



© PSS Central Institute of Vocational Education, 2012

Copyright protects this publication. Except for purposes permitted by the Copyright Act, reproduction, adaptation, electronic storage and communication to the public are prohibited without prior written permission.

# Student Details

**Student Name:** \_\_\_\_\_

**Student Roll Number:** \_\_\_\_\_

**Batch Start Date:** \_\_\_\_\_

## Acknowledgements

The following partners were instrumental in providing the content:

1. Accenture India's Corporate Citizenship Program (Skills 4 Life) has provided the content material for English and have commissioned and developed as well as provided access to their implementing partners (Dr. Reddy's Foundation and QUEST Alliance).
2. The Wadhvani Foundation team involved in designing and building this curriculum and content include Ms. Sonia Kakkar, Mr Karthik Chandru, Ms. Toral Veecumsee, Ms. Rekha Menon, Mr. Ajay Goel and Mr. Austin Thomas.
3. The PSSCIVE's team was involved in guidance and editing the content.
4. In addition, various public domain sources have been leveraged to create materials and illustrations across module. The contributions of all these sources is gratefully acknowledged and recognized.

<b>Acknowledgements .....</b>	<b>4</b>
<b>Preface.....</b>	<b>6</b>
<b>Session 1: Introduction to Web Design &amp; HTML .....</b>	<b>8</b>
<b>Session 2: HTML Elements &amp; Attributes - The Basics.....</b>	<b>11</b>
<b>Session 3: HTML Elements &amp; Attributes - Text formatting.....</b>	<b>15</b>
<b>Session 4: HTML Elements &amp; Attributes - Lists and Links .....</b>	<b>19</b>
<b>Session 5: HTML Elements &amp; Attributes - Tables, Images and Multimedia.....</b>	<b>23</b>
<b>Session 7: HTML Elements &amp; Attributes - Forms and FRAMES .....</b>	<b>27</b>
<b>Session 9: HTML Elements &amp; Attributes - Metatags, Doctypes and Validation....</b>	<b>31</b>
<b>Session 10: Introduction to CSS .....</b>	<b>34</b>
<b>Assignment .....</b>	<b>38</b>
<b>Session 11: Overview of XML &amp; XHTML .....</b>	<b>40</b>
<b>Session 12: Web Site Design Scripting Tools.....</b>	<b>44</b>
<b>Session 13: Writing Javascript Program .....</b>	<b>48</b>
<b>Session 14: Javascript Syntax .....</b>	<b>52</b>
<b>Session 15: Data Types.....</b>	<b>56</b>
<b>Session 16: Logical Operators and ConditionAL Statements.....</b>	<b>59</b>
<b>Session 17: Functions .....</b>	<b>64</b>
<b>Session 18: Loops .....</b>	<b>67</b>
<b>Session 19: Arrays and Objects.....</b>	<b>72</b>
<b>Session 20: Overview of Dhtml, Ajax and JQuery.....</b>	<b>77</b>
<b>Session 21: Introduction to VBscript .....</b>	<b>80</b>
<b>Session 22: Syntax and Variables .....</b>	<b>83</b>
<b>Session 23: Operators and Conditional Statements .....</b>	<b>88</b>
<b>Session 24: Loops .....</b>	<b>93</b>
<b>Session 25: Procedures and Functions.....</b>	<b>96</b>
<b>Session 26: Overview of Iis and Apache Web Server.....</b>	<b>99</b>
<b>Session 27: Overview of ASP.....</b>	<b>103</b>
<b>Session 28: Overview of PHP .....</b>	<b>105</b>

## Preface

The National Curriculum Framework, 2005, recommends that children's life at school must be linked to their life outside the school. This principle makes a departure from the legacy of bookish learning which continues to shape our system and causes a gap between the school, home, community and the workplace.

The student workbook on “**Web Designing Part 1**” is a part of the qualification package developed for the implementation of National Vocational Education Qualification Framework (NVEQF), an initiative of Ministry of Human Resource Development (MHRD), Government of India to set common principles and guidelines for a nationally recognized qualification system covering Schools, Vocational Education and Training Institutions, Technical Education Institutions, Colleges and Universities. It is envisaged that the NVEQF will promote transparency of qualifications, cross-sectoral learning, student-centred learning and facilitate learner's mobility between different qualifications, thus encouraging lifelong learning.

This student workbook, which forms a part of vocational qualification package for student's who have passed Class X or equivalent examination, was created by a group of experts. The IT-ITeS Skill Development Council approved by the National Skill Development Corporation (NSDC) for the IT/ITeS Industry developed the National Occupation Standards (NOS). The National Occupation Standards are a set of competency standards and guidelines endorsed by the representatives of IT Industry for recognizing and assessing skills and knowledge needed to perform effectively in the workplace.

The Pandit Sunderlal Sharma Central Institute of Vocational Education (PSSCIVE), a constituent of National Council of Educational Research and Training (NCERT) in association with Wadhvani Foundation has developed modular curricula and learning materials (Units) for the vocational qualification package in IT/ITes sector for NVEQ levels 1 to 4; level 1 is equivalent to Class IX. Based on NOS, occupation related core competencies (knowledge, skills, and abilities) were identified for development of curricula and learning modules (Units).

This student workbook attempts to discourage rote learning and to bring about necessary flexibility in offering of courses, necessary for breaking the sharp boundaries between different subject areas. The workbook attempts to enhance these endeavours by giving higher priority and space to opportunities for contemplation and wondering, discussion in small groups and activities requiring hands-on-experience. We hope these measures will take us significantly further in the direction of a child-centred system of education outlined in the National Policy of

Education (1986). The success of this effort depends on the steps that school Principals and Teachers will take to encourage children to reflect their own learning and to pursue imaginative and on-the-job activities and questions.

Participation of learners in skill development exercises and inculcation of values and creativity is possible if we involve children as participants in learning, and not as receiver of information. These aims imply considerable change in school routines and mode of functioning. Flexibility in the daily time-table would be a necessity to maintain the rigour in implementing the activities and the required number of teaching days will have to be increased for teaching and training.

## SESSION 1: INTRODUCTION TO WEB DESIGN & HTML

### RELEVANT KNOWLEDGE

You have surfed several websites, exchanged emails, performed online transactions, etc. in the previous sessions. If you want to create websites like the ones you have seen, you need to learn a variety of technologies, tools and techniques.

Web designing refers to the skills that are required to create and manage websites. Web design as a whole refers to designing a user interface, authoring (programming) or a combination of both. Designing a website requires multiple skills. However, it is recommended to master and become skilled in one particular technology once you get a taste of various technologies involved.

Following topics are covered in this module:

- HTML & CSS
- Web Authoring Tools, Microsoft Expression Web
- Overview of Client-side & Server-side scripting, JavaScript & VBScript

### Web Publishing

A number of markup languages such as HTML, DHTML, XML are used to create web based content. Apart from these, software products such as FrontPage, Dream Weaver etc are also used to develop web based content. Web based applications are built using technologies such as ASP (Active Server Pages), JSP (Java Server Pages), PHP (Hypertext Preprocessor), ASP.net etc.

### Understanding HTML

HTML or *Hypertext Markup Language* is a language used for created web pages that can be viewed in a web browser.

- **Hyper** - means active and all over the place! When you use a Web browser, you can move all over the place, without a specified order as to how to access pages/sites.
- **Text** - You will be working with text files only.
- **Markup** - To create a Web page, you will type in the text and then “Mark up” the text.
- **Language** - This is a method of creating something or communicating and in this case, you are creating Web pages using a particular syntax.

HTML elements are the building blocks of a website. Web browsers interpret the tags written in HTML and display a web page.

Hypertext refers to the text that can be linked to another text or page which a user can access. Hypertext is the underlying concept behind the World Wide Web.

HTML is developed and maintained by World Wide Web Consortium abbreviated as W3C. When HTML was first released, there were a handful of tags available and used for basic web page structures. As time evolved, several new tags and attributes were introduced. Popular HTML versions include HTML 4.01 and HTML 5.0.

## Designing Web pages using HTML

All you need is a text editor such as notepad to create a web page using HTML. HTML files have the extension .htm or .html. You can open the web pages using web browsers such as Mozilla Firefox, Internet Explorer, Google Chrome, etc.

## Viewing Source Code

Web browsers have the ability to display the source code used for a web page. You can use this to enhance your knowledge To view the code, do the following:

1. Open the Web Browser and **right-click** anywhere in the web page and select **View Page Source** from the **context** menu similar to the one below:



2. Notice, the source code of the web page is visible with different color codes. The colors have no meaning in terms of function and are only used to differentiate different sets of code.

## Saving a web page

You can save web pages for future reference using the following procedure:

1. Select **File > Save Page As...**

2. Type a file name and click **Save**.

## EXERCISE

Perform the following activities until you are confident:

S.No.	Activities
1.	Download and use different web browsers such as Mozilla Firefox, Google Chrome, Internet Explorer (recent version).

## ASSESSMENT

Answer the following questions:

1. Explain HTML.
2. List any three popular web browsers.
3. Explain the procedure to view source code in Mozilla Firefox.
4. Explain the procedure to view source code in Internet Explorer.
5. Explain the procedure to view source code in Google Chrome.

Fill in the blanks:

1. Acronym for HTTP \_\_\_\_\_.
2. Acronym for HTML \_\_\_\_\_.
3. Acronym for W3C \_\_\_\_\_.
4. Default File extension of an HTML document \_\_\_\_\_ or \_\_\_\_\_.
5. Organization that manages the web standards \_\_\_\_\_.

## SESSION 2: HTML ELEMENTS & ATTRIBUTES - THE BASICS

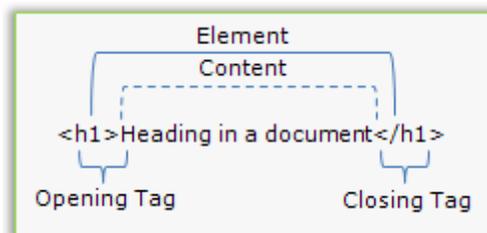
### RELEVANT KNOWLEDGE

When you work with a document, you create headings, sub-headings, tables, etc. to give the document a structure. Similarly in HTML, tags or elements are used to create a web pages' structure.

Look at an example of an online magazine. Online magazines have articles with text content, images, videos, links or continuation to related articles, advertisements, etc. They appear as a single web page to visitors. Have you noticed that the headings are of varying sizes to indicating topics and sub-topics and different from the content? All this formatting is possible using combinations of several HTML elements.

### Elements & Attributes

Elements have content placed between a start tag and an end tag. Tags act as containers for text, images and other types of content. Tags start an opening tag and end with a closing tag, using angle brackets (<>) to define the structure.



**Note:** To learn to use the tags, throughout this document, you will find examples of code displayed within dotted boxes. Copy it in a text editor, save it in your local computer and view using web browsers.

### Writing your First HTML Code

HTML uses tags enclosed in angle brackets; most tags are used in pairs such as `<html></html>`, etc. Following is a simple example of HTML code:

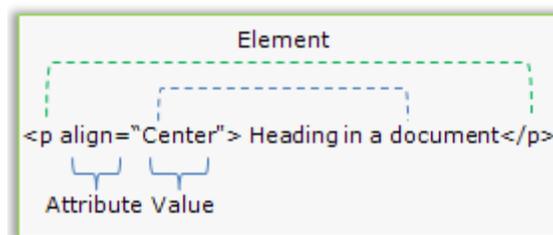
```
<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<p>Hello World!</p>
<!--this is an example of simple HTML code created using Notepad-->
</body>
</html>
```

In the above mentioned code, there are a few HTML tags used. All HTML documents start with `<html>` and end with `</html>`. The rest of the document contains other elements and web page content placed using elements.

- Document type or DOCTYPE is an element used for indicating the version of HTML used in a web page. This is used by the web browser to apply a set of rules for understanding the document structure.
- HTML element indicates the start and end of an HTML document. `<HTML>` indicates the beginning of this HTML document and `</HTML>` indicates the end. HTML is also the root of an HTML document and has two parts, *Head* and *Body* elements.
  - **Head** element represents the header section of a document and contains information about a web page. Head elements start with `<head>` and end with `</head>`. This element includes all other elements that need to be executed or understood before the content (included in the body element) is displayed to the user.
    - Content placed within the `<Title>` and `</Title>` element is displayed as the title of a web page and used by search engines to create indexes of content.
  - **Body** element represents the section that contains body content and is displayed in the web browser. Most HTML elements are used within the body element between `<body>` and `</body>` tags.
- If you want to place hidden text in a web page, you can use the comment element (`<!-->` and `<-->` tags). Text placed using comment element will not be displayed in a web browser but can be seen when viewing the source code. This is useful for developers or users who would like to share their experience in writing HTML code but hide the text from general visitors.

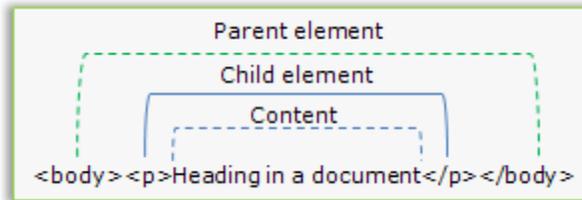
## HTML Attributes

Attributes are used to provide additional information about an element. Attributes contain values and some elements use multiple attributes.

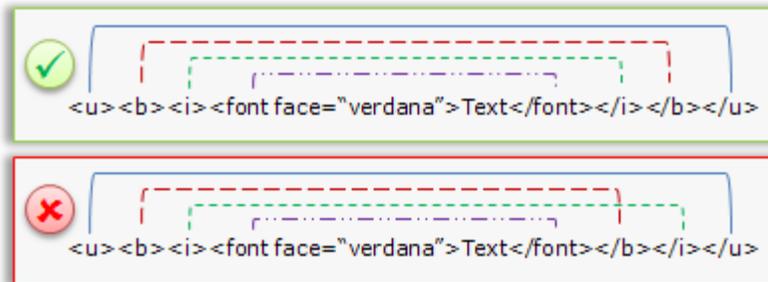


## Nesting

You can nest other elements within elements. Elements nested within another element are referred to as a child elements and the nesting element is referred to as a parent element.



