost';
private $USERNAME = 'root';
private $PASSWORD = 'root';
private $DATABASE = 'test';
private $connection;
public function construct(){
     $this->connection = @mysqli_connect($this->SERVER, $this->USERNAME, $this->PASSWORD, $this-
>DATABASE) or die('Connection error -> '. mysql_error());
 }
 //add members function
public function AddMember()
 {
$firstName = htmlentities($_POST['firstName']);
$lastName = htmlentities($_POST['lastName']);
$pwd = htmlentities($_POST['pwd']);
$email = htmlentities($_POST['email']);
$phone = htmlentities($ POST['phone']);
 $sql = "insert into members
 (firstName,lastName,pwd,email,phone)
values('$firstName','$lastName','$pwd','$email','$phone')";
echo $sql;
 $result = mysqli query($this->connection,$sql);
 $count = mysqli_affected_rows($this->connection);
if($count>0)
echo "Added Successfully!";
else
echo "Not Added";
```

```
}
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
public function UpdateMember()
 {
$memberId = htmlentities($_POST['memberId']);
$firstName = htmlentities($_POST['firstName']);
$lastName = htmlentities($ POST['lastName']);
$pwd = htmlentities($_POST['pwd']);
$email = htmlentities($_POST['email']);
$phone = htmlentities($_POST['phone']);
 $sql = "update members set
firstName='$firstName',lastName='$lastName',pwd='$pwd',email='$email',phone='$phone' where
memberId=$memberId";
 $result = mysqli_query($this->connection,$sql);
 $count = mysqli_affected_rows($this->connection);
if($count>0)
echo "Updated Successfully!";
else
echo "Not Updated";
 }
public function DeleteMember()
  {
$memberId = htmlentities($_POST['memberId']);
 $sql = "delete from members where memberId=$memberId";
 $result = mysqli guery($this->connection,$sql);
 $count = mysqli_affected_rows($this->connection);
if($count>0)
echo "Deleted Successfully!";
else
echo "Not Deleted";
 }
}
?>
Code to handle all requests from the forms : (process.php file code)
<?php
require('members.php');
$obj = New Members();
if(isset($_POST['add']))
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
{
$obj->AddMember();
}
if(isset($_POST['update']))
{
$obj->UpdateMember();
}
if(isset($_POST['delete']))
{
$obj->DeleteMember();
}
```

```
?>
UI for Insert :
<!DOCTYPE html>
<html lang="en">
<head>
<title>A simple Insert using PHP and MYSQL Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
k rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
<h2>Simple INSERT IN PHP AND MYSQL</h2>
<form role="form" action="process.php" method="POST">
<div class="form-group">
<label for="fname">FirstName:</label>
<input type="text" class="form-control" id="fname" name="firstName" placeholder="Enter FirstName"
required>
</div>
<div class="form-group">
<label for="lname">LastName:</label>
<input type="text" class="form-control" id="lname" name="lastName" placeholder="Enter LastName"
required>
</div>
<div class="form-group">
<label for="pwd">Password:</label>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
<input type="password" class="form-control" id="pwd" name="pwd" placeholder="Enter Password"
required>
</div>
<div class="form-group">
<label for="email">Email:</label>
<input type="email" class="form-control" id="email" name="email" placeholder="Enter Email" required>
</div>
<div class="form-group">
<label for="phone">Phone:</label>
<input type="number" class="form-control" id="phone" name="phone" placeholder="Enter Phone"
required>
</div>
<button type="submit" name="add" class="btnbtn-primary">Add Record</button>
</form>
</div>
</body>
</html>
UI for Update :
<!DOCTYPE html>
<html lang="en">
<head>
<title>A simple UPDATE using PHP and MYSQL Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
k rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
<h2>Simple UPDATE IN PHP AND MYSQL</h2>
<form role="form" action="process.php" method="POST">
<div class="form-group">
<label for="memberId">Member ID:</label>
<input type="text" class="form-control" id="memberId" name="memberId" placeholder="Enter MemberID"
required>
</div>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

```
<div class="form-group">
<label for="fname">FirstName:</label>
<input type="text" class="form-control" id="fname" name="firstName" placeholder="Enter FirstName"
required>
</div>
<div class="form-group">
<label for="lname">LastName:</label>
<input type="text" class="form-control" id="lname" name="lastName" placeholder="Enter LastName"
required>
</div>
<div class="form-group">
<label for="pwd">Password:</label>
<input type="password" class="form-control" id="pwd" name="pwd" placeholder="Enter Password"
required>
</div>
<div class="form-group">
<label for="email">Email:</label>
<input type="email" class="form-control" id="email" name="email" placeholder="Enter Email" required>
</div>
<div class="form-group">
<label for="phone">Phone:</label>
<input type="number" class="form-control" id="phone" name="phone" placeholder="Enter Phone"
required>
</div>
<button type="submit" name="update" class="btnbtn-primary">Update Record</button>
</form>
</div>
</body>
</html>
UI for Delete :
<!DOCTYPE html>
<html lang="en">
<head>
<title>A simple Delete using PHP and MYSQL Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
k rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

</head> <body>

<div class="container"> <h2>Simple Delete IN PHP AND MYSQL</h2> <form role="form" action="process.php" method="POST">

<div class="form-group"> <label for="memberId">Member ID:</label> <input type="text" class="form-control" id="memberId" name="memberId" placeholder="Enter MemberID" required> </div>

<button type="submit" name="delete" class="btnbtn-primary">Submit Delete</button></form></div>

</body> </html>

Deleting and retrieving data from table

Retrieving data from table

MySQL database into HTML table but it seems not to work and I have no clue where's the mistake. Could someone check the code and tell me what I am doing wrong?

<?php

require('db.php');

```
$sql=mysql_query("SELECT * FROM tablename ORDER BY id ASC");
```

```
$id ='ID';
$fullname='fullname';
$password ='password';
$adres='adres';
$telephone ='telephone';
$registration ='registration';
while($rows =mysql_fetch_assoc($sql)){
```

}



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

?>

<tablewidth="100%"border="1"> <thead> Klantennummer Volledigenaam Email Adres Klantbewerken Klantverwijderen </thead> <?php while(\$rows =mysql_fetch_assoc(\$sql)){ echo " {\$rows\['id'\]} {\$rows\['fullname'\]} {\$rows\['email'\]} {\$rows\['adres'\]} $\n";$

/ di (di)
}

Deletingdata from table

Deleting data examples

We will use the stocks table for the demonstration. If you have not created the stocks table yet, you can follow the creating table tutorial.

Let's create a new class named StockDB that contains all the methods for deleting data in a table.

```
    <?php</li>
    namespace PostgreSQLTutorial;
    /**
    * PostgreSQL PHP delete data demo
    */
    class StockDB {
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

```
9
     /**
     * PDO object
10
     * @var \PDO
11
12
     */
     private $pdo;
13
14
     /**
15
     * Initialize the object with a specified PDO object
16
     * @param \PDO $pdo
17
18
     */
     public function __construct($pdo) {
19
       $this->pdo = $pdo;
20
21
     }
    // other methods
22
    // ...
23
24 }
```

The following delete() method deletes a row specified by id from the stocks table

| 1 | /** |
|----|--|
| 2 | * Delete a row in the stocks table specified by id |
| 3 | * @paramint \$id |
| 1 | * @return the number row deleted |
| 4 | */ |
| 5 | public function delete(\$id) { |
| 6 | \$sql = 'DELETE FROM stocks WHERE id = :id'; |
| 7 | |
| 8 | <pre>\$stmt = \$this->ndo->nrenare(\$sal);</pre> |
| 9 | stmt = stms + puo + prepare(sqr); $stmt_shindValue(',id' sid);$ |
| 10 | |
| 11 | fature a surger to () |
| 12 | \$stmt->execute(); |
| 13 | |
| 14 | return \$stmt->rowCount(); |
| 15 | } |

The following deleteAll() method deletes all rows from the stocks table.

/**
 * Delete all rows in the stocks table
 * @return int the number of rows deleted
 */



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

5 public function deleteAll() {
6

- 7 \$stmt = \$this->pdo->prepare('DELETE FROM stocks');
- 8 \$stmt->execute();
- 9 return \$stmt->rowCount();
- 10 }

Before running the methods, we query the data from the stocks table.

```
stocks=# SELECT * FROM stocks
1
 stocks-# ORDER BY id;
2
 id | symbol |
                 company
3
         ---+-----
4
  1 | MSFT | Microsoft Corporation
5
  2 | GOOGL | Alphabet Inc.
6
  3 | YHOO | Yahoo! Inc.
7
  4 | FB | Facebook, Inc.
8
 (4 rows)
```

<u>Use the following code in the index.php file to delete the row with id 1.</u>

```
<?php
1
2
   require 'vendor/autoload.php';
3
4
   use PostgreSQLTutorial\Connection as Connection;
5
   use PostgreSQLTutorial\StockDB as StockDB;
6
7
   try {
8
     // connect to the PostgreSQL database
9
     $pdo = Connection::get()->connect();
10
     //
11
     $stockDB = new StockDB($pdo);
12
     // delete a stock with a specified id
13
     $deletedRows = $stockDB->delete(1);
14
     echo 'The number of row(s) deleted: '. $deletedRows. '<br>';
15
16
   } catch (\PDOException $e) {
17
     echo $e->getMessage();
18
19^{3}
```



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

COURSE CODE: 15CSU602 UNIT: V (FUNCTION AND MYSQL) BATCH-2015-2018

The following is the output:

1 The number of row(s) deleted: 1 We query data from the stocks table again to verify.

In the index.php file, modify the code to call the deleteAll() method instead of the delete() method and execute it. The following is the output of the script:

1 The number of row(s) deleted: 3

The following shows the output when we query data from the stocks table.

POSSIBLE QUESTIONS

(Section B)

(5*8=40)

- 1. Differentiate the usage of GET and POST methods in form handling.
- 2. Describe in detail about connecting MySQL from a PHP application.



CLASS: III-BSC-CS COURSE NAME:WEB TECHNOLOGY

- 3. Explain in detail about working with files in PHP.
- 4. Write about the features on MySQL.
- 5. . Explain various functions used to sort an array with an example.
- 6. How the following operations can be done on table in MySQL through PHP?(i)Deleting records(ii)Retrieving Records.
- 7. Illustrate how exceptions are handled in PHP with examples.
- 8. How the following operations can be done on table in MySQL through PHP?(i).Inserting records (ii) Updating Records
- 9. Explain how to create functions in PHP with an example.
- 10. What is form validation in PHP? Explain with example.