When you nest elements, you need to ensure the elements are used in proper order. Observe the correct way of using opening and closing tags illustrated in the figure below:



## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	<p>Create a HTML document for the following topics:</p> <ol style="list-style-type: none"><li>World Wide Web</li><li>Wireless Application Protocol (WAP)</li><li>Android Operating System</li><li>Firefox Operating System</li></ol> <p>Use the following guidelines:</p> <ul style="list-style-type: none"><li>Use Headings for titles</li><li>Include content within paragraphs (&lt;p&gt;)</li><li>Use comment element to include the source of information</li><li>Use appropriate titles for topics using Title Tag</li></ul>

## ASSESSMENT

Answer the following questions:

1. Explain Markup Language
2. Explain HTML element and attribute with an example each.
3. Explain the purpose of the following HTML elements:
  - a. Head
  - b. Body
  - c. Title
  - d. Paragraph
  - e. Comment

Fill in the blanks:

1. \_\_\_\_\_ tag indicates the version of HTML used.
2. \_\_\_\_\_ tag is used for displaying content in a browser's title bar.
3. \_\_\_\_\_ tag is used for including comments hidden from a web browser.
4. Documents always should start with \_\_\_\_\_ tag and end with \_\_\_\_\_ tag.
5. \_\_\_\_\_ area is treated as header section of a HTML document.
6. \_\_\_\_\_ provides additional information through a name and value in HTML documents.

## SESSION 3: HTML ELEMENTS & ATTRIBUTES - TEXT FORMATTING

### RELEVANT KNOWLEDGE

#### Text formatting

You can format the text content of a web page to make it attractive by using other HTML elements such as Headings, Bold, Italics, etc.

Element	Description
Heading	There are six levels of heading elements from H1 to H6. H1 has the largest text size and H6 has the smallest heading text size.
Emphasize	To emphasize a particular text in a sentence (for example a text that should be pronounced different), use this element. Text enclosed within <code>&lt;em&gt;</code> and <code>&lt;/em&gt;</code> are displayed in <i>italics</i> .
Italics	If you want the text to be displayed in alternate voice such as a text translated from a foreign language, technical words or present text in typographic italics, enclose the text within <code>&lt;i&gt;</code> and <code>&lt;/i&gt;</code> elements.
Strong	If you want to emphasize a text with stronger importance in a sentence, use the <b><i>strong</i></b> element. Text enclosed within <code>&lt;strong&gt;</code> and <code>&lt;/strong&gt;</code> are displayed in bold.
Bold	To make the text such as keywords visually bold, enclose the text within <code>&lt;b&gt;</code> and <code>&lt;/b&gt;</code> , the bold element.
Underline	To underline the text, enclose the text within <code>&lt;u&gt;</code> and <code>&lt;/u&gt;</code> , the underline element.
Mark	To highlight the text in a sentence, enclose the text within mark element : <code>&lt;mark&gt;</code> and <code>&lt;/mark&gt;</code>
Big	If you want to display a particular text in a sentence bigger than rest of the text in a sentence, enclose the text within the big element : <code>&lt;big&gt;</code> and <code>&lt;/big&gt;</code> .
Small	To display a particular text in a sentence as smaller than the rest of the text in a sentence, enclose the text within the small element: <code>&lt;small&gt;</code> and <code>&lt;/small&gt;</code> .
Subscript & Superscript	Text enclosed within <code>&lt;sub&gt;</code> and <code>&lt;/sub&gt;</code> are displayed as subscripts and Text enclosed within <code>&lt;sup&gt;</code> and <code>&lt;/sup&gt;</code> are displayed as superscripts.
Pre	Text enclosed using <code>&lt;pre&gt;</code> and <code>&lt;/pre&gt;</code> tags are displayed as it is (in the example below - using a mono-space font such as Courier).
INS & DEL	Text enclosed within <code>&lt;del&gt;</code> and <code>&lt;/del&gt;</code> are displayed as struck out while text enclosed within <code>&lt;ins&gt;</code> and <code>&lt;/ins&gt;</code> is underlined.
Break	To separate single lines or use multiple line spacing, use the break element. Breaks between lines are represented through use of <code>&lt;br&gt;</code> . No closing tag is needed.
HR	To separate sections, use a horizontal line to indicate distinctiveness. Horizontal line is indicated through use of <code>&lt;hr&gt;</code> element and requires no closing tag.

#### Sample Code - Text Elements

```

<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<p> HTML supports six levels of headings, H1 to H6 with H1 being the
largest.</p>
<h1>Main Heading</h1>
<h2>Sub heading, Level 1</h2>
<h3>Sub heading, Level 2</h3>
<h4>Sub heading, Level 3</h4>
<h5>Sub heading, Level 5</h5>
<h6>Sub heading, Level 6</h6>
<p> You are required to <em>agree</em> with our terms and conditions.</p>
<p> <i>HTML</i> is simple to learn.</p>
<p>It is important to know the common <strong>HTML Elements and
attributes.</strong></p>
<p>This an example of text displayed in <b>bold</b>.</p>
<p>This an example of text displayed <u>underlined</u>.</p>
<p>This is an example of highlighted <mark>text.</mark></p>
<p><big>Sun</big> is the largest of all planets with 696,000 KM as it's
radius.</p>
<p><small>Mercury</small> is the smallest planet in the Solar System ever
since Pluto lost its status and was reclassified as a dwarf planet.</p>
<p>this is an example of subscript for a formula, H<sub>2</sub>O</p>
<p>this is an example of superscript for a formula, (A+B)<sup>2</sup></p>
<pre>
/* Hello World program */
#include<stdio.h>
main()
{
    printf("Hello World");
}
</pre>
<p>this is the correct spelling of <del>entreprenaur</del>
<ins>entrepreneur</ins></p>
</body>
</html>

```

## Referencing Elements

Element	Description
<b>Quote</b>	To include short text such as name of an author, publisher, etc. in quotes, use the <i>quote</i> element. You can also use the <i>cite</i> attribute to indicate the source. Text included within <code>&lt;q&gt;</code> and <code>&lt;/q&gt;</code> is displayed within quotations.
<b>Blockquote</b>	To include lengthy quotes, include text within <code>&lt;blockquote&gt;</code> and <code>&lt;/blockquote&gt;</code> elements.
<b>Cite</b>	To specify a title of a work such as title of a book, movie, painting, etc. include text within <code>&lt;cite&gt;</code> and <code>&lt;/cite&gt;</code> elements.
<b>Abbreviation</b>	To include abbreviations of acronyms, include the abbreviation between

**<abbr>** and **</abbr>** elements. Text included within this element will be displayed when the cursor is moved over the text.

## Sample Code - Referencing Elements

```
<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<p>HTML Tutorial
<q>Created by Author Name</q>

<blockquote cite="http://en.wikipedia.org/wiki/Google_Analytics">

Google Analytics (GA) is a service offered by Google that generates detailed
statistics about a website's traffic and traffic sources and measures
conversions and sales.</blockquote>
<p>You can learn more about web design at <cite>
http://en.wikipedia.org/wiki/Web_design</cite>
<p><abbr title=" Hypertext Markup Language">HTML</abbr> has several elements
and attributes. </p>
</p>
</body>
</html>
```

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Modify the web page created in the earlier session using the following guidelines: <ul style="list-style-type: none"><li>• Use Heading 1 for titles and Heading 2 for sub-titles.</li><li>• Highlight important sentences using Mark Element.</li></ul>
2.	Create a webpage for the following topics: <ul style="list-style-type: none"><li>• Compile a list of ten Algebraic with formula</li><li>• Compile a list of ten Chemicals with the molecular formula.</li></ul> Use the following guidelines: <ul style="list-style-type: none"><li>• Write at least one or two sentences about each topic.</li><li>• Use Subscript and Superscript elements as required.</li></ul>
3.	Create a webpage to include the following list of acronyms: <ul style="list-style-type: none"><li>• OSI Model</li></ul>

	<ul style="list-style-type: none"> <li>• UTP Cables (with categories of cables)</li> <li>• Application Layer Protocols (with examples)</li> </ul> <p>Use the following guidelines:</p> <ul style="list-style-type: none"> <li>• Write at least one or two sentences about each topic.</li> <li>• Use Abbreviation Element to include acronyms in text content.</li> </ul>
--	---

## ASSESSMENT

Answer the following questions:

- a. Explain the purpose of the following elements with an example:
- a. Heading
  - b. Emphasize
  - c. Strong
  - d. Mark
  - e. Break
  - f. HR
  - g. Subscript
  - h. Superscript
  - i. Pre
  - j. INS
  - k. DEL
  - l. Quote
  - m. Blockquote
  - n. Abbreviation

Fill in the blanks:

1. HTML provides \_\_\_ levels of heading.
2. \_\_\_\_\_ tag should be used for displaying text with the largest heading size.
3. \_\_\_\_\_ tag is used to make a text visually bold.
4. \_\_\_\_\_ tag is used to make a text to be visually displayed in Italics.
5. \_\_\_\_\_ tag is used to make a text underlined.
6. \_\_\_\_\_ tag is used for indicating paragraphs.
7. \_\_\_\_\_ tag is used to create subscripted text.
8. \_\_\_\_\_ tag is used for aligning text to center of a document.
9. \_\_\_\_\_ tag is used for displaying abbreviations.
10. \_\_\_\_\_ tag is meant for quoting lengthy references.

## SESSION 4: HTML ELEMENTS & ATTRIBUTES - LISTS & LINKS

### RELEVANT KNOWLEDGE

You can list elements to represent ordered, unordered or definition items. This could be useful when you want to list instructions, numbered sets, etc. on a web page.

Element	Description
<b>UL</b>	To create a list with no sequence, use the Unordered List element, the tags <code>&lt;ul&gt;</code> and <code>&lt;/ul&gt;</code> represent an unordered list and <code>&lt;li&gt;</code> & <code>&lt;/li&gt;</code> represent items in the list.
<b>OL</b>	To create a list with sequence, use the Ordered List element, the tags <code>&lt;ol&gt;</code> and <code>&lt;/ol&gt;</code> represent an ordered list and <code>&lt;li&gt;</code> & <code>&lt;/li&gt;</code> represent items in the list.
<b>Definition</b>	Definition lists are used when creating lists of terminologies along with some descriptive text. The definition term tag is <code>&lt;dt&gt;</code> and the definitions are paced between the <code>&lt;dd&gt;</code> and <code>&lt;/dd&gt;</code> elements. The List of definitions are enclosed within the <code>&lt;dl&gt;</code> and <code>&lt;/dl&gt;</code> elements.

### Sample Code - Unordered Lists

```
<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<p>Junk food is a slang term for food that is of little nutritional value
and often high in fat, sugar, salt, and calories; eating Junk food is
considered unhealthy. Health food is food considered to be beneficial to
health in ways that go beyond a normal healthy diet required for human
nutrition. Following is a generalized list of Junk & Health food.
<ul>Junk Food
<li>Potato Chips
<li>Pizza
<li>Soda
</ul>
<ul>Health Food
<li>Milk
<li>Fruits
<li>Raw or Cooked Vegetables
</ul></p></body></html>
```

### Sample Code - Ordered Lists

```

<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<p>To check if the HTML code works in a variety of browsers, do the
following:
<ol>Instructions to use Mozilla Firefox
<li>Go to http://www.mozilla.org/en-US/firefox
<li>Select Mozilla Firefox Free download for desktop
<li>Install once complete
<li>Open the web page in Mozilla Firefox
</ol>
</p>
</body>
</html>

```

## Sample Code - Definition Lists

```

<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<p>Following are examples of definition lists. </p>
<dl>
<dt>Pizza</dt>
<dd>Pizza is an oven-baked, flat, round bread typically topped with a tomato
sauce, cheese and various toppings. Pizza was originally invented in Naples,
Italy, and the dish has since become popular in many parts of the world. An
establishment that makes and sells pizzas is called a "pizzeria".</dd><br>
<dt>Potato Chips</dt>
<dd> A potato chip is a thin slice of potato that is deep fried or baked
until crunchy. Potato chips are commonly served as an appetizer, side dish,
or snack. The basic chips are cooked and salted; additional varieties are
manufactured using various flavorings and ingredients including seasonings,
herbs, spices, cheeses, and artificial additives.</dd><br>
<dt>Fruit</dt>
<dd> A fruit is a part of a flowering plant that derives from specific
tissues of the flower, one or more ovaries, and in some cases accessory
tissues. Many hundreds of fruits, including fleshy fruits like apple, peach,
pear, kiwifruit, watermelon and mango are commercially valuable as human
food, eaten fresh and as jams, marmalade and other preserves. Fruits are also
used in manufactured foods like:
<ul><li>Cookies<li>Muffins<li>Yogurt<li>Ice cream<li>Cakes, and many more.
</ul></dd>
</dl> </body> </html>

```

## Links

Links are used for interlinking to web pages of other websites, different sections of a web page, display email addresses, links to websites etc. Links that are used within a website are referred to as Internal Links. These are used when you want to navigate between pages or link to text, image, etc. within a website. Links that are outside a website are referred to as external links. When using hyperlinks, you need to provide a path that points to the resource. You can use:

- An absolute path contains complete path details, for example:
  - `http://www.example.com/folder1/subfolder1/filename1.htm`
- A relative path refers to a folder or location that is shorter, for example:
  - `index.htm`
  - `../tutorials` (previous folder)
  - `/foldername`

Absolute paths are used when creating external hyperlinks as you need to provide a complete path. Relative path is more common when you link resources within a website.

Links are created using the `<a>` anchor element and `href` attribute along with link to the resource. Text enclosed within `<a href>` and `</a>` are displayed as clickable hyperlinks.

### Sample Code - Links

```
<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<p>This is an example of an hyperlink that points to another <a href="page2.htm">page</a> within a website.</p>
<p>This is an example of an hyperlink that points to <a href="http://www.google.com">Google</a>, a popular search engine.</p>
<p>this is an example of an hyperlink that opens the default email client when clicked to send a mail to <a href="mailto:webmaster@example.com">webmaster</a> of example.com</p>
<a href="http://www.google.com" target="_blank">This is an example of link that opens in a new window or tab.</a>
</html>
```

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a unordered list the types of soil found in India. Save the webpage for later sessions.
2.	Create a ordered list for New Indian states (e.g. Uttranchal, Chattisgarh etc). Save the webpage for later sessions.
3.	Create a web page with links to the following: a. <a href="http://www.htmlhelp.com">www.htmlhelp.com</a> b. <a href="http://www.w3schools.com">www.w3schools.com</a> c. <a href="http://developer.mozilla.org/en-US/learn/html">developer.mozilla.org/en-US/learn/html</a> Save the webpage for later sessions.

## ASSESSMENT

Answer the following questions:

1. Explain the procedure to create an unordered list.
2. Explain the procedure to create an ordered list.
3. Explain the procedure to create a definition list.
4. Explain the procedure to create a hyperlink to website that needs to open in a new window.
5. Explain the procedure to include a clickable email address in a web page.
6. Explain Relative and Absolute Paths with an example.

Fill in the blanks:

1. \_\_\_\_\_ tag is used for indicating a unordered list.
2. \_\_\_\_\_ tag is used for indicating a ordered list.
3. \_\_\_\_\_ tag is used for indicating a definition list.
4. \_\_\_\_\_ element & \_\_\_\_\_ attribute is used for creating hyperlinks.