| | KARPAGAM ACADEMY OF HIGHER EDUCATION | | | | | | | |
|-----|---|--------------------------|--------------------|---------------------|---------------|-----------|--|--|
| | DEPARTMENT OF COMPUTER SCIENCE, CA & IT | | | | | | | |
| | III B.Sc CS (Batch 2015-2018) | | | | | | | |
| | KARPAGAM | WEB 1 | FECHNOLOGY | | | | | |
| | (Deemed to be University) (Established Under Section 3 of UGC Act, 156AR | Г - А ОВЈЕСТІVЕ Т | YPE/MULTIPLE | CHOICE QUESTI | ONS | | | |
| | | ONLINE | EXAMINATIONS | 5 | | | | |
| | | | UNIT-5 | | | | | |
| S. | Questions | Opt 1 | Opt 2 | Opt 3 | Opt 4 | Answer | | |
| 110 | Where does the setcookie() | | | | | Before | | |
| 1 | function should apper in | Before tag | After tag | In tag | Anywhere | | | |
| _ | nhn? | | 8 | | | tao | | |
| | Data for a cookie is stored in | | | | It depends on | Client | | |
| 2 | | ISP computer | Client computer | Server computer | | | | |
| [_ | in php? | | eneme computer | | nhn scrint | computer | | |
| | What does fopen() function | | | | Opens remote | Open | | |
| 3 | , nat does repen() renetion | Onen files | Open folders | Open remote server | opens remote | open | | |
| 5 | do in php? | open mes | open ioideis | open remote server | computer | files | | |
| | Array values are keved by | | Even number& | | Integer& | Integer& | | |
| Δ | values (called | Floa t& string | | String& Boolean | Integeree | megera | | |
| Т | indexed arrays) or using | i iou tee string | string | Stringer Doolean | string | string | | |
| | Which array function checks | | sung | | arrays key ev | array key | | |
| 5 | if the specified key exists in | array kay avist() | array kay axists() | array kave aviete() | anays_kcy_cx | anay_key | | |
| 5 | the array? | | allay_key_exists() | allay_keys_exists() | ista() | arrists() | | |
| | There are three different kind | Numerie errou String | Numerie errou | Numaria array | Ists() | _exists() | | |
| 6 | | orrow | | Associative array, | Const array, | orrow | | |
| 0 | - f | Multidimensional | Associative array, | Multidimensional | array | Δssociati | | |
| | OF arrays: | internetional | Dimensional array | | array, | | | |
| _ | what runction computes the | 1:66 | 1:66 | 1:00 | 1:66 | array_ulf | | |
| / | | array_diff | diff_array | arrays_diff | diff_arrays | c. | | |
| | difference of arrays? | | | | | f | | |

| | What functions count | | | | | |
|----|--------------------------------|-----------------------|--------------------|---------------------|-----------------|-----------|
| 8 | | count | Sizeof | Array_Count | Count_array | count |
| | elements in an array? | | | | | |
| | What array will you get if | An array with | An array with | An array with | An array with | An array |
| 9 | you convert an object to an | properties of that | properties of that | properties of that | keys of that | with |
| | array? | object as the array's | array as the | object as the Key | object as the | propertie |
| | Which of the following | | | Microsoft SQL | None of the | None of |
| 10 | DBMSs do not have a native | MySQL | IBM DB/2 | | | |
| | PHP extension? | | | Server | above | the above |
| | In PHP in order to access | mysqlconnect() | mysql-connect() | mysql_connect() | sql_connect() | |
| 11 | MySQL database you will | | | | | mysql_co |
| | use: | function | function | function | function | nnect() |
| | Which of the following is not | | | | CURRENT_D | CURRE |
| 12 | | AVG | SUM | MAX | | NT_DAT |
| | an SQL aggregate function? | | | | ATE() | E() |
| | What does the DESC | It causes the dataset | It causes rows | It causes rows with | It causes rows | It causes |
| 13 | keyword do in the following | returned by the query | with the same ID | the same ID to be | to be sorted by | rows with |
| | query? SELECT *FROM | to be sorted in | to be sorted by | sorted by NAME in | NAME first | the same |
| | The statement is | | | | REMOVE | DROP |
| 14 | | DROP TABLE | DELETE TABLE | DEL TABLE | | |
| | used to delete a table. | | | | TABLE | TABLE |
| | What will happen at the end | The contents of | The contents of | The contents of | The database | The |
| 15 | of the following sequence of | OTHERTABLE will | both | OTHERTABLE | will remain | database |
| | SQL commands?BEGIN | be deleted | OTHERTABLE | will be deleted, as | unchanged | will |
| | Use the to delete | | | TRUNCATE | REMOVE | TRUNC |
| 16 | the data inside the table, and | DROP TABLE | DELETE TABLE | | | ATE |
| | not the table itself? | | | TABLE | TABLE | TABLE |
| | How many error levels are | | | | | |
| 17 | | 14 | 15 | 16 | 17 | 16 |
| | available in PHP? | | | | | |

| | What is the description of | | | | Fatal Compile- | Fatal run- |
|----|--------------------------------|------------------------|-------------------|---------------------|------------------|------------|
| 18 | | Fatal run-time error | Near-fatal error | Compile-time error | | |
| | Error level E_ERROR? | | | | time error | time error |
| | Which version of PHP was | | | | | |
| 19 | added with Exception | PHP 4 | PHP 5 | PHP 5.3 | PHP 6 | PHP 5 |
| | handling? | | | | | |
| | How many methods are | | | | | |
| 20 | available for the exception | 5 | 6 | 7 | 8 | 7 |
| | class? | | | | | |
| | Which of the following | throws new | throw new | | new throws | throw |
| 21 | statements invoke the | | | new Exception(); | | new |
| | exception class? | Exception(); | Exception(); | | Exception(); | Exceptio |
| | Which one of the following | Returns the message | Returns the | Returns the | Returns the | Returns |
| 22 | is the right description for | if it is passed to the | message if it is | message if it is | message if it is | the |
| | the method getMessage() ? | constructor. | passed to the | passed to the file. | passed to the | message |
| | | | Source PHP | | Source PHP | Standard |
| 23 | What does SPL stand for? | Standard PHP Library | | Standard PHP List | | PHP |
| | | | Library | | List | Library |
| | How many predefined | | | | | |
| 24 | exceptions does SPL provide | 13 | 14 | 15 | 16 | 13 |
| | access to? | | | | | |
| | Which of the following | ignore_repeated_error | ignore_repeat_err | repeatedly_ignore_ | repeated_error | ignore_re |
| 25 | statements causes PHP to | | | | | peated_er |
| | disregard repeated error | S | or | error | _ignore | rors |
| | Which of the following | | | | | |
| 26 | magic constant of PHP | _LINE_ | _FILE_ | _FUNCTION_ | _CLASS_ | _FILE_ |
| | returns full path and | | | | | |
| | Which of the following | | | | | array_rev |
| 27 | function can be used to get | array_reverse() | array_search() | array_shift() | array_slice() | |
| | an array in the reverse order? | | | | | erse() |

| | Which of the following | | | | | array_slic |
|----|---------------------------------|----------------------------------|-------------------|---------------------|----------------|------------|
| 28 | function returns selected | array_reverse() | array_search() | array_shift() | array_slice() | |
| | parts of an array? | | | | | e() |
| | Which of the following | | | | | file_exist |
| 29 | function is used to check if a | Fopen() | fread() | filesize() | file_exist() | |
| | file exists or not? | | | | | 0 |
| | Which of the following is an | | | | | \$_COOK |
| 30 | associative array of variables | \$GLOBALS | \$_SERVER | \$_COOKIE | \$_SESSION | |
| | passed to the current script | | | | | IE |
| | Which one of the following | | | | | |
| 31 | databases has PHP supported | Oracle Database | SQL | SQL+ | MySQL | MySQL |
| | almost since the beginning? | | | | | |
| | The updated MySQL | | | | | |
| 32 | extension released with PHP | MySQL | mysql | mysqli | mysqly | mysqli |
| | 5 is typically referred to as | | | | | |
| | Which one of the following | extension=php_mysql | extension=mysql.d | extension=php_my | extension=my | extension |
| 33 | lines need to be | | | | | =php_my |
| | uncommented or added in | i.dll | 11 | sqli.dl | sqli.dl | sqli.dll |
| | Which one of the following | CREATE TABLE | CREATE | CREATE | CREATE | CREATE |
| 34 | statements is used to create a | table_name | table_name | table_name | TABLE | TABLE |
| | table? | (column_name | (column_type | (column_name | table_name | table_na |
| | Which one of the following | <pre>\$mysqli=select_db('d</pre> | mysqli=select_db(| mysqli- | \$mysqli- | \$mysqli- |
| 35 | statements can be used to | | | >select_db('databas | >select_db('da | >select_d |
| | select the database? | atabasename'); | 'databasename'); | ename'); | tabasename'); | b('databa |
| | Which one of the following | | | mysqli_connect_err | mysqli_conne | mysqli_c |
| 36 | methods can be used to | connect_errno() | connect_error() | | | onnect_er |
| | diagnose and display | | | no() | ct_error() | rno() |
| | If there is no error, then what | | | | | Empty |
| 37 | will the error() method | TRUE | FALSE | Empty String | 0 | |
| | return? | | | | | String |

| 38 | Which one of the following methods is responsible for sending the query to the | query() | send_query() | sendquery() | query_send() | query() |
|----|--|------------------------------|------------------------------|----------------------------------|---------------------------------------|-----------------------------------|
| 39 | Which one of the following methods recuperates any memory consumed by a | destroy() | remover() | alloc() | free() | free() |
| 40 | Which of the methods are used to manage result sets using both associative and | get_array() and get_row() | get_array() and get_column() | fetch_array() and fetch_row() | fetch_array() and fetch_column(| fetch_arr ay() and fetch_ro |
| 41 | Which one of the following method is used to retrieve the number of rows affected by | num_rows() | affected_rows() | changed_rows() | new_rows() | affected_ rows() |
| 42 | Which method retrieves each row from the prepared statement result and assigns | get_row() | fetch_row() | fetch() | row() | fetch() |
| 43 | Which one of the following methods is used to recuperating prepared | end() | finish() | final() | close() | final() |

Register Number:

[15CSU602]

KARPAGAM ACADEMY OF HIGHER EDUCATION

Eachanari, Coimbatore-641021. (For the candidates admitted from 2015 onwards) B.Sc COMPUTER SCIENCE FIRST INTERNAL EXAMINATION – JAN 2018 Sixth Semester WEB TECHNOLOGY

Date & Session: Class :III B.Sc(Cs) **Duration :** 2 Hours **Maximum :** 50 Marks

PART - A (20 * 1 = 20 Marks)(Answer all the Questions) 1. is the tag used to define the definition terms a. dl **b. dt** c. dd d. dterms 2. defines a table footer within a table b. <HEAD> c. <THEAD> d. <HEADER> a. <TFOOT> 3. ______ attribute is used to divide table into columns a. group **b. colgroup** c. rowgroup d. tablegroup 4. Which is not a type of list in html? a. ordered list b. unordered list c. definition list d. graphics list 5. The data of definition list is provided using tag a. <DL> b. c. <DT> d. <DD> 6. _____ is the tag used to define horizontal line **a. hr** b. hrr c. br d. brr 7. ______ is the tag used to emphasized the text a. <emphasize> b. <emp> c. d. <emph> 8. Select the HTML tag for inserting a line break? **a.
** b. <lb> c. <break> d.br 9. Which of these tags are all tags? a. <head><tfoot> b. <thead><body> c. d. <tt> 10. Tag for using graphic is IMG followed by a. SRC b.RCS c. CRS d. SCR 11. Choose the HTML tag to make a text italic a. $\langle ii \rangle$ b. $\langle italics \rangle$ c. $\langle i \rangle$ d. $\langle it \rangle$ 12. A website can be made interactive with the help of a. tables b. frames c. text d.forms 13. DHTML is _____ a. Dynamic HTML b. Detailed HTML c. Discontinued HTML d. Dead HTML 14. is used to clear the contents of the form a submit b clear c. reset d. post 15. The *<*STYLE> tag must be present the body of the document a. bottom b. within c. before d. after 16. How many spaces should be provided in writing a HTML tag? a.2 b. 1 c. 3 d. No restriction

| 17De | efines the docu | iment's body | | | | |
|--|---|------------------------|--|---|----|----------------------|
| a. <body></body> | b. <bg></bg> | c. <bodydoc></bodydoc> | d. < body statement> | | | |
| 18. How can ye | 18. How can you make an e-mail link? | | | | | |
| a. <a href:<="" td=""><td>=''mailto:xxx</td><td>@yyy''></td><td>b. </td><td></td> | =''mailto:xxx | @yyy''> | b. | | | |
| c. <mail>xx</mail> | xx@yyy <td>l></td> <td>d. <mail href="xxx@yyy"></mail></td> <td></td> | l> | d. <mail href="xxx@yyy"></mail> | | | |
| 19. Choose the | correct HTM | L to left-align th | he content inside a tablecell | | | |
| a. <td a<="" td=""><td>align="left"></td><td>b. <td leftalign<="" td=""><td>c.</td><td>d. <tdleft></tdleft></td></td></td></td> | <td>align="left"></td> <td>b. <td leftalign<="" td=""><td>c.</td><td>d. <tdleft></tdleft></td></td></td> | align="left"> | b. <td leftalign<="" td=""><td>c.</td><td>d. <tdleft></tdleft></td></td> | <td>c.</td> <td>d. <tdleft></tdleft></td> | c. | d. <tdleft></tdleft> |

20. _____ attribute is used to provide the thickness of borders around each table cell a. bgcolor b. thick **c. border** d. light

PART – B (3 * 10 = 30 Marks) (Answer all the Questions)

21.a.(i) Differentiate ordered list, unordered list and definition list with example program.