## SESSION 5: HTML ELEMENTS & ATTRIBUTES - TABLES, IMAGES & MULTIMEDIA

### RELEVANT KNOWLEDGE

#### Tables

You can create a table using the Table Element. Tables are made of rows and columns similar to that of a spreadsheet. Rows are contained within table row element **<tr>** and data is stored within row element **<td>**.

```
<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<table>
<tr><td>Name</td><td>Age</td><td>City</td></tr>
<tr><td>Anand</td><td>17</td><td>Delhi</td></tr>
<tr><td>Rohit</td><td>23</td><td>Bangalore</td></tr>
</table>
</body>
</html>
```

#### Images

You can include images in your web page by using the **<img>** image element. Use the **SRC** attribute to specify the location of the image and **width** and **height** attributes to specify the dimensions of the image. Also it is a recommended practice to use the **ALT** attribute for displaying text if the image is not downloaded.

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

</body></html>
```

#### Working with Audio, Video & Flash

Element	Description
<b>Audio</b>	To include an audio clip, enclose the audio within the <code>&lt;audio&gt;</code> and <code>&lt;/audio&gt;</code> elements and point to the clip using <b>SRC</b> attribute. To automatically play the audio, use the <b>autoplay</b> attribute. To display controls, use the <b>controls</b> attribute.
<b>Video</b>	To include a video clip, enclose the audio within the <code>&lt;video&gt;</code> and <code>&lt;/video&gt;</code> elements and point to the clip using <b>SRC</b> attribute. Also use <b>width</b> and <b>height</b> attributes to specify the dimensions of a video clip.
<b>Flash</b>	To embed a flash animation, use the <code>&lt;embed&gt;</code> and <code>&lt;/embed&gt;</code> elements and point to the animation using <b>SRC</b> attribute.

## Sample Code - Multimedia

```

<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<audio src="pump_im.mp3" autoplay controls>
Your browser does not support the audio element.
</audio>
</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<video src="sample_mpeg4.mp4" height="320" width="200" controls>
Your browser does not support the Video element.
</video>
</body>
</html>

```

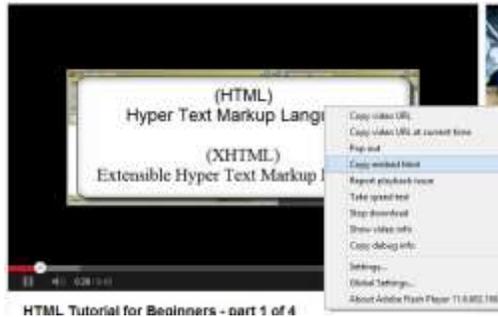
```

<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<embed src="sample.swf" height="200" width="320">
</body>
</html>

```

You can also include videos from other sources such as youtube.com, etc. To include a video from youtube.com, do the following:

1. Go to Youtube.com
2. Select a video clip
3. Right-click on the video and select Copy embed html



Write the code within <body> and </body> section (sample below).

```

.<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<iframe width="640" height="360"
src="http://www.youtube.com/embed/v4oN4DuR7YU?feature=player_detailpage"
frameborder="0" allowfullscreen></iframe></body>
</html>

```

Now save and open this web page using a web browser.

Note: Refer to Youtube.com FAQ' for detailed instructions.

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a audio dictionary using the following guidelines: <ol style="list-style-type: none"> <li>a. Choose at least five words from dictionary.com or other source such as Wikipedia. Download relevant audio files (pronunciation) and include next to the word.</li> </ol>
2.	Create web based tutorial for the following topics: <ol style="list-style-type: none"> <li>a. HTML</li> <li>b. HTML Templates</li> <li>c. Web Hosting</li> </ol> Use the following guidelines: <ol style="list-style-type: none"> <li>a. Link to tutorials relevant to the topic from youtube.com.</li> <li>b. Use videos only that are marked as public domain.</li> </ol>

## ASSESSMENT

**Answer the following questions:**

1. Explain the procedure to insert an audio clip in a HTML document.
2. Explain the procedure to insert a video clip in a HTML document.
3. Explain the procedure to insert a flash animation in a HTML document.
4. Explain any three image file formats for use in web with examples.

**Fill in the blanks:**

1. \_\_\_\_ tag is used for inserting audio clips in a HTML 5 document.
2. \_\_\_\_ tag is used for inserting video clips in a HTML 5 document.
3. Acronym for GIF \_\_\_\_\_.
4. Acronym for JPEG \_\_\_\_\_.
5. Acronym for PNG \_\_\_\_\_.
6. File extension for flash animations \_\_\_\_\_.

## SESSION 7: HTML ELEMENTS & ATTRIBUTES - FORMS & FRAMES

### RELEVANT KNOWLEDGE

#### FORMS

Forms contain input elements such as text boxes, dropdown menus, check boxes, radio buttons, etc. to collect information from visitors and sent to a web server for further processing such as displaying search results or send to a database for storage.

Forms are created using `<form>` element along with action and method attributes. **Action** attribute is used for specifying the URL to where the data needs to be submitted. Method attribute is used for specifying the HTTP method, **get** or **post**. POST attribute is used for storing or updating a database or for sending emails and GET attribute is used for retrieving data.

Form element uses text boxes to collect text information and radio buttons, check boxes and dropdown menus for collecting information through pre-defined options.

Control	Description
Textbox	Single-line text input box is typically used for collecting shorter text such as usernames or passwords. Multiple-line text input boxes are used for collecting lengthier text such as narrative feedback, etc.
Radio Buttons	Used for collecting single value from multiple values. You should use <b>type = radio</b> attribute for using radio buttons in a web page.
Checkbox	Used for collecting multiple values. You should use <b>type = checkbox</b> attribute for using checkboxes in a web page.
Dropdown	Used for collecting single or multiple values from a dropdown menu. You should use the <b>option</b> attribute along with values to present multiple options.
Submit	Submit button allows visitors to submit the value.
Reset	Reset button clears all the selections made.

#### Sample Code - Forms

```
<!DOCTYPE html>

<html>
<head>
<title>Learn HTML</title>
</head>
<body>
<p>Subscribe to your weekly Newsletter</p>
<form action="http://www.example.com/signup.asp" method="post">
Name <input type="name" name="Name" size="30">
Password <input type="password" name="Name" size="30">
<textarea rows="3" name="Comments" cols="30"></textarea>
```

```

<p>I agree to the Terms and Conditions.
<input type="radio" value="Agree" name="R1"> Yes
<input type="radio" value="Diagree" checked name="R1"> No</p>
<p>10th <input type="checkbox" name="Class_10" value="ON"> 12th
<input type="checkbox" name="Class_12" value="ON"> Bachelors Degree
<input type="checkbox" name="Bachelors_Degree" value="ON"> Masters Degree
<input type="checkbox" name="Masters_Degree" value="ON"></p>
<p>This is an example of Dropdown Menu (Single Selection)</p>
<p>School Education Board<select size="1" name="School">
<option>State Board</option>
<option>Matriculation</option>
<option>CBSE</option>
</select> </p>
<p>
<input type="submit" value="Submit" name="Submit">
<input type="reset" value="Reset" name="Reset"></p>
</form>
</body>
</html>

```

## Frames

Frames are used to display one or more pages within a single web page; collections of different frames are referred to as a frameset. Frames are used when you want to display different pages as a single page in a web browser.

```

<html>
<frameset cols="25%,50%,25%">
<frame src="sample.htm">
<frame src="sample.htm">
<frame src="sample.htm">
<noframes>
Text to be displayed in browsers that do not support frames
</noframes>
</frameset>
</html>

```

## iframe

iframe or inline frames is an element used for embedding another document or an URL with a web page.

```

<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title></head>
<body>
<p>You can use iframe to include a web page.</p>
<iframe width="600" height="300" src="lesson002.htm"></iframe>
<hr>

```

```

<p>You can use iframe to view another website live.</p>
<iframe width="600" height="300" src="http://www.mozilla.org"></iframe>
<hr>
<p>You can restrict scrolling in iframe.</p>
<iframe width="600" height="300" src="http://www.wikipedia.org"
scrolling=no></iframe>
</body>
</html>

```

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities																											
1.	<p>Create a form using the following guidelines:</p> <table border="1"> <thead> <tr> <th>Field Type</th> <th>Field</th> <th>No. of Characters</th> </tr> </thead> <tbody> <tr> <td>Text box</td> <td>First Name</td> <td>30</td> </tr> <tr> <td>Text box</td> <td>Last Name</td> <td>30</td> </tr> <tr> <td>Text box</td> <td>LoginName</td> <td>30</td> </tr> <tr> <td>Text box</td> <td>Password</td> <td>15</td> </tr> <tr> <td>Text box</td> <td>ConfirmPassword</td> <td>15</td> </tr> <tr> <td>Text box</td> <td>Email</td> <td>30</td> </tr> <tr> <td>Text box</td> <td>Homepage</td> <td>50</td> </tr> <tr> <td>Dropdown (Single Selection)</td> <td>Qualification (10<sup>th</sup>, 12<sup>th</sup>, Diploma, Bachelor's Degree, Master's Degree)</td> <td></td> </tr> </tbody> </table> <p>a. Include Submit and reset buttons b. Configure the form to send a mail to your personal email address when submitted.</p>	Field Type	Field	No. of Characters	Text box	First Name	30	Text box	Last Name	30	Text box	LoginName	30	Text box	Password	15	Text box	ConfirmPassword	15	Text box	Email	30	Text box	Homepage	50	Dropdown (Single Selection)	Qualification (10 <sup>th</sup> , 12 <sup>th</sup> , Diploma, Bachelor's Degree, Master's Degree)	
Field Type	Field	No. of Characters																										
Text box	First Name	30																										
Text box	Last Name	30																										
Text box	LoginName	30																										
Text box	Password	15																										
Text box	ConfirmPassword	15																										
Text box	Email	30																										
Text box	Homepage	50																										
Dropdown (Single Selection)	Qualification (10 <sup>th</sup> , 12 <sup>th</sup> , Diploma, Bachelor's Degree, Master's Degree)																											
2.	Create a feedback form using a free online form service provider and integrate it with your blog.																											
3.	<p>Create a web page that includes the following websites using iframe:</p> <ul style="list-style-type: none"> <li>• dictionary.cambridge.org</li> <li>• www.merriam-webster.com</li> <li>• www.thefreedictionary.com</li> </ul>																											

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of forms with an example.
2. Explain the purpose of Frames with an example.
3. Explain iframe with an example.
4. Explain the procedure to include a special character with an example.

**Fill in the blanks:**

1. \_\_\_\_\_ element is used for collecting information from a visitor in a web page.
2. \_\_\_\_\_ element is used for collecting names and passwords.
3. \_\_\_\_\_ , \_\_\_\_\_ & \_\_\_\_\_ elements can be used for collecting information through pre-defined options.
4. \_\_\_\_\_ attribute or method is used for storing information.
5. \_\_\_\_\_ attribute or method is used for retrieving information.
6. \_\_\_\_\_ element is used for including frames.
7. Acronym for iframe \_\_\_\_\_

## SESSION 9: HTML ELEMENTS & ATTRIBUTES - METATAGS, DOCTYPES & VALIDATION

### RELEVANT KNOWLEDGE

Meta elements are used to provide structured metadata about a webpage. Metadata, also referred to as data about data is used for providing internal information i.e information that is understood by web browsers or web servers. Meta tags are used by web browsers, search engines or web services.

- Meta tags must be included within the `<head></head>` sections.
- Use name attribute and value author to include an Author's name.
- Use name attribute and value description to include descriptions.
- Use name attribute and value keywords to include keywords. Keywords are used by search engine to index later which is used by search engines to provide results to a visitor.
- Use name attribute and value copyright to include copyright details.

### Sample Code - Meta Tags

```
<!DOCTYPE html>
<html>
<head>
<title>Learn HTML</title><meta name="author" content="First Name, Last
Name">
<meta name="description" content="Learn about Meta Elements and Attributes">
<meta name="keywords" content="Meta Element, Meta Attributes, HTML Author
Element">
<meta name="copyright" content="COMPANY or AUTHOR NAME">
</head>
<body>
<p>Use View Source to view the code as code placed in header section will
not be displayed in the body section.</p>
</body>
</html>
```

### DOCTYPES

DOCTYPES are declarations used for providing a formal definition of the HTML version used in a web page. Browsers sniff or use this DOCTYPE declaration to understand the HTML tags (elements & attributes) used in a web page. DTD or Document Type Definition refers to set of markup declarations that define a standard document type.

You have learnt a variety of HTML elements and attributes in this module. Not all elements, attributes or combination of both is supported by all browsers; to ensure compatibility, it is considered a best practice to use compatible HTML elements and

attributes and also indicate the version of HTML standard used by a web page through use of a DOCTYPE.

DOCTYPES must be indicated in the first line of a document as it helps the browser understand the format used in a web page. Following are examples of DOCTYPES:

```
<!DOCTYPE html>
<html>
<head>
<title>DOCTYPE for HTML5</title>
</head>
<body>
<p>Documents marked with !DOCTYPE html are considered to be HTML 5
complaint</p>
</body>
</html>
```

## HTML Validation

Though there are a variety of browsers available, it is important that you test the code on popular browsers before making it available on the Internet. If it is not checked, you cannot be sure of how it will be displayed to a user using a different browser than yours. Also note that all computers will not have the recent versions of web browsers; some users may still be using older versions of the web browsers. It is the responsibility of a developer to check, verify and confirm if a website can be published after thorough testing.

There are a variety of tools available for checking the HTML validity. Some of them include:

- Online HTML Markup Validation Service such as:
  - <http://validator.w3.org>
  - <http://browsershots.org>
  - <http://browserlab.adobe.com>
- Web Browser extensions:
  - HTML Validator for Firefox (<http://users.skynet.be/mgueury/mozilla/>)
  - HTML Validator for Google Chrome (<http://robertnyman.com/html-validator/>)
- Use offline software program such as CSE HTML Validator Lite (<http://www.freehtmlvalidator.com/>)

## Additional Reading

Though you have learned about basics of HTML, use the following resources for enhancing your knowledge on HTML:

1. <http://www.w3schools.com/html/default.asp>
2. <http://www.htmlhelp.com/>
3. <http://en.wikipedia.org/wiki/HTML>

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Include the following details using meta tags to the web pages created in previous sessions: <ul style="list-style-type: none"><li>• use author and fill your name</li><li>• use keywords and fill at least three to five keywords based on the topic of your web page used for this exercise</li><li>• include instructions for the web page not to be cached by a browser</li></ul>
2.	Navigate to <a href="http://validator.nu">http://validator.nu</a> and validate all the web pages created in previous sessions.

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of meta elements.
2. Explain any three meta elements with an example.
3. Explain the purpose of DOCTYPEs with examples.
4. Explain the purpose of HTML validation with an example.

Fill in the blanks:

1. \_\_\_\_\_ meta name is used for providing author's name.
2. \_\_\_\_\_ meta name is used for providing detailed information about a web page.
3. \_\_\_\_\_ meta name is used for refreshing a page automatically every few seconds or minutes.
4. \_\_\_\_\_ meta name is used to instruct a web browser never to cache.
5. DOCTYPE to be used for pages utilizing HTML 5 \_\_\_\_\_ .

## SESSION 10: INTRODUCTION TO CSS

### RELEVANT KNOWLEDGE

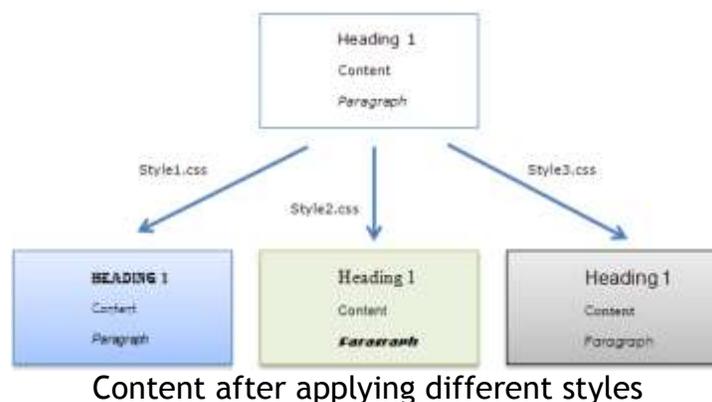
CSS or Cascading Style Sheets is a language used for describing the presentation format of a web page. While HTML focuses on *structure*, CSS focuses on the *presentation*.

Imagine that you are constructing a house. All the rooms, windows, kitchen, hall etc have been built to specification. But still, it is only after the coats of paint are given, does the finish and appeal to the home come. CSS is like the beautiful coat of paint that you apply to the HTML structure to make it aesthetically appealing.

Advantages of CSS include:

- Separate content from presentation that is easier for developers to work with.
- Use different style sheets to target different web browsers or devices.
- Save bandwidth by writing several lines of presentation code in a separate style sheet and link to it multiple pages.
- Provide customized web pages to users based on their preferences.

Styles can be applied to a web page just with a click of a mouse. For example if you have a web page with content using different HTML elements, you can apply different styles to all the HTML elements at once. You can also create multiple style sheets that can be applied to a web page on a rotation basis.



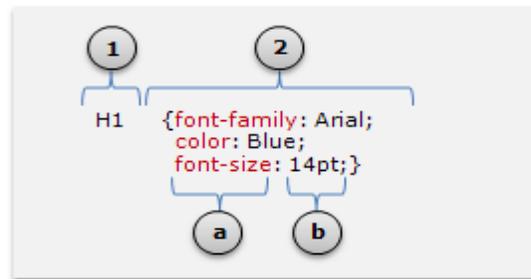
### CSS Versions

Like HTML, CSS also uses simple syntax and is easier to learn. CSS has also evolved over time and popular versions include CSS 2.1 and CSS3.

## CSS Parts

CSS has two main parts, Selector and Declaration that associates rules for HTML elements. (Refer figure below)

1. Selector specifies the element that needs to be applied (example: H1)
2. Declaration specifies the parameters for the selector. Declaration has two parts, Properties and Values
  - a. Properties specify the aspects of an element such as font, color, size, border, etc.
  - b. Values specify the value of an aspect. (example: 14pt, blue, etc.)



## Writing your first CSS Code

You can use a simple text editor such as notepad to create CSS code. You can integrate the CSS code in a HTML file by one of the three following methods:

### 1. Using INLINE Code

```
<!DOCTYPE html>
<html>
<head>
<title>Inline Code</title>
</head>
<body>
<p>Use View Source to view the code as code placed in header section
will not be displayed in the body section.</p><hr>
<h1 style="font-family: Arial; font-size: 14pt; color: #0000FF">This
is heading 1 formatted with Inline Code</h1>
</body>
</html>
```

Notice the CSS code is placed within sections of the web page.

## 2. Embedding CSS Code

```
<!DOCTYPE html>
<html>
<head>
<title>Embedded Code</title>
<style>
h1    { font-family: Arial; font-size: 14pt; color: #0000FF }
</style>
</head>
<body>
<p>Use View Source to view the code as code placed in header section
will not be displayed in the body section.</p><hr>
<h1>This is heading 1 formatted with Inline Code</h1>
</body>
</html>
```

Notice the CSS code is placed with `<style></style>` tags. Usually it is considered as a best practice to place the code within the header section of a web page i.e. the `<head></head>` section. This is to enable browsers to speed up the process of applying styles to rest of the content. If you place the CSS code within the body, the browser will reload the page to apply the code resulting in a slower display of the web page.

In this case, you need to add the CSS Code to all the web pages in a website.

## 3. Referring to an external style sheet

```
<!DOCTYPE html>
<html>
<head>
<title>External CSS</title>
<link rel="stylesheet" type="text/css" href="CSSExercise001.css">
</head>
<body>
<p>Use View Source to view the code as code placed in header section
will not be displayed in the body section.</p><hr>
<h1>This is heading 1 formatted with CSS Code from an external
CSS</h1>
</body>
</html>
```

```
h1    { font-family: Arial; font-size: 14pt; color: #0000FF }
```

In this case there are two documents, a .htm and a .css file. CSS code is placed within the .CSS file is linked to a HTML page with a single line of code. This can help in saving time by excluding complete CSS code in all web pages of a

website. You can also alter the code in the CSS file that will automatically get executed when the HTML page is loaded next time. Style sheets have the file extension .CSS.

Now change the CSS code,

```
h1 { font-family: Arial; font-size: 14pt; color: red }
```

Refresh the page that include references this style sheet and notice the results displayed.

### Additional Reading

Though you have learned about basics of CSS, use the following resources for enhancing your knowledge on CSS:

1. <http://www.w3schools.com/css/default.asp>
2. <http://www.caniuse.com>
3. <http://www.css3.info>
4. [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)

### EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create an external style sheet for the following elements. a. Heading 1 (H1): Font Verdana, Color Violet, Size 10pt b. Heading 2 (H2): Font Verdana, Color Dark Green, Size 8pt c. Paragraph (P): Font Verdana, Color Black, Size 8pt
2.	Create a web page using the following HTML elements (this will be used over the next few sessions in CSS Module): a. H1, H2 & H3 b. Paragraph c. Table (with some data) d. Use Horizontal Breaks e. Hyperlinks to at least three websites f. Registration Form (Name, Mobile, Email Address, etc.)

## ASSESSMENT

Answer the following questions:

1. Explain CSS with an example.
2. Explain different procedures to implement CSS.

Fill in the blanks:

3. \_\_\_\_\_ is a language used for describing the presentation format of a web page.
4. Acronym for CSS \_\_\_\_\_.
5. Recent version of CSS is \_\_\_\_\_.
6. Three methods of implementing CSS are \_\_\_\_\_ , \_\_\_\_\_ & \_\_\_\_\_.
7. Extension of style sheets \_\_\_\_\_.
8. \_\_\_\_\_ and \_\_\_\_\_ associate rules for HTML elements.
9. CSS Code must be place within \_\_\_\_\_ & \_\_\_\_\_ tags.

## ASSIGNMENT

1. Create a website with minimum 5 web pages for the following topics:
  - a. Windows 8 Mobile APP development.
  - b. Android Mobile APP development.
  - c. iOS Mobile APP development.
  - d. Free APP makers (Android, iOS & Windows).
  - e. Life Skill Development.
  - f. Learning at home using elearning.
2. Use the following guidelines for completing this exercise:
  - a. Use Heading 1 for title .
  - b. Use heading 2 for sub-titles.
  - c. Use tables whenever required.
  - d. Use Meta tags.
    - i. Provide brief overview of each page content (Title Element).
    - ii. Provide description of each page in less than 50 characters (Description Element).

- e. Provide the links and resources that you used for developing this website as comments (Comment Element).
- f. Use adequate breaks between paragraphs
- g. Include at least three to five abbreviations
- h. List the objectives at the starting as an ordered or unordered list.
- i. Store all images in \images folder.
- j. Use absolute path for all external links
- k. Use relative path for all internal links.
- l. Use hover effects for hyperlinks
- m. Include relevant images as thumbnails. Configure images to open in new window.
- n. Include at least three videos from youtube.com relevant to the selected topic.
- o. Create a feedback form. Use websites such as emailmeform.com and integrate the form in a separate web page (for example, feedback.htm).
- p. Include navigation menu for including links to other web pages within this website.
- q. Use at least three tables. Tables must have only dotted borders.
- r. Use HTML 5 as DOCTYPE and must pass validation.
- s. Use CSS rounded corners for tables.
- t. Use CSS for ALL elements.
- u. Store all style sheets in \css folder.
- v. Use CSS 2.1 as CSS Schema and must pass validation.
- w. Include link to homepage on all web pages placed at the bottom.
- x. Link to W3C HTML validator must be available on all web pages.

## SESSION 11: OVERVIEW OF XML & XHTML

### RELEVANT KNOWLEDGE

As you learn more about web designing or development, you need to learn some other technologies that are used along with HTML & CSS such as XML & XHTML.

#### XML

XML or *Extensible Markup Language* is a markup language that defines a set of rules for encoding formats. Unlike HTML, apart from being used in web pages, XML is also used for variety of purposes such as database storage or retrieval and data exchange. XML is used for separating data and document structure.

For example, when you design web sites, you may display product catalog or shopping cart to visitors. The details of the products can be stored locally on the visitor's computer in an XML file instead of storing on the server thereby reducing the number of trips in between.

#### Use of XML

Imagine if you want to send hundred records of matching products to a visitor based on a specified keyword for example books. Here, each record may contain information on book title, description, image of the cover, author, publisher, number of pages, etc. and displaying these records on a single webpage may make it difficult for the visitor to scroll up and down (a single lengthy page!). On the other hand if you want to display only ten records a time, you need to send the request multiple times to the server thus resulting in increased traffic and bandwidth usage. In such a case, you can store the values of all records in a single file, write some code using HTML or JavaScript and display ten records at a time in the visitor's browser!

XML is also useful when you use incompatible database management systems. For example, if a company uses proprietary database software which is different from the client's software, they may encounter issues while sending the data to their clients. In such cases, XML can be used to send the data that can be exported or imported at both ends.

Another advantage is that you can use any other application other than HTML to display the data. For example you can write your own application for addressing people with visual disabilities and read out the data stored in XML.

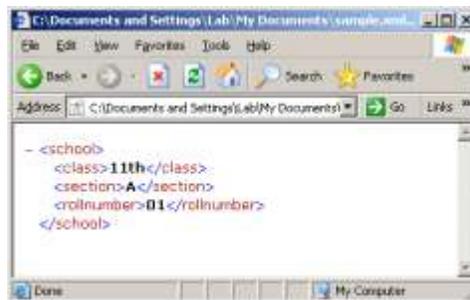
## XML Code

XML is written as string of characters. XML document contains markup and data. XML has a structure similar to HTML but it does not have any pre-defined elements like HTML. You need to create your own tags. Like HTML or CSS, you can use a simple text editor such as notepad to create XML code. XML code is case sensitive, requires closing tags and should be properly nested (unlike HTML).

1. Type the following code in a text editor such as notepad.