Unordered Lists:

An unordered list is a collection of related items that have no special order or sequence. The most common unordered list on the Web is a collection of hyperlinks to other documents.

(E.G):-

The following example list is created by using $\langle u \rangle$ tag. Each item in the list is marked with a butllet. The bullet itself comes in three flavors: squares, discs, and circles. The default bullet displayed by most web browsers is the traditional full disc.

One Movie list is given below:

<center> <h2>Movie List</h2> </center> Independence Day Horror Titanic Ghost in the ship This will produce following result:

Movie List

- Independence Day
- Horror
- Titanic
- Ghost in the ship

Type attribute can be used to specify the type of bullet you like. By default it is a disc. Following are the possible way:

| <ul square''="" type=""> | <ul disc''="" type=""> | <ul type="circle"> |
|--|--|--|
| Hindi English Maths Physics | Hindi English Maths Physics | Hindi English Maths Physics |

Ordered Lists:

The typical browser formats the contents of an ordered list just like an unordered list, except that the items are numbered instead of bulleted. The numbering starts at one and is incremented by one for each successive ordered list element tagged with

This list is created by using *<*ol> tag. Each item in the list is marked with a number.

One Movie list is given below:

```
<center>
<h2>Movie List</h2>
</center>
Independence Day 
Horror 
Titanic
Ghost in the ship
```

This will produce following result:



Definition Lists:

HTML and XHTML also support a list style entirely different from the ordered and unordered lists we have discussed so far - definition lists . Like the entries in a dictionary or encyclopedia, complete with text, pictures, and other multimedia elements, the Definition List is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

- <dl> Defines the start of the list
- $\langle dt \rangle$ A term
- <dd>- Term definition
- </dl> Defines the end of the list

Example:

<dl>

<dt>HTML</dt> <dd>This stands for Hyper Text Markup Language</dd> <dt>HTTP</dt> <dd>This stands for Hyper Text Transfer Protocol</dd> </dl>

This will produce following result:

HTML

This stands for Hyper Text Markup Language

HTTP

This stands for Hyper Text Transfer Protocol

(ii) DESCRIBE THE HISTORY OF HTML

- 1960s-GML (Generalized Markup Language) was developed by Charles Goldfarb, Edward Musher and Raymond Lorie at IBM for organizing vast amount of documentation.
- 1978- ANSI took basics of GML and fashioned nationwide standard called GCA.
- Six years later ISO began to work on new global version called SGML (Standard Generalized Markup Language).
- 1989-Tim Berners Lee a physicist at CERN(Centre European pour la Recherche Nucleaire) created a method for the scientists to share papers. In the same year a memo proposing hypertext system was released.
- Late 1990- Tim Berners Lee specified HTML and wrote a browser and server software.
- Late 1991-public description of HTML was released containing 18 elements.
- 1995-HTML 2.0.
- Jan 1997-HTML 3.2 (W3C Recommendation).
- Dec 1997- HTML 4.0 (W3C).
- April 1998 HTML 4.0 got minor edits.
- Dec 1999-HTML 4.0.1.
- Jan 2008-HTML 5.

- Mean while two languages related to hyper text where developed.
 - ✓ DHTML→combination of HTML 4, Cascading Style Sheets and JavaScript to create dynamic Web pages. These work on basis of DOM(Document Object Model) to change the content of the web page even if it's loaded into the browser.
 - ✓ XML→XML stands for Extensible Markup Language using which we can create our own markup languages. It is developed by W3C.

(**OR**)

b. EXPLAIN IN DETAIL WITH EXAMPLE:

i) Putting background image in table

- A background image can be added to either to the whole table or individual cells.
- The attribute for displaying a background image in a table or table cell is BACKGROUND and the URL of the image will be assigned to this attribute.
- Here the individual cells may be the standard or header cells.

(E.G):-

```
<html>
<body>
<table
       background="C:\Documents
                                     Settings\All
                                                 Users\Documents\My
                               and
Pictures\Sample Pictures\Water lilies.jpg" border=5>
<font color="White">
 <font color="White">
 ISBN</font>
 <font color="White">Title</font>
 <font color="White">Price</font>
 3476896
 My first HTML
 $53
 </body>
</html>
Output:
```



ii) Heading across two or more columns in table

- Sometimes if there is a need to display a single heading over two columns then it's possible by making a single cell to span two or more columns.
- That's made possible by the use of COLSPAN attribute. When it is used with th or td tag then we can span a number of columns assigned to the colspan attribute.

```
• (E.G):-
```

```
Month
Savings
January
$100
February
$100
Sum: $180
Output
```


iii) Adding borders in table

- Overall a table can be highlighted using a border.
- To provide a border for a table the attribute border can be used in table tag.
- A numerical value can be assigned to the border attribute, based on the numerical value of the border the thickness of the table value varies.
- The attributes related to the colors of table border are
 - iii) BORDERCOLOR- applies a single color to the whole table border and the color can be applied through color name, hexadecimal value and rgb component value.
 - iv) BORDERCOLOR-DARK- applies a color to the darker parts of the table border.
 - v) BORDERCOLOR-LIGHT- applies a color to the lighter parts of the table border.

```
(E.g):-

Month
Savings

Savings

January

$100

$100

$100
```

The output will be a table with a border thickness of one point

iv) Create tables

- Tables are very useful to arrange in HTML and they are used very frequently by almost all web developers.
- Tables are just like spreadsheets and they are made up of rows and columns.
- A table is created by using tag. Inside element the table is written out row by row.
- A row is contained inside a tag . which stands for table row.
- Each cell is then written inside the row element using a tag . which stands for table data.