```
<school>
<class>11th</class>
<section>A</section>
<rollnumber>01</rollnumber>
</school>
```

2. XML files have the extension .XML. Now save this file as *intro.xml*.
3. Open this file (intro.xml) in a web browser. Notice the results displayed (figure below).



Since you need to write your own code, given below are some of the rules you should follow:

- Names cannot start with a number or punctuation character.
- Names can contain letters, numbers, and other characters.
- Names cannot start with the letters xml.
- Names cannot contain spaces (white space is allowed in data).
- Elements are extensible (you can append tags in between).
- Entity reference such should be used instead of special characters such as >, <, &, ' & '.
- Comments can be placed between <!--COMMENT -->.
- XML documents must have a root element.
- XML documents are case sensitive.
- XML elements require proper opening and closing tags.
- Attributes must be used within single or double quotes.
- Elements can use multiple values, attributes cannot.

- Elements are expandable, attributes are not.

## XML Today

Many languages are created based on XML:

- WAP and WML as markup languages for handheld devices.
- RSS languages for news feeds.
- SMIL for describing multimedia for the web.

Though you have gone through a high-level overview of XML, it is recommended that you use the following websites to learn more about XML and real-world examples:

- <http://en.wikipedia.org/wiki/XML>
- <http://www.w3schools.com/xml/>

## XHTML

Extensible HyperText Markup Language or XHTML is an extension of HTML. XHTML follows strict rules similar to that of XML and is used for creating well-formed HTML documents. All elements need to have an opening and a closing tag; some elements that do not require a closing tag in HTML require a self-closing syntax when used in XHTML. For example, to include break `<br>` is used HTML and `<br/>` is used in XHTML.

You can configure your HTML editor to add necessary code automatically. However, you need to specify the DOCTYPE to be used by the editor. For example to configure MEW to use XHTML, do the following:

1. Open *Expression Web*.
2. Select *Tools > Page Editor Options.....*
3. Select *Authoring tab*.
4. Select the version of XHTML from *Document Type Declaration:* dropdown menu. For example, XHTML 1.1.
5. Click *OK*.
6. Select *File > New > HTML*.
7. Select *View > Page > Code*.

Notice the XHTML version of XHTML used along with XML namespace is automatically included in the web page.

Note: If you have already used a different version, then you need to add the code manually.

Learn more about XHTML at:

- <http://en.wikipedia.org/wiki/XHTML>
- [http://www.w3schools.com/html/html\\_xhtml.asp](http://www.w3schools.com/html/html_xhtml.asp)

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	<p>Create a XML file for defining the following custom elements:</p> <ul style="list-style-type: none"> <li>a. CD Catalog               <ul style="list-style-type: none"> <li>i. CD Title</li> <li>ii. Name of the artist</li> <li>iii. Title of the song</li> </ul> </li> <li>b. Library               <ul style="list-style-type: none"> <li>i. Category of Book (Humor, Mystery, Murder, Classics etc)</li> <li>ii. Title of the book</li> <li>iii. author</li> </ul> </li> <li>c. Hotel Menu               <ul style="list-style-type: none"> <li>i. Category of Cuisine (North Indian, South Indian etc)</li> <li>ii. Item Category (Breakfast, Lunch, Tiffin,Snacks etc)</li> <li>iii. Item Name</li> </ul> </li> </ul> <p>Save the file and view in web browser.</p>

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of XML with an example.
2. Explain the purpose of XHTML with an example.
3. List at least five differences between HTML and XHTML.

Fill in the blanks:

1. Acronym for XML \_\_\_\_\_.
2. Acronym for XHTML \_\_\_\_\_.

### RELEVANT KNOWLEDGE

#### Web Site Design Scripting Tools

Web site design tools refer to tools used for web designing and web development that help in creating interactive and intuitive websites. These tools save time as they may contain ready to use, pre-defined codes. Some of them help you edit, debug or manipulate graphics. Some of the popular tools include:

- Aptana Studio
- Notepad++
- Firebug
- jEdit
- CSSED

#### Scripting Language

Scripting language refers to code written manually and executed either at client side (handled by the local browser at the client end) or server side (handled by the web server). Scripts help in extending the functionality that is not possible using standard HTML or CSS Code. For example look at a search engine like Google or train reservation system like IRCTC; these are designed to cater to thousands of visitors based on their request - like displaying search results or seat availability for a particular train. Though there are a variety of programming languages used, scripting is broadly classified into client-side and server-side scripting.

- **Client-Side scripting** refers to scripts or programs processed by the web browser at the client's end. These web browsers interpret the script and execute them on the client's computer. Hence these are referred to as Client-side scripts. Client-side scripts are usually embedded within the HTML code or in a separate file referenced in HTML and downloaded along with the HTML files. JavaScript and VBScript are some of the popular client-side scripting languages.
- **Server-Side scripting** refers to scripts or programs processed by a web server. The script is executed on the server before the results are sent to the visitor. This is useful when you need to retrieve data from a database server and send only the results to be displayed on a client computer. It enables the website owner to minimize user access to source code (which may be proprietary and also valuable). Most popular server-side scripting languages include PHP, ASP, Java via Java

Server Pages, Server Side Java Script, Server Side VB Script, Perl CGI, Ruby,Python, etc.

## Overview of JavaScript

JavaScript or JS is a popular programming language used widely on the Internet for creating interactive web pages and special effects. JavaScript provides many exciting features that cannot be provided through HTML. It should be noted that JavaScript as well as VBScript can be used as both client-side as well as server-side scripting languages.

Like HTML, JavaScript is understood and executed by web browsers such as Mozilla Firefox, Internet Explorer, Google Chrome, etc. and is therefore referred to as Client-side scripting.

JavaScript is very user friendly and is used for purposes such as:

- Providing popup text to assist visitors in filling forms.
- Redirecting to different web pages or websites.
- Displaying slideshows, attractive animations and cool effects.
- Providing personalized experience to repeating visitors (by remembering the items in their shopping cart, search keywords, Jobs searched, etc.)

The image shows a web form titled "User informations" with the following fields and error messages:

- Desired username :** Input field with error messages: "\* This field is required" and "\* No special characters allowed".
- First name :** Input field with error messages: "\* This field is required" and "\* Letters only".
- Last name :** Input field with error messages: "\* This field is required" and "\* Letters only".
- radio :** Three radio buttons with error message: "\* Please select an option".
- Age :** Input field with error messages: "\* This field is required" and "\* Numbers only".
- Telephone :** Input field with error messages: "\* This field is required" and "\* Invalid phone number".

Form with text displayed as popup message

## History

JavaScript was first created in May 1995 by Netscape, now called Mozilla. It was first called Mocha. In December 1995, the name JavaScript was adopted. JavaScript was made into a standard by an organization called ECMA. Though ECMAScript is the name of the official standard, JavaScript is the popular name being used.

## EXERCISE

Perform the following activities till you understand where JavaScript can be used

S.No.	Activities
1.	<p>Assume that you are asked to help a travel operator to design an interactive form for his website.</p> <p>He has eight buses (mix of AC, Non-AC, Sleeper, Semi Sleeper) scheduled at different times plying between multiple locations.</p> <p>The user should be able to key in his details, his destination, no. of seats required, select the type of bus and date and time of departure, after checking the availability of different types of buses, their timing and the amount that to be paid.</p> <p>Analyze and list the fields needed for creating the form to match the business requirement.</p> <p>Specify the fields that the user has to key in data and those where the computer has to display data.</p>
2.	<p>Imagine that you have to create a form to accept online deposits for the <b>Bank of India</b>. Deposits may be of various types - Fixed Deposit, Recurring Deposit, Savings Bank Deposit etc. Account numbers should be valid and also amount available in a account should be displayed when required. Also, if the amount of deposit is greater than 50000, it should ask for a PAN number. On submitting the form, it needs to display the total amount available for the depositor which is inclusive of the amount entered.</p> <p>Create a form on a chart and Indicate those areas where data is to be validated and errors displayed.</p>
3	<p>Assume that you are asked to create a form for a medical shop. Design a form on chart which will take in the Patients name, address and phone no. details, the doctor's name, the medicines he buys, their rate, qty and total. Maybe there is a discount that may be given by the shop.</p> <p>Specify which fields need to be accessed using code and the validations that need to be done on the fields.</p>

## ASSESSMENT

Answer the following questions:

1. Explain why JavaScript is said to be very 'User-Friendly'?
2. Why is JavaScript called a Client Side Scripting program?

Fill in the blanks:

1. \_\_\_\_\_ is a program widely used for creating interactive web pages.
2. JavaScript was created by \_\_\_\_\_ in the year \_\_\_\_\_

## SESSION 13: WRITING JAVASCRIPT PROGRAM

### RELEVANT KNOWLEDGE

Consider JavaScript as an add-on to HTML. While HTML is used for creating objects and CSS for presentation, JavaScript is used for manipulating objects to provide a particular function. For example you may have designed a feedback form including fields for First Name, Last Name, Age, Email Address, etc. However if you want to display a popup text to guide the visitor to fill the details properly, you can use JavaScript.

JavaScript is easy and fun to learn. You can use a text editor such as notepad to create scripts and then run them using a web browser. JavaScript codes are straight forward, simple and interpreted (understood and executed) by the browser. This means you can write the code and test it immediately.

JavaScript runs within the browser when the code is added to the existing HTML code. You need to add **special tags** to indicate the JavaScript code present within the HTML document. To indicate the presence of JavaScript code, you need to place the code within `<script>` `</script>` HTML tags.

Let us start with a simple JavaScript program to display a message box with message “Hello World”. To do this, open a blank page on a notepad and type the code

#### Example Code: First JavaScript program

```
<html>
<head>
<title>First JavaScript Program</title>
</head>
<body>
<script>           //This line tells the browser that JavaScript
follows
  alert("Hello World"); // This alerts the user with the text given inside
</script>
</body>
</html>
```

1. Save the file in your desktop as a .html or .htm (eg: FirstJS.html)
2. Double click on the file FirstJS.html. The program will open on a web browser and hurray!! A message box with the text “Hello World” with an “OK” pops up!!



3. Now, Click on the “OK” button!!

Simple isn't it? Now, let us understand the code. This is a very simple HTML code with an extra `<script>` tag which tells the browser that the JavaScript code follows. The JavaScript code displays an alert box with the text given inside quotations.

### Methods of including JavaScript Code

JavaScript code can be used in web pages in three ways:

1. Inside the HTML file: using `<script>` and `</script>` tags (see sample above).
2. Using an external JavaScript file: By providing the location of the file using the SRC attribute (sample below).

```
<script src = "First.js"> </script>
```

3. In HTML events: By incorporating JavaScript within the body element as HTML events; scripts are executed when triggered manually or automatically. For example, a popup may be displayed when a visitor fails to fill a particular field in a form, display a confirmation popup once the form is filled successfully or redirect a visitor to a website when clicking a button on a web page (sample below).

Example of HTML event *onclick* to Redirect user to a Website:

```
<html>
<head>
<title>Redirect visitor to a website</title>
<script language="javascript" > </script>
</head>
<body>
<button onclick="window.location= 'http://www.w3schools.com/'">
Click here</button>
</body>
</html>
```

An example of an *onload* HTML event:

```
<html>
<head>
<title> JavaScript Program using a HTML event </title>
</head>
<body onload = "alert('Hello World');">
```

```
Testing Javascript code
</body>
</html>
```

Some of the HTML events include onclick, onsubmit, onfocus, onkeypress, onmousedown, onmouseup, etc.

### Making your website user friendly.

Remember we said that JavaScript is user friendly and that you could have popups and redirections done when you visit websites? Let us now see how to do it.

JavaScript has 3 types of dialog boxes available - an **alert box** which displays a message to the user, a **confirm box** which asks the user to confirm something and a **prompt box** which waits for input from the user

#### Example: Alert box

```
<html>
<head>
<title> JavaScript Program to display user friendly options </title>
<script language="javascript" type = "text/javascript"> </script>
</head>
<body>
<input type="button" onclick="alert('Hey,this is an alert using
JavaScript');"
value="Alert Box Example" />
</body>
</html>
```

#### Example: Confirmation box

```
:
:
<input type="button" onclick="confirm('Are you sure you want to delete
this?');" value="Confirm Box Example" />
```

#### Example: Prompt Box

```
:
:
<input type="button" onclick="prompt('Please enter your favourite
subject','computers');" value="Prompt Box Example" />
```

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Write a program with alerts to: a) Display the text “Good Morning Sir”. The JavaScript code must be in the <head> tag. b) Display the text “I am learning JavaScript”. The JavaScript code must be in an another file called “MyJavascript.js”
2.	Write a program to display your name, school and class teacher with an onload event.
3	Write a program to display an alert box which says “Welcome to the world of computing”
4	Write a program to display a confirmation box which says “Did you enjoy the session?”
5	Write a program to display a prompt box which says “Please give your feedback here”

## ASSESSMENT

Answer the following questions:

1. Where will we run a JavaScript program?
2. What are the steps needed to create and run a simple program using JavaScript?
3. Which attribute of the <script> tag helps you include an external JavaScript file?
4. What are the three ways in which JavaScript can be included?
5. JavaScript is “interpreted” - what does this imply?

Fill in the blanks:

1. A file which ends with .js is an \_\_\_\_\_.
2. JS programs are included within \_\_\_\_\_ & \_\_\_\_\_ of a HTML document.
3. Generally HTML is preferred for creating \_\_\_\_\_ while CSS is preferred for \_\_\_\_\_ and JavaScript is preferred for \_\_\_\_\_

## SESSION 14: JAVASCRIPT SYNTAX

### RELEVANT KNOWLEDGE

JavaScript syntax is the set of rules that define a structured JavaScript.

#### Case Sensitivity:

JS is case sensitive. For example, in the previous session the word “alert” has a lower case “a”. If we type the word with an uppercase “A” then JavaScript will show an error and will not display the alert box.

#### Whitespace & Semi Colon

Whitespaces (Spaces & tabs) are used for the code to be understood better. It is good programming practice to leave spaces and write code legibly.

A semicolon is a line or statement terminator in JavaScript. It is optional and we need not have to use semicolons in our program. JavaScript will not show an error unless two statements are merged into a single line.

#### Comments

Comments in JavaScript are of two kinds : *Single line* and *Multi line* comments

1. Single line comment begins with two forward slashes  
Eg : `// This is a single line comment`
2. Multi line comments begin with `/*` and end with `*/`  
Eg: `/* This is a multi line comment  
and will continue across lines */`

Comments are ignored by the interpreter and are not executed. Comments play an important role in helping the programmer to understand complex logic.

#### Operators

An Operator is a symbol that is used to perform an operation. In this session we will see the Assignment operators, Arithmetic operators, Concatenation operators and the Comparison operators. Logical operators are dealt with in later sessions.

#### Assignment operators

Assignment operators help in assigning values to a variable. The table below shows how the assignment operators work.

Assume that  $x = 9$  and  $y = 2$ , then the assignment operators will give the following results.

Operator	Example	Same As	Result
=	$x=y$		$x=2$
+=	$x+=y$	$x=x+y$	$x=11$
-=	$x-=y$	$x=x-y$	$x=7$
*=	$x*=y$	$x=x*y$	$x=18$
/=	$x/=y$	$x=x/y$	$x=4.5$
%=	$x%=y$	$x=x\%y$	$x=1$

The equal to sign = is used to **assign** a value to a variable.

**Note:** In JavaScript, check for an **equal to** condition (with regard to the value) using two equal to signs i.e with ==. To check if the value is **exactly equal to** (with regard to the value and type) we make use of 3 equal to signs i.e ===

### Arithmetic Operators

These are your standard math operators. They include Addition, Subtraction, Multiplication, Division, Modulus, Increment and Decrement operators.

Operator	Description	Example
+	Addition	$x=10+5$
-	Subtraction	$x=10-2$
*	Multiplication	$x=10*2$
/	Division	$x=10/2$
%	Modulus (division remainder)	$x=10\%2$
++	Increment	$x=++y$ $x=y++$
--	Decrement	$x=-y$ $x=y--$

### Concatenation operator (+ Sign)

The + Operator is also used to concatenate strings i.e to add strings together.

Note:  $X = \text{"Welcome"} + 5$  gives an output Welcome5 and  $X = \text{"45"}+5$  gives 455 (Remember: Anything within quotes becomes a string).

The Modulus (%) operator will return the **remainder** obtained after the division. So  $10\%3$  returns 1 as that is the remainder when you divide 10 by 3.

## Comparison Operators

Comparison operators are used to compare any two things and take action depending on the condition. For example, you can display an alert box asking for PAN number if the amount deposited in a bank is greater than Rs.50000.00. This is done by comparing the amount entered with the minimum amount beyond which a PAN number is mandatory.

## Comparison Operators

Operator	Description
==	is equal to
===	is exactly equal to (value and type)
!=	is not equal
!==	is not equal (neither value nor type)
>	is greater than
<	is less than
>=	is greater than or equal to
<=	is less than or equal to

Let us now see a program which will show you how to use the assignment operators:

```
<html>
<body>
<p>Given that y=7, calculate x=y/2, and display the result.</p>
<button onclick="myAssignment()">Try it</button>
<p id="demo"></p>
<script>
function myAssignment()
{
var y=7;
var x=y/2;
var demoP=document.getElementById
("demo")
demoP.innerHTML="x=" + x;
}
</script>
</body>
</html>
```

Copy and paste this code onto a notepad, name it with a .html extension and run it on the web to see the results.

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a program to: Add two numbers and display the sum in an alert box. Subtract two numbers and display the difference.
2.	Assume Rama has 25 pencils which he wants to give to 10 students. If he is to give an equal number of pencils to each of them, create an alert to specify how many pencils each of them will get and how many will be left with Rama.

## ASSESSMENT

Answer the following questions:

1. Explain the role of case sensitivity in JavaScript.
2. Explain the use of comments and how are they added in JavaScript.
3. What is the difference of output that you get when you use the following operators `=`; `==` ; and `===`.
4. Explain what you understand by the modulo operator.
5. Specify four types of operators that can be used in JS.

Fill in the blanks:

1. is a set of \_\_\_\_\_ that define a structured JavaScript.
2. Plus sign is used to \_\_\_\_\_ numbers and strings.
3. When I want to display an alert which says 'Welcome' and then the name of the student, I will have the use the \_\_\_\_\_ operator.
4. If  $x = 14$  and  $y = 3$  then the output of  $x^*=y$  is \_\_\_\_\_ .
5. if  $x = 15$  and  $y =3$  then the output of  $x/=y$  is \_\_\_\_\_.

## SESSION 15: DATA TYPES

### RELEVANT KNOWLEDGE

#### Variables

Variables can be thought of as named containers in the computer's memory. You can place data in these containers and refer to the value using the name. For example, to store the name of a fruit and the rate of the fruit during program run, we need storage locations which are easily identifiable.

A Variable consists of an identifier, a scope and a specific data type.

**An Identifier** the name of a variable. It must begin with a character, a dollar sign, or an underscore. Do not use reserved keywords as variables.

Following table lists the reserved Keywords:

break	else	new	var
case	final	return	void
catch	for	switch	while
continue	function	this	with
default	if	throw	delete
in	try	do	instanceof
typeof	interface	null	undefined

Following are some examples of variables:

```
var x                // declare the variable using the var keyword
X = 10              // use the = operator to assign a value
var x = 10;         // assigning a value to a variable when you declare it
var y = 5;          // y is a variable that holds a value of 5
var sum = x + y ;   // sum is a variable that stores the sum of variables
                    x and y
```

Note that **var** is the keyword in JavaScript to declare the variable.

You can separately declare the variable and then assign values to it or straight-away declared and initialize the variables.

#### Variable Scope

The scope of a variable refers to the area of the program inside which it is accessed. Variables have two types of scopes - global scope and local scope.

*The Global Scope:* Look at the example - the rate of sugar is fixed in the fair price shop by the government and is the same throughout the state in all the fair price shops - This is an example of the variable named SugarRate being declared globally.

*The Local Scope:* Look at the example -The rate of sugar in different retails shops will be specific to the rating norms of each shop -this is an example of a local variable and is specific within the scope of each shop.

Note: You should NOT declare the same variable twice inside the same scope.

Example:

```
var x = 10;
var y = 5;
var sum = x + y ;
var firstName = "Jawaharlal"; //Variable is a string here
var lastName = "Nehru"
var completeName = firstName+ " " + lastName;
x = 7 //This is allowed
var x = 7 // NOT allowed as x has already been declared
```

**Data Types:** JavaScript uses a number of data types such as Numbers, Strings, Boolean, Arrays and Objects.

Data Type	Examples	
Numbers	123, 54.45, 88.3,1543e2	integers, decimals and extra large/extra small numbers with scientific notation
String	"This is a string"	series of characters within quotes
Boolean	True, False	True or False values. Normally used for conditional testing.

Note: You can use quotes inside a string as long as they are not the same as the quotes surrounding the string. For example, "It's my turn to take the wheel now" is accepted as a string while "His name is "Rama" and he will be here soon" is not.

Complex data type such as Arrays and Objects are dealt in the later sessions.

Please note JavaScript is a **loosely typed language**. You need not declare if a variable is a number or a string. During runtime, based on the value assigned, it automatically decides the data type.

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Write a program to calculate the total_salary of an employee which is given as the sum of the basic and the hra minus the tax_deduction. Use the variables total_salary, basic , hra and tax_dd.
2.	Write a program to display an alert which Welcomes you specifying your name and then says “Have a great day”. Example: “Welcome NAME Have a Great day” (Replace NAME with your name).

## ASSESSMENT

Answer the following questions:

1. Explain if the variables “name” and “Name” similar.
2. Explain the rules to write an identifier.
3. What do you understand by the term local scope?
4. All students have to come to school by 9 A.M . Would this be declared as a local scope or a global scope?
5. Will this be an acceptable string data type “Yay I’m getting through’ this”?

Fill in the blanks:

1. \_\_\_\_\_ should not be used as a variable name because it is a reserved keyword in JavaScript.
2. JS has two types of variable scope \_\_\_\_\_ & \_\_\_\_\_ .
3. JavaScript variables are declared using \_\_\_\_\_ as the keyword\_
4. You should\_not declare the same variable \_\_\_\_\_ in the same scope.

## SESSION 16: LOGICAL OPERATORS AND CONDITIONAL STATEMENTS

### RELEVANT KNOWLEDGE

#### Logical Operators

The AND, OR and NOT operators are logical operators which check a number of conditions and specify if the result is true or false.

For example, if  $a = 5$  and  $b = 3$ ,

Operator	Description	Example
<code>&amp;&amp;</code>	Called Logical AND operator. If both the conditions are satisfied then condition becomes true.	<code>(a == 5 &amp;&amp; b &gt; 2)</code> is true.
<code>  </code>	Called Logical OR Operator. If any one of the two conditions is satisfied then the condition becomes true.	<code>(a &gt; 5    b &lt; 5)</code> is true.
<code>!</code>	Called Logical NOT Operator. Used to reverse the logical state of its operand. If a condition is true then the Logical NOT operator will make it false	<code>!(a == 5 &amp;&amp; b &gt; 2)</code> is false. <code>!(a &gt; 5    b &lt; 5)</code> is false

#### Conditional operators

JavaScript assigns a value to a variable based on the output of some condition using the Conditional operator.

#### Syntax for Conditional operator

```
variablename = (condition)?value1:value2
```

For example if the amount deposited in a bank is greater than 50000 then you need to send an alert to get the PAN, if the amount is not greater than 50000, normal operations can continue. This can be written as

```
amt_deposited = (greater than 50000) ? "get PAN" : "continue"
```

#### Conditional Statements

There are situations where a condition needs to be checked and if it is true, some action needs to be taken. In such cases, you use the if statement

#### Syntax for If statement

```
If (condition) {
```

```
// Execute code if the condition turns to be true
}
```

For example, if the time is less than 11 A.M say “Good Morning”

Sometimes we come across situations where we need to do a particular set of actions based on the result of a comparison of two quantities. For example if the mark of a student is greater than or equal to 40 you give him a ‘Pass’ and if it is less than 40 you specify it as ‘Fail’

In such cases the “IF-ELSE” syntax is what you will use.

Syntax for if-else statement

<pre>If (condition) {     //Execute for positive results } else {     //Execute for negative results }</pre>	<pre>}</pre>	<pre>If condition is satisfied execute the Block. Code written inside flower brackets "{ }" is called a block.</pre>
<pre>}</pre>	<pre>}</pre>	<pre>else Block</pre>

**Note:** In the above syntax after “if” there is a paranthesis/brackets ().The condition that needs to be executed is written inside the brackets. There are NO brackets () for the else.

Consider a situation, where we pick a card with a number on it. It is decided that based on a condition which tests the value on the card, you should increment or decrement the value. This can be done using the code given below.

```
var number = 10; //Assume the number on the card is 10
if(number > 5) {
    number --; // Decrement the number
} else {
    number ++; //Increment the number
}
alert("The value of the number is " + number);
```

Once the above code gets executed the user will get an alert message which says “The value of the number is 9”. This happens because 10 is greater than 5 and hence satisfies the ‘if ‘condition and executes the first block of code which decrements the value of the number by 1.

We can also compare strings. Consider a case where we have to give a “Green” ball to Rahul and a “Blue” ball to “Ravi”

```
var person = "ravi "

if(person == "rahul"){
  alert("Give a green ball");
}
else {
  alert("Give a blue ball");
}
```

**Note** that the output here will be “Give a blue ball”. This happens because the person is not “rahul” so, the **else block** has been executed.

### If-else if - else

JavaScript allows nesting of if statements using if- else if- else. There are a number of scenarios where we might have to do different operations for different sets of values based on specific conditions.

**Example:** Assume that a person has to pay a tax deduction of 30% if his salary is equal to or greater than Rs 15 lakh per annum , 20% if it is between Rs 10 lakh & Rs 15 lakh, 10 % if it is between Rs 5 lakh & Rs 10 lakh and no tax deduction if it is less than Rs 5 lakhs. This can be captured using the if - else if - else statement.

```
if(condition) {
  //IF condition satisfied
} //END OF IF
Else if() {
  //ELSE IF 1 condition satisfied
} //END OF ELSE IF 1
else if() {
  //ELSE IF 2 condition satisfied
} //END OF ELSE IF 2
else {
  //None of the above conditions satisfied
} //END OF ELSE
```

**Note** that the nesting of if statements happens in an ordered way. First the **if block** gets completed and subsequently the **else if blocks** and finally the **else block**.

**Switch:** This is an enhanced version of the if-else statement. The Syntax for the switch statement is as given below.

Syntax	Example
<pre>switch (expression){ case value1: statement;</pre>	<pre>var a = 1;  switch(a) { case 1:</pre>

```

break;
case value2:
statement;
break;
default : statement;
}
alert("Case 1 is executed");
break;
case 2:
alert("Case 2 is executed");
break;
case 3:
alert("Case 3 is executed");
break;
default:
alert("Default is executed");
}

```

The Switch statement always begins with the keyword "switch" and the expression. In the expression, a variable is evaluated once. The value of the expression is then compared with the value of the case. If there is a match with the case value, that particular case executes and breaks the loop with the "break" statement at the end of the case. This prevents the other code to be executed. If none of the case values match with the value of the expression the "default" is executed.

**Note:** In the above example case 1 is executed. If we had declared the variable a=5 then the default would have been executed.

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Write a program to: <ul style="list-style-type: none"> <li>Determine if a student has passed in his exam or not. If the marks are greater than 40 then he gets a pass else he gets a fail.</li> <li>Determine what grade he has obtained based on marks obtained if 90-100 is A+; 80-90 is A ;70-80 is B+; 60-70 is B ; 50-60 is C; 40-50 is D and Below 40 is fail [Hint : Use the if-else if..-else]</li> </ul>
2.	Write a program using the switch statement to print the names of Class Teachers of all grades from 1 to 10. Hint: let the variable be the grade. Depending on its value, print the names of the class teachers .
3	If a student has an amount greater than Rs 50 he uses an auto , else if he has an amount between Rs 20 to Rs 50 he uses the share auto else if he has between Rs 5 to Rs 20 he uses the public bus or Else he walks to his school. Write this using the if-else if -else condition

## ASSESSMENT

Answer the following questions:

1. What is the purpose of the break statement?
2. What is the purpose of the switch statement? Give some examples .
3. What is the use of a conditional operator? Give some examples.
4. Explain what a logical NOT operator does using some examples.

Fill in the blanks:

1. Switch statements start with \_\_\_\_\_ keyword.
2. If none of the case values in the switch statement match the \_\_\_\_\_ is executed.
3. \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_ are all conditional statements that can be used in JavaScript
4. \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_ are all logical operators used in Java Script

## SESSION 17: FUNCTIONS

### RELEVANT KNOWLEDGE

A Function is a piece of code that does not execute until you specifically call it. Functions save time when doing repetitive tasks. The code can be reused in many places across files. This saves a lot of code from been re-written unnecessarily.

Depending on the need you can write the function on your own or reuse functions written by other professional programmers. If you decide to use someone else's function then you just need to give the proper input!! For example, consider the function as a motorbike. We all might know how to ride the bike but need not know how to manufacture the bike!!

Syntax	Example
<pre>/*Function declaration */  function &lt;name&gt;() {     //your logic goes here }  &lt;name&gt;(); /*Call the function*/</pre>	<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;First JavaScript Program&lt;/title&gt; &lt;script&gt;     ↑     function popup() {         alert("Hello World!!!");     }     ↓ &lt;/script&gt; &lt;/head&gt; &lt;body onload="popup();" &gt; &lt;/body&gt; &lt;/html&gt;</pre> <p>Function is declared</p> <p>Function is called</p>

**Note:** The syntax for a function is started by typing the keyword “function” followed by an appropriate name and the open and close brackets. The flower brackets { } represent the block or the length of the function.

The line with `<body onload="popup()">` is where the function is called and on executing the above example we get the Hello World alert box.

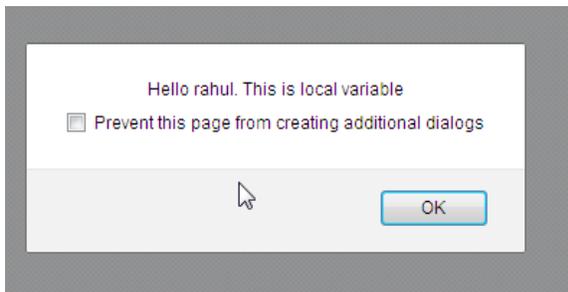
Let us revisit the Scope of a Variable and understand it better using functions.

```
<html>  
<head>  
<title>First JavaScript Program</title>  
<script>  
var a = 10;  
function popup() {  
    var a = "rahul "  
}
```

```

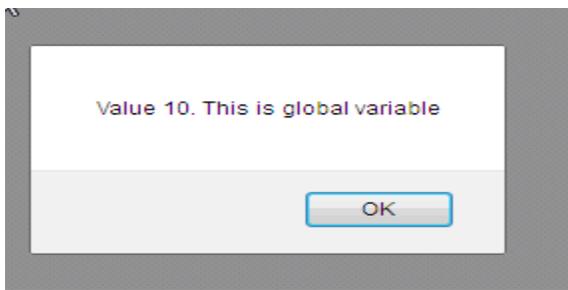
    alert("Hello " + a + ". This is a local variable");
}
popup();           //Function is called
alert("Value " + a + ". This is a global variable");
</script>
</head>
<body>
</body>
</html>

```



Notice that “var a” is in two places - one just after a script tag which holds an integer value “10” and another inside the function that holds the string variable “rahul”.

The string value stays inside the function and is said to be ‘local’ to a function.



Global variable:

This scope is lost outside the function block and when it is called outside the function, it takes the value of 10 which is the value of the global variable.

**Arguments:** These are variables that the function is supposed to do something with. For example consider a function called **add** where we pass 2 arguments a & b,

```

function add (a,b) {
    var c = a+b;
    alert("The sum of the two numbers is " + c);
}
add(5, 6);

```

Here the values that you pass are a=5 & b=6 and the Variable c gets a value equal to value of a + value of b. The output will be “The sum of the two numbers is 11”.

You can pass any number of arguments - function compute(a,b,c,d,e) is valid

**Return:** JavaScript allows you to return a single value from a function. The return statement must be the last line in the function. In the last example we passed 2 numbers as arguments to a function and got the sum as a result. This result can be returned and have a return type as a number, string or a Boolean value.

return 0; return "a string"; return true are all valid return statements

```
function add (a,b) {  
    var c = a+b;  
    return c;  
}  
var sum = add(5,6);  
alert("The sum of two numbers " + sum);
```

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a program using functions to: a. Pass 3 arguments a,b and c to a function. Add the values a & b. Multiply c with the sum of a & b. b. Pass your name as an argument and display "Hello <yourname>"
2.	Write a program and pass a number to a function and determine if the number is greater than 50. Return true if it is greater and false if it not.

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of a function.
2. Explain the use of a return statement.
3. What do you understand by local and global variables? Explain using a function specifying the value of the local variable and the global variable.

Fill in the blanks:

1. A \_\_\_\_\_ is not executed until you call it.
2. The Return statement returns a \_\_\_\_\_ value to the calling statement.

## SESSION 18: LOOPS

### RELEVANT KNOWLEDGE

Loops help in performing the same set of operation over and over again with different values for the variables. This reduces the number of lines of code written.

#### Understanding the effect of a loop

Consider a situation where you need to add the numbers from 1 to 9, our code will be as follows:

```
function add(a, b) {  
  var c = a+b;  
  return c;  
}  
var sum1 = add(1,2);  
var sum2 = add(sum1,3);  
var sum3 = add(sum2,4);  
var sum4 = add(sum3,5);  
var sum5 = add(sum4,6);  
var sum6 = add(sum5,7);  
var sum7 = add(sum1,8);  
var sum = add(sum1,9);  
alert("The sum of 1 to 9 is = " + sum);
```

Same set of code, calling add function. Just that the second argument increases sequentially. The variables sum1 to sum7 is never used.

Notice that the 8 lines of code are written to do the same operation. It is just that the second argument increases sequentially with an increment value of 1. You can do this better with a loop. Given below is an example for the *for loop*.