(E.G):-

```
Cell1
Cell2
Cell2
Cell3
Cell3
```

This will produce following result:

| Cell1 | Cell2 |
|-------|-------|
| Cell3 | Cell4 |

22.a. i)Explain Structure of HTML with examples

The basic structure of a HTML document and the description of the tags present in it are as below.

- **<!DOCTYPE>-** Defines the document type.
- **<HTML>-** indicates the browser that it is a HTML document and **</HTML>** tells the browser that HTML document is completed.
- **<HEAD>-** container for all the head elements.

<!DOCTYPE html> <html> <head><title>HTML</title></head> <body> <h1>MY FIRST PROGRAM</h1>

| OUTPUT: | |
|------------------|--|
| MY FIRST PROGRAM | |
| SAMPLE | |

- \checkmark **<TITLE>-** defines title of the -engine results. (E.g):- **<title>**
- √
- ✓ <STYLE>- style information for an HTML document is defined here (i.e):how HTML elements should render in a browser is defined.

(E.G):-<style> h1{color:red;} p{color:blue;} </style>

 ✓ <BASE>- specifies the base URL/target for all relative URLs in a document. Contains either a href attribute or target attribute or sometimes both. All the elements at the body of the html document will be referring the base tag's.

(E.G):-<base href="www.w3schools.com/images"> </base>



In the above example if the url in img tag is not able to give the image then the base url will be referred if present then the image will be displayed.

✓ <LINK>- defines the relationship between a document and an external resource.

(E.G):-

k rel="stylesheet" type="text/css" href="sample.css">

In the above example a document which is external called sample.css is linked. It is a cascading style sheet file linked using href attribute. Apart from that there are two more attributes called type-which specifies the type of file linked and rel-specifies the type of link between the current document and the external file. Most probably the style sheet files will be linked in to a HTML document.

✓ <META>- provides metadata about the HTML document. Metadata will not be displayed on the page. Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

(E.G):-

<meta charset="UTF-8"> <meta name="description" content="Example"> <meta name="keywords" content="HTML"> <meta name="author" content="BSCCS">

• **<BODY>-** contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc. This is the place where the actual content of the web page required for display will be given.

The attributes of the body tag are

- ✓ alink- The alink attribute specifies the color of an active link in a document. The default color of the active link in a document will be red.
- ✓ background- specifies a background image for a document using the image's URL. Proper image that is image with correct resolution should be chosen to display and if not then the display image will be inconvenient to the user/client.
- ✓ bgcolor- specifies the background color of a document. For specifying the color of the document the either the name of the particular color can be used or the hexadecimal value for a color can be used or rgb value can be used.
- ✓ link- specifies the default color of unvisited links in a document. The default color of an unvisited in a HTML document will be blue.
- ✓ text- used to specify the color of text displayed in a document. Default color of the body text is black or it may differ based on the web browser of the client.
- ✓ vlink- used to specify the color of visited links displayed in a document. All the visited links in a HTML document will be pink in color.

ii) Create a Resume in HTML document using all basic tag

<html> <head> <title>Resume</title> s.no edu details <h2>Area of interest</h2> $\langle ul \rangle$ C C++ java Python </body> </html>

(**OR**)

b. i) explain in detail about Frames creation with example

- HTML frames allow authors to present documents in multiple views, which may be independent windows or sub windows.
- In other words if the web page developer want to display the contents in multiple sections under one window then this frames are useful.
- Also the frames can be used to navigate between pages without leaving the current page.
- The process of dividing the window into multiple sections involves the use of two tags / elements.
- They are <frameset> and <frame> elements.
- The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

- The attributes inside the frameset element are cols and rows. Both of them specify the number of columns and rows that a frameset should contain. The values for these attributes will be in pixels, % and *.
- The contents inside divided columns and rows in a frameset are given using the frame element.
- The attributes of frame tag are as follows.
 - ✓ The frameborder attribute specifies whether or not to display a border around a frame. The input value for this attribute is numerical.
 - \checkmark The src attribute specifies the URL of the document to show in a frame.
 - ✓ The longdesc attribute specifies a URL to a page that contains a long description of the content of a frame. It is not supported today. As an alternative a link can be introduced in the particular frame which takes the user to another oage that contains the long text.
 - ✓ The marginheight attribute specifies the height between the content and the top and bottom of the frame, in pixels.
 - ✓ The marginwidth attribute specifies the width between the content and the left and right sides of the frame, in pixels.
 - ✓ The name attribute is used to reference the element in a JavaScript, or to act as a target for a link.
 - ✓ The noresize attribute specifies that a <frame> element cannot be resized by the user. By default, each <frame> in a <frameset> can be resized by dragging the border between the frames. However, this attribute locks the size of a frame.
 - \checkmark The scrolling attribute specifies whether or not to display scrollbars in a \langle frame \rangle .

** The <noframes> element can be used to link to a non-frameset version of the web site or to display a message to users that frames are required.

```
(E.g):-
```

```
<frameset cols="25%,*,25%">
```

```
<frame src="frame_a.htm">
```

```
<frame src="frame_b.htm">
```

```
<frame src="frame_c.htm">
```

```
</frameset>
```

Output:

| Frame A | Frame B | Frame C |
|---|---------|---------|
| Note: The frameset, frame, and noffames elements are not supported in HTMLS. | | |
| | | |

ii) Write short note on Menus with example program

- Menus in html are used to display a list of items in a webpage. The menu can be otherwise called as drop down list.
- The <select> element is used to create a drop-down list.
- Mainly used to display list of items in commercial sites, job sites etc.
- The attributes under this select are as follows
 - ✓ Disabled-The disabled attribute is a Boolean attribute. When present, it specifies that the drop-down list should be disabled. A disabled drop-down list is unusable and un-clickable. The disabled attribute can be set to keep a user from using the drop-down list until some other condition has been met (like selecting a checkbox, etc.). Then, a JavaScript can remove the disabled value, and make the drop-down list usable.
 - ✓ **Multiple**-The multiple attribute is a boolean attribute. When present, it specifies that multiple options can be selected at once. Selecting multiple options vary in different operating systems and browsers:
 - For windows: Hold down the control (ctrl) button to select multiple options
 - > For Mac: Hold down the command button to select multiple options
 - ✓ Size-The size attribute specifies the number of visible options in a drop-down list. If the value of the size attribute is greater than 1, but lower than the total number of options in the list, the browser will add a scroll bar to indicate that there are more options to view.
 - ✓ Name- The name attribute specifies the name for a drop-down list. The name attribute is used to reference elements in a JavaScript, or to reference form data after a form is submitted.
- The items for a menu can be displayed using the OPTION tag. The attributes for option tag are
 - ✓ Disabled- The disabled attribute is a boolean attribute. When present, it specifies that an option should be disabled. A disabled option is unusable and un-clickable.
 - ✓ Label- The label attribute specifies a shorter version of an option. The shorter version will be displayed in the drop-down list.

- ✓ Selected-The selected attribute is a boolean attribute. When present, it specifies that an option should be pre-selected when the page loads. The pre-selected option will be displayed first in the drop-down list.
- ✓ Value- The value attribute specifies the value to be sent to a server when a form is submitted. The content between the opening <option> and closing </option> tags is what the browsers will display in a drop-down list. However, the value of the value attribute is what will be sent to the server when a form is submitted.

<select>

<option value="volvo">Volvo</option>
<option value="saab">Saab</option>
<option value="mercedes">Mercedes</option>
<option value="audi">Audi</option>
</select>

23.a. i)Write short note on Creating Forms with Radio Button, Text box and Check box with example.

- ✓ Concept that is used to make the website or webpage to be interactive is called as forms.
- ✓ The main idea here is to get the input data from the users of the website/page and respond with a result.
- \checkmark The controls that are used to get input may be textboxes, password boxes, checkboxes, radio buttons etc.

Creating a Form:

- \checkmark Basically the form tag/element contains 2 attributes, they are action and method.
- ✓ The action attribute specifies the url of the CGI script in the server. The input data given are processed using this CGI script.
- ✓ This Common Gateway Interface (CGI) is a common method used to generate dynamic content in the web pages. This CGI when implemented provides an interface between the web page that generates contents and the web server.
- \checkmark The method attribute has two values they are GET and POST.
 - GET- in get method the form data is appended into the URL. The length of the URL is limited to 3000 characters. This method is not suitable for sensitive data. When the result page of form submitted is to be stored using bookmarks then it is possible in this method. Query strings are good example for this GET.
 - POST- appends form data inside the body of HTTP request and it's not shown in the URL. There is no size limitations for the data submitted in this method. These

kind of pages cannot be bookmarked. The new email id creation is a good example for this POST method.

(E.G):-

<html>

<body>

<form action="demo_form.asp">

```
First name: <input type="text" name="FirstName" value="Mickey"><br>
```

```
Last name: <input type="text" name="LastName" value="Mouse"><br>
```

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

```
</form>
```

```
Click the "Submit" button and the form-data will be sent to a page on the server called "demo_form.asp".
```

</body>

</html>

** In the Below example the default form method is get where the data is carried using the URL value. The page where the CGI script runs is also specified in the action attribute. Two text boxes with a submit button is displayed. When the submit button is clicked the data is posted into the URL and it's processed in the action URL.

Output:

| First name: Last name: | Mickey | | 7 | | |
|------------------------------|-----------------------------|-------------------------|--------------|----------------|-------------|
| Submit | | | 1 | | |
| Click the "S server calle | ubmit" butto d "demo for | n and the fo m.asp". | ərm-data wil | l be sent to a | page on the |
| | | ····· | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

iii) Explain in detail about Working with Forms.

- ✓ Concept that is used to make the website or webpage to be interactive is called as forms.
- ✓ The main idea here is to get the input data from the users of the website/page and respond with a result.
- \checkmark The controls that are used to get input may be textboxes, password boxes, checkboxes, radio buttons etc.

Creating a Form:

- \checkmark Basically the form tag/element contains 2 attributes, they are action and method.
- ✓ The action attribute specifies the url of the CGI script in the server. The input data given are processed using this CGI script.

- ✓ This Common Gateway Interface (CGI) is a common method used to generate dynamic content in the web pages. This CGI when implemented provides an interface between the web page that generates contents and the web server.
- \checkmark The method attribute has two values they are GET and POST.
 - GET- in get method the form data is appended into the URL. The length of the URL is limited to 3000 characters. This method is not suitable for sensitive data. When the result page of form submitted is to be stored using bookmarks then it is possible in this method. Query strings are good example for this GET.
 - POST- appends form data inside the body of HTTP request and it's not shown in the URL. There is no size limitations for the data submitted in this method. These kind of pages cannot be bookmarked. The new email id creation is a good example for this POST method.

```
(E.G):-
<html>
<body>
<form action="demo_form.asp">
First name: <input type="text" name="FirstName" value="Mickey"><br>
Last name: <input type="text" name="LastName" value="Mouse"><br>
<input type="submit" value="Submit">
</form>
Click the "Submit" button and the form-data will be sent to a page on the server called "demo_form.asp".
```

</body>

</html>

** In the Below example the default form method is get where the data is carried using the URL value. The page where the CGI script runs is also specified in the action attribute. Two text boxes with a submit button is displayed. When the submit button is clicked the data is posted into the URL and it's processed in the action URL.

Output:



Working with menus:

• Menus in html are used to display a list of items in a webpage. The menu can be otherwise called as drop down list.