```
function add() {  
  var a = 1;  
  var c = 0;  
  
  for (a = 1; a < 10; a++) {  
    c += a;  
  }  
  
  return c;  
}  
var sum = add();  
alert("The sum of 1 to 9 is = " + sum);
```

Executes for 9 times until a = 9. Stops when a=10 which fails the condition

The add() function is called here

**Note** that the add function is called only once. The **for loop** executes repetitively from a=1 to a=9 with an increment of 1. The line c+=a gets executed 9 times, until the condition a<10 is satisfied.

We will be seeing 3 types of loops in this session namely the:

- for loop - used when the number of iterations is known
- while loop - loops through a block of code as long as a condition is true
- do while loop - similar to the while loop, but checks the condition at the end of the first iteration. So, the loop executes at least once

## For Loops

The For loop is best suited when you already know the number of times the statements will be executed. The loop executes until the condition becomes false.

### Syntax for the For Loop

```
for(initialization; condition; increment) {  
    //Statements  
}
```

### When a for loop executes

1. The initializing expression is executed. This expression usually initializes one or more loop variables.
2. The condition expression is evaluated. If the value of condition is true, the loop statements execute. If the value of condition is false, the for loop terminates.
3. The increment expression executes and increments the value by the specified step value.
4. The statements execute, and control returns to step 2.

In our example above the expression **a=1** initializes **a** to 1, the expression **a< 10** is the condition telling the loop to continue as long as **a** is less than 10. Finally the expression **a++** increments the value of **a** by 1.

### Example code for a for loop

```
<html>  
<body>  
<p>Click the button to loop through a block of code four times.</p>  
<button onclick="myForLoop()">Try this</button>  
<p id="demo"></p>  
<script>  
function myForLoop()  
{  
var x="";
```

```

for (var i=0;i<4;i++)
{
  x=x + "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML=x;
}
</script>
</body>
</html>

```

## While Loops

The While Loop is another commonly used loop in JavaScript. The purpose of the while loop is to execute a block of statements over and over again until the condition fails. It is best suited in a scenario where we don't know beforehand as to how many times the loop will execute.

### Syntax for the While loop

```

while(condition) {
  //statements
}

```

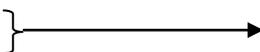
The example discussed in the for loop can also be modified to use the while loop

```

function add() {
  var a = 1;
  var c = 0;
  while (a < 10) {
    c += a;
    a++;
  }
  return c;
}
var sum = add();

alert("The sum of 1 to 9 is =" + sum);

```



Executes until a = 9. Stops when a=10 which fails the condition

## Do While Loops

This is another loop slightly different from the for loop and the while loop. This loop will execute the statement at least once. i.e. the statements inside the loop will always get executed at least once, **even if the condition is false**. The condition check happens after the loop has executed. The loop will continue to execute or will terminate based on the condition. The Syntax is

```

do {
  //Statements
}

```

```
} while(condition);
```

## Break and Continue

The break statement is very useful to break the loop in the middle of an execution.

Consider a real world situation where you are trying to check if the word 'JavaScript' is present in any of the 10 sentences given using a **for loop**. The computer finds it on the 3<sup>rd</sup> sentence, but keeps continuing the loop unnecessarily till the end. By using the **break statement** here, we can exit the loop and stop the unnecessary execution.

The **Continue statement** is used inside loops so that we can skip the execution of a loop for a particular value and continue with the execution from the next iteration.

Consider a situation where you should execute the loop from 1- 10 except for the condition where  $i == 3$ . This is accomplished with the **continue** statement.

In a real world situation, you could have a rule which says all students need to be in at 9 AM or they will have to stand out of the class. The class leader alone could be exempted from this rule as he is given a grace time of 10 minutes due to his duties.

Break	Continue
<pre>for (i=0; i&lt;=10; i++) {   if (i==3){break;}   alert("The number is " + i); }</pre>	<pre>for (i=0; i&lt;=10; i++) {   if (i==3){continue;}   alert("The number is " + i); }</pre>
loop executes from $i = 0$ to $i = 2$ but when $i$ becomes 3 the loop is terminated	the loop is executed for 9 times except for the condition when $i = 3$ where the loop is skipped

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a program using the for loop: a. Add all even numbers and odd numbers from 1 to 20. Write a single function to do the addition. Display the results. b. Run the loop from 1 to 100 and print the numbers 2,4,9,16,25,36,49,64,81 and 100.
2.	Create a program to reverse a string [Example: input string = school, output = loohcs] using the while loop

## ASSESSMENT

**Answer the following questions:**

1. Explain the purpose of loops in JavaScript.
2. Explain the while and do-while loops and when to use each of them.
3. Explain a real world situation where the break statement helps.
4. Explain where the continue statement could help.

**Fill in the blanks:**

1. \_\_\_\_\_ execute the same set of operations over and over again.
2. The \_\_\_\_\_ executes at least once irrespective of the condition.
3. \_\_\_\_\_ statement skips the loop execution for a particular iteration.
4. The \_\_\_\_\_ statement exits the loop and stops unnecessary execution of the loop.

## SESSION 19: ARRAYS AND OBJECTS

### RELEVANT KNOWLEDGE

#### Arrays

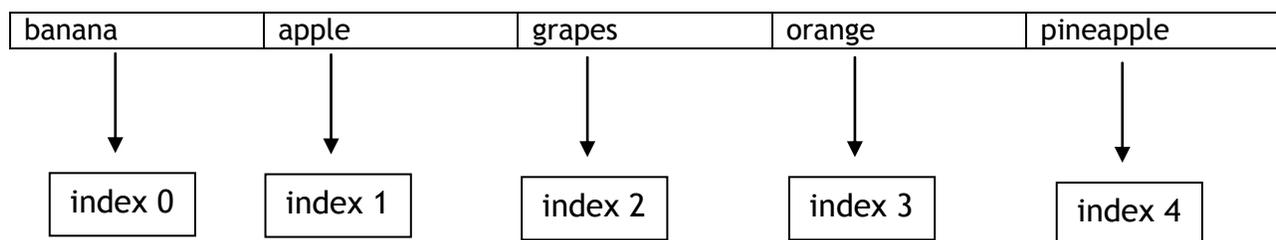
An array is another kind of variable that can hold many values. We can use arrays to create a shopping cart on the web.

Until now, you must have noticed that only a single value can be assigned to a variable. Like `a = 5` or `s = "rahul"` etc

Consider a case where you want to store 5 fruit names. We can store them using 5 different variables or use a single array variable.

Variable	Array variable
<pre>var banana = "banana"; var apple = "apple"; var grapes = "grapes"; var orange = "orange"; var pineapple = "pineapple";</pre>	<pre>var fruits = ["banana", "apple", "grapes", "orange", "pineapple"] ;</pre>

Notice that instead of writing 5 statements we wrote just a single statement using the array variable and achieved the same output. You can think of an array as a container storing multiple values sequentially. Each storage location is called an index and is referred to by a number starting from 0. Accessing the values in the array is through the index number.



There are 3 ways to create an array in JavaScript.

1. Literal

```
var fruits = ["banana", "apple", "grapes", "orange", "pineapple"] ;
```

2. Regular

```
var fruits = new Array();
```

```

var fruits[0] = "banana";
var fruits[1] = "apple";
var fruits[2] = "grapes";
var fruits[3] = "orange";
var fruits[4] = "pineapple";

```

### 3. Condensed

```
var fruits = new Array("banana", "apple", "grapes", "orange", "pineapple");
```

Notice that the fruits array created by step 2 & 3 has the keyword "Array" and "new". Arrays are treated as objects in JavaScript. You will learn about objects soon. Note that we created an array with 5 elements in it. So, the length of the array is 5. If you create an array with 12 elements the length of the array is 12.

#### Accessing an Array

Arrays are accessed by the index numbers.

```

var fruit = fruits[2];      //Grapes is accessed
var fruit = fruits[4];     //pineapple is got

```

**Note:**Remember if we access the array with an index number greater than the length of the array then it will throw an error "undefined".

For example,

```
var fruit = fruits[5]; // Throws an error undefined.
```

#### Functions & Properties

JavaScript comes with a set of predefined functions and properties. They are very useful when we use them in a program.

Imagine you need to find the length of an array, sort an array in an alphabetical order, add new items in the array or delete one or more elements. In these cases these predefined functions are very handy.

For example : `fruits.length;` // returns the length of an array which is 5 in our case.

function	usage	Description
sort()	fruits.sort();	Sorts and returns the array. The new values will be ["apple", "banana", "grapes",

		“orange”, “pineapple”]
indexOf()	fruits.indexOf(“banana”)	Returns the index number of banana which is 0
reverse()	fruits.reverse()	Reverses the array elements
splice()	fruits.splice()	Adds/Removes an element from the array
push()	fruits.push()	Adds a new element at the end of an array
pop()	fruits.pop()	Deletes the last element of the array and returns that element

## Objects

An object is a complex data type that can hold anything. Consider an object as a big container that can have numbers, strings, arrays and other objects as well.

Objects have properties and methods. For example a car is an object which has properties such as color, size, doors, windows etc and methods to go forward, take reverse etc. In JavaScript, objects can be accessed using the dot (.) operator.

### Examples of Objects in JavaScript

Date - Displays the current system date

document - The HTML file is considered as an document object

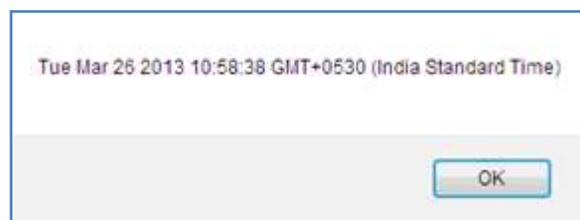
String - The string value is an object

Array - Array variable is an object

### Date

```
var date = new Date(); // This prints the current system date and time.
```

```
date.getHours(); // Returns the Hour
```

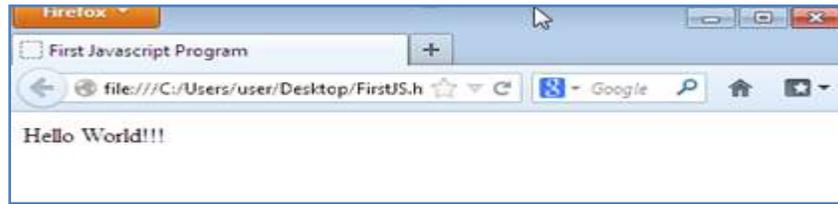


### Document

The document object provides access to all HTML elements in a page. The write() function of the document object writes to the HTML document.

```
document.write(“Hello World!!!”);
```

This statement writes Hello World text on the browser as shown below.



## String