- The <select> element is used to create a drop-down list.
- Mainly used to display list of items in commercial sites, job sites etc.
- The attributes under this select are as follows
 - ✓ Disabled-The disabled attribute is a Boolean attribute. When present, it specifies that the drop-down list should be disabled. A disabled drop-down list is unusable and un-clickable. The disabled attribute can be set to keep a user from using the drop-down list until some other condition has been met (like selecting a checkbox, etc.). Then, a JavaScript can remove the disabled value, and make the drop-down list usable.
 - ✓ **Multiple**-The multiple attribute is a boolean attribute. When present, it specifies that multiple options can be selected at once. Selecting multiple options vary in different operating systems and browsers:
 - ➢ For windows: Hold down the control (ctrl) button to select multiple options
 - > For Mac: Hold down the command button to select multiple options
 - ✓ Size-The size attribute specifies the number of visible options in a drop-down list. If the value of the size attribute is greater than 1, but lower than the total number of options in the list, the browser will add a scroll bar to indicate that there are more options to view.
 - ✓ Name- The name attribute specifies the name for a drop-down list. The name attribute is used to reference elements in a JavaScript, or to reference form data after a form is submitted.
- The items for a menu can be displayed using the OPTION tag. The attributes for option tag are
 - ✓ Disabled- The disabled attribute is a boolean attribute. When present, it specifies that an option should be disabled. A disabled option is unusable and un-clickable.
 - ✓ **Label-** The label attribute specifies a shorter version of an option. The shorter version will be displayed in the drop-down list.
 - ✓ **Selected-**The selected attribute is a boolean attribute. When present, it specifies that an option should be pre-selected when the page loads. The pre-selected option will be displayed first in the drop-down list.
 - ✓ Value- The value attribute specifies the value to be sent to a server when a form is submitted. The content between the opening <option> and closing </option> tags is what the browsers will display in a drop-down list. However, the value of the value attribute is what will be sent to the server when a form is submitted.

<select>

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="mercedes">Mercedes</option>

<option value="audi">Audi</option>

</select>

**displays a drop down list with the names of various cars.

(**OR**)

b. i) What is DHTML? and explain How will you align the message at the center of the page in DHTML

Dynamic HTML, or DHTML, is an <u>umbrella term</u> for a collection of technologies used together to create interactive and animated <u>web sites</u> by using a combination of a static <u>markup language</u> (such as <u>HTML</u>), a <u>client-side scripting</u> language (such as <u>JavaScript</u>), a presentation definition language (such as <u>CSS</u>), and the <u>Document Object Model</u>.

DHTML allows scripting languages to change <u>variables</u> in a web page's definition language, which in turn affects the look and function of otherwise "static" HTML page content, after the page has been fully loaded and during the viewing process. Thus the dynamic characteristic of DHTML is the way it functions while a page is viewed, not in its ability to generate a unique page with each page load.

By contrast, a <u>dynamic web page</u> is a broader concept, covering any web page generated differently for each user, load occurrence, or specific variable values. This includes pages created by client-side scripting, and ones created by <u>server-side scripting</u> (such as <u>PHP</u>, <u>Perl</u>, <u>JSP</u> or <u>ASP.NET</u>) where the web server generates content before sending it to the client.

Heading and Horizontal Line:

The <hr> tag defines a thematic break in an HTML page. The <hr> element is used to separate content (or define a change) in an HTML page.

In HTML5, the <hr> tag defines a thematic break. In HTML 4.01, the <hr> tag represents a horizontal rule.

However, the <hr> tag may still be displayed as a horizontal rule in visual browsers, but is now defined in semantic terms, rather than presentational terms. All the layout attributes are removed in HTML5. Use CSS instead.

An example for the Horizontal Rule is given below.



Add some text for the TITLE tag. Add an H1 heading, with some text between the two H1 tags. Now add a paragraph of text below the H1 heading. To create an Inline Style, click after the first H1. Tap the space bar on your keyboard. Type the following:

<H1 style=text-align:center>

Make sure there is no space after the colon and before the word 'center'.

Your HTML code should look something like ours below:



(One thing British English users should note is the two different spelling of the word 'centre'. American English is used in HTML, so it's 'er' rather than 're'.)

When you have finished typing the code, view the results in your browser. You should see that the heading is centered on your page:



When a hr tag is used after this aligned heading then the output will be as follows.

| Styles - Windows Internet Explorer | |
|---|-----|
| 🕞 🔄 🖉 ML\newpage.html 👻 🤸 🔀 Bing | - 9 |
| 🔶 Favorites 🛛 🎭 🚰 Google 💓 Suggested Sites 👻 🔊 Get More Add-ons 👻 | >> |
| Styles | |
| | ~ |
| Centred Heading | |
| Style sheets can be used to format text on a web page. | |
| | |
| | |
| | |
| | |
| | |

HIDDEN MESSAGE:

To display a hidden message in a Web page then the visibility property can be set to hidden in style attribute. This can be done in any element available in html. Later that hidden message can be retrieved using a java script coding with the help of that element's name.

```
(E.G):-
<!DOCTYPE html>
<html>
<body>
This is a p element.
<button type="button" onclick="myFunction()">Hide content of p</button>
<script>
function myFunction()
{
    document.getElementById("myP").style.visibility="hidden";
    }
</script>
</body>
</html>
```

The above coding yields the following output.

```
This is a p element.
Hide content of p
```

When the button is clicked the text will be disappearing above.

ii)How will you create the moving box in DHTML with example?

MOVING BOXES:

To create an effect of moving boxes in a web page we have to create a set of boxes using div tag and then change their positions accordingly. Such that the end position of one box will be the starting position of another box.

Apart from that the size and color of all the boxes should be same to create such an effect.

(E.G):-

<div id="d1" style="position:Absolute;left:150;top:180;width:120;height:60"></div> <div id="d1" style="position:Absolute;left:270;top:180;width:120;height:60"></div> Here by the above coding two boxes will be created but they will be looking like moving.