```
var name = "rahul";
```

```
name.length;           // This will return the length of the string name which is 5
```

**Note:** Apart from the objects mentioned above, we have a number of objects available in JavaScript:

- Window
- Object
- Boolean
- Math
- RegExp
- WeakMaps
- Maps
- Sets

## Additional Reading

Now you have learned about the basics of JavaScript, use the following resources for enhancing your knowledge on JavaScript:

1. <http://en.wikipedia.org/wiki/JavaScript>
2. <http://www.w3schools.com/js>
3. <http://www.javascriptkit.com>
4. <https://developer.mozilla.org/en/docs/JavaScript>
5. <http://www.dynamicdrive.com>

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a program to: a. Store the numbers from 1 to 10 in an array. Use the for loop for the numbers. b. Define an array of vegetables and sort them in the alphabetical order. c. Display today's date using a date object. Use the document object to display the results
2.	Define a string variable and determine the length of the string.

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of an array.
2. Explain the procedure to access an array.

Fill in the blanks:

1. An Array is accessed by the \_\_\_\_\_ numbers.
2. \_\_\_\_\_ property is used to find the number of elements in an array.
3. The properties and methods of an object are accessed using the \_\_\_\_\_.

## SESSION 20: OVERVIEW OF DHTML, AJAX AND JQUERY

### RELEVANT KNOWLEDGE

DHTML or Dynamic HTML refers to sets of technologies used together to create interactive or animated websites. Typically DHTML uses the following combination:

1. HTML (for document structure)
2. CSS (for presentation, used for quickly changing styles)
3. Client-side scripting language (for interactions or animations from visitors)

DHTML is generally used for changing variables which in turn affects the look and feel of content. For example visitors may be allowed to pick a theme for the website by presenting a dialog box to choose from a list of themes. Another classic example is the navigation menu presented as a dropdown menu similar to the ones used in applications and a ticker that updates cricket scores time to time.

You may find samples of navigation menus, animation effects, tooltips, image effects, scrolling menus, form effects, etc. in the following websites:

- <http://www.dynamicdrive.com/>
- <http://www.dhtmlgoodies.com/>
- [http://www.quackit.com/dhtml/dhtml\\_examples.cfm](http://www.quackit.com/dhtml/dhtml_examples.cfm)

View samples of DHTML and see the kind of things that can enhance the look and feel of websites. Refer to the website for detailed information to integrate them in web pages. If you observe, most of them are developed using a combination of HTML, CSS & JavaScript.

Following is a list of URL's that you can use for enhancing your knowledge:

- [http://en.wikipedia.org/wiki/Dynamic\\_HTML](http://en.wikipedia.org/wiki/Dynamic_HTML)
- <http://www.htmlgoodies.com/beyond/dhtml>
- <https://developer.mozilla.org/en/docs/DHTML>
- <http://www.javascriptkit.com/dhtmltutors/>

### DHTML Navigation Menus

You can use attractive navigation dropdown menus using DHTML scripts or a product such as DHTML Menu builder lite that creates the menu for you.

1. Download DHTML Menu builder lite from  
<http://www.xfx.net/utilities/dmbuilderlite/download.php>
2. Follow the on-screen instructions and complete the installation.

### 3. Select *Start > Programs > DHTML Menu builder*

Refer to the help file for further instructions on creating a navigation menu and to integrate that in your website.

## **AJAX**

AJAX or Asynchronous JavaScript & XML is set of techniques used for developing asynchronous web applications. Usually data is downloaded to the visitor's computer and displayed by the web browser; when additional data needs to be viewed, browser sends request and the response is again displayed in the web browser. In such cases, the entire web page will reload to display the new set of data. For example when you use an online shopping website, you may notice the first 10 results are displayed and when displaying the next set of results (11-20), the entire page refreshes. In such cases you can write code to fetch the next set in the background without interfering with the display of existing web page resulting in rich experience for the visitor.

AJAX uses a combination of HTML, CSS, XML and JavaScript. AJAX is widely used today and is implemented through popular Ajax frameworks such as JQuery, MooTools, Prototype, YUI Library, Dojo Toolkit, etc.

Use the following websites to find samples of AJAX framework, refer to the website for detailed instructions on integrating them in your website.

- <http://jquery.com/>
- <http://mootools.net>
- <http://prototypejs.org/>
- <http://yuilib.com/>
- <http://dojotoolkit.org/>

Note: These website provide complete tutorials on learning and using them.

## **JQUERY**

JQuery is one of the most popular web application framework used for many web applications. You may have noticed fancy effects such as slideshows, collapsible tables, etc. in few websites that look attractive (sample below).



### **Slideshows**

You can use JQuery to produce such effects. Use the search engine to find samples or demos of sliders, etc. and refer to the website for including them in your web page. Most of them are easy to implement as the instructions are straight-forward.

### **EXERCISE**

Perform the following activities till you are confident:

S.No.	Activities
1.	Download a DHTML navigation menu and implement it in the website created in the earlier sessions.
2.	Download JQUERY and integrate it with a website. Modify the script to apply different slide effects.

### **ASSESSMENT**

Answer the following Questions

1. Explain the purpose of DHTML with an example.
2. Explain the purpose of JQUERY with an example.

Fill in the blanks

1. Acronym for DHTML \_\_\_\_\_.
2. Acronym for AJAX \_\_\_\_\_.

## SESSION 21: INTRODUCTION TO VBSCRIPT

### RELEVANT KNOWLEDGE

Visual Basic Script or VBScript is a popular scripting language developed by Microsoft and modeled on Visual Basic. You do not have to buy a copy of VBScript from the computer store or install a VBScript disk on your computer. All that you do is install the Internet Explorer (IE) browser, which will support VBScript. The IE has the VBScript engine built inside and has the ability to create Web pages with VBScript. It provides the functionality that is not possible through HTML coding.

VBScript is a lighter version of the programming language Visual Basic and comes with a fast interpreter. It has been installed by default in every desktop release of Microsoft Windows 98 and in the Windows server since Windows NT 4.0 Option pack.

#### Uses of VBScript

VBScript is a general-purpose scripting language and can be used for both Client-side and Server-side scripting. It is widely used among system administrators who work on the Microsoft environment and is the scripting language for the Quick Test Professional which is a test automation tool. Some embedded applications which are in use in the industry (e.g. - industrial operator interface) also make use of VBScript. When used for client side web development, it is similar in function to JavaScript. i.e. it is used to write executable functions which can be used in HTML pages and which interact with the Document Object Model (DOM) of the page to provide functionality which HTML alone cannot give. It is also used for server-side processing of web pages and is a pre-requisite for Microsoft ASP.(Active Server Pages).

#### Using VBScript

Consider VBScript as an add-on to HTML. While HTML is used for creating objects and CSS for presentation, VBScript is used for manipulating DOM objects to provide a particular function. For example you may have designed a feedback form including fields for First Name, Last Name, Age, Email Address, etc. However if you want to display popup text to guide the visitor to fill the details properly, you can use VBScript.

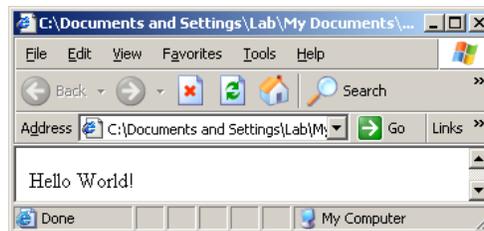
VBScript is easy and fun to learn. You can use a text editor such as notepad to create scripts, save them with the .html extension and run them using Microsoft **Internet Explorer** web browser.

You need to add special tags to indicate the VBScript code present within the HTML document. To indicate the presence of VBScript code, you need to place the code within `<script></script>` HTML tags. The script tag can be placed in the `<head>` tag or the `<body>` tag of the HTML.

Look at the following example:

```
<html>
<body>
<script type="text/vbscript">
document.write("Hello World!")
</script>
</body>
</html>
```

Notice that in the above example, the `<script>` tag has a type attribute which tells the browser that it's a vbscript. To print a text on the webpage use the `document.write()` command. The output is as follows on the IE Browser.



## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a simple program to: a. Display "Hello Class" b. Display "I am learning VBScript"

## ASSESSMENT

Answer the following questions:

1. Explain the uses of VBScript.
2. Explain the procedure to write a simple program using VBScript.
3. Expand DOM and ASP.

**Fill in the blanks:**

1. \_\_\_\_\_ is a program widely used for creating interactive web pages for Microsoft related technologies.
2. VBScript programs are included within \_\_\_\_\_ & \_\_\_\_\_ tag of a HTML document.
3. \_\_\_\_\_ command is used to print a text on to the HTML page.
4. VBScript can be used for both \_\_\_\_\_ as well as \_\_\_\_\_ scripting
5. VBScript is widely used among \_\_\_\_\_ who work on the Microsoft environment

## SESSION 22: SYNTAX AND VARIABLES

### RELEVANT KNOWLEDGE

#### Syntax:

Syntax is a set of rules. Here it is the set of rules that define a structured VBScript. VBScript has syntax which is identical to the Visual Basic programming language.

**Case-Insensitive:** VBScript is a case-insensitive scripting language. For example myFunction, MyFunction, myfunction are all the same in VBScript.

**No Semicolons:** A new line (and not a semicolon) represents the end of statement. For example,

```
document.write("No semicolons are used in the end of this statement")
```

**Underscore:** If a statement exceeds a line then we use an underscore (\_) to append the lines. For example,

```
document.write("Multi line breaker. This statement exceeds a single line " &_
```

```
"So we use the underscore" &_
```

```
"Ampersand symbol is the string concatenater in VBScript")
```

**Dot Operator:** The methods and properties of an object can be accessed using the dot(.) operator. For example,

```
document.write("Hello World!!!")
```

Here, document is an object and write is a method. Notice that the write method is accessed using the dot operator

#### VBScript variables - An overview

As you know, a variable is a name associated with a value and can be manipulated at different points in a program. These are named containers. In VBScript, variables are always of one fundamental data type - *Variant*. The Variant data type is used to declare a variable whose type is not explicitly given. This means that a Variant data type can hold any type of value you want.

VBScript variables can be classified broadly into *scalar variables* and *array variables*. The array variable is be of two types - *fixed* or *dynamic*.

Example : Declaring a Variable and retrieving its value :

```
<script type="text/vbscript">
Dim name
name = "Rahul"
document.write("My name is " " name)
</script>
```

When declaring a variable, using the *explicit method*, we use the *Dim* keyword followed with the name of the variable that you declare. The Dim statement tells VBScript that what follows is a variable. It declares and allocates storage space in memory for the variable. Here, name is a variable that has been explicitly declared. You can declare more than one variable using the comma separator. For example, if you want to create two variables called studentName, and studentMarks you declare it as Dim studentName, studentMarks

**Note:** The Dim statement must be used only once for a variable. After declarations are made, you use only the variable and not the Dim keyword again.

### Naming Variables

The following are the rules for choosing a variable name in VBScript:

1. Have meaningful descriptive names.
2. The length of the variable name must not be too long.
3. Spaces should not be used in variable names.
4. Variable names cannot have dot (.).
5. Should not use special characters like !@#%&\*^ . Variable names can start with an underscore.
6. The variable name must not exceed 255 characters.

Notice that in the above example to print a variable you DO NOT enclose the variable in double quotes.

### Scope of variables:

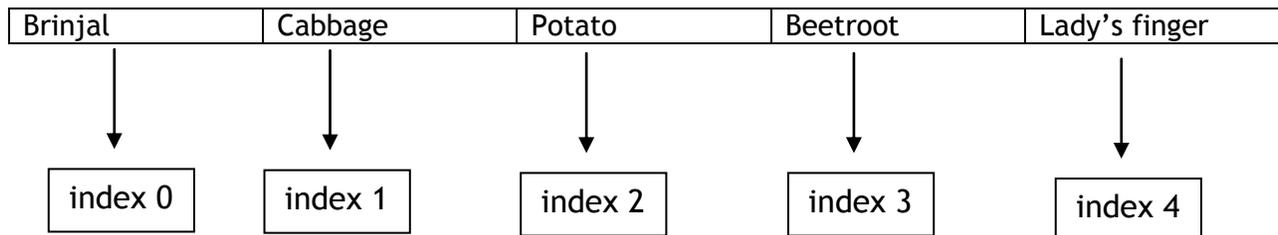
The life time of a variable is called the scope of a variable. Variables can be of two types of scopes - the local scope and the global scope. This will make more sense when learn about procedures.

Variables can be any of the following

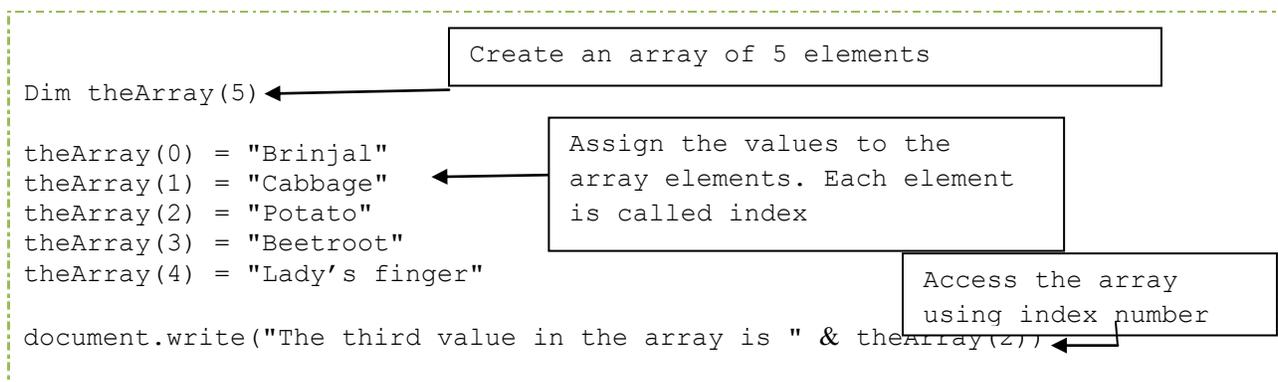
1. Number e.g.: 6,548.50, 2000
2. String e.g.: "This is a string"
3. Boolean e.g.: true or false
4. Array e.g.: ["brinjal", "cabbage", "potato"]
5. Object e.g.: document

### Arrays:

Just as you create scalar variables, you can also create array variables. These are variables which help you group a number of values together into a set. i.e. An array will help you create a variable with a single name, which contains a series of values inside it. Imagine a situation where you need numbers 1 to 100. You will use 100 different variables to store them. With the help of an array you can store all the numbers in a single variable. As shown below you can visualize an array as a storage place sequential in operation and which can be accessed with an index number.



This can be done using VBScript as:



Notice the syntax above shows us how to create an array, assign a value to array and access the elements in an array.

In VBScript you have two types of arrays - Fixed and Dynamic. In a fixed array, the number of elements that can be stored is pre-determined while in a dynamic array,

the number of elements stored may vary. (Remember the shopping cart which can have any number of elements that you decide to buy?)

We declare the two types of arrays using the Dim statement as shown below.

fixed array:           Dim arrStudentName(40)

dynamic array:       Dim arrBooksPurchased ().

Note: The dynamic array does not specify the number of elements it will hold. It can later be programmatically set using the ReDim statement which will be given as

ReDim arrBooksPurchased(nCount - 1) where you would have found the count of books selected programmatically and fixed the array size.

Declaring Constants:

A Constant performs the same function that a variable does, i.e. it is a symbolic replacement for an item of data in memory. The difference is that a constant has the same value throughout its lifetime. Constants are declared using the Const keyword.

For example Const PI = 3.414

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a program to: a. Declare a variable called schoolName and assign the school name to it. Display the results b. Declare a variable called fruit and assign a fruit name and display the results.
2.	Create a program to access 5 types of books - Text books, Humor, Biographies, Murder Mysteries and Personal Development books. These books should be taken into a shopping cart and should be accessible by index. (Hint : Use an array)

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of a variable.
2. Explain the naming rules to create a variable in VBScript.
3. Explain the purpose of an array and how to use it in a program
4. Explain the different kinds of arrays mentioning where they could be of use.

5. What is the difference between a variable and a constant.

**Fill in the blanks:**

1. VBScript is a case \_\_\_\_\_ language.
2. VBScript has two types of variable scope \_\_\_\_\_ & \_\_\_\_\_.
3. VBScript variables are declared using \_\_\_\_\_ as the keyword.
4. \_\_\_\_\_ is used to store multiple values in a variable.
5. \_\_\_\_\_ Array can be programmed to store any number of values depending on the count of choices made by the user.

## SESSION 23: OPERATORS AND CONDITIONAL STATEMENTS

### RELEVANT KNOWLEDGE

#### Operators:

Operators are used to perform an operation. For example, an operator is used to assign a value to the variable. Compare two values or variables. You can perform math on numbers, manipulate strings with these operators.

The different kinds of operators

1. Arithmetic Operators
2. Assignment Operator
3. Comparison Operators
4. Logical Operators
5. Concatenation Operators

#### Arithmetic Operators:

Operator	Description	Example
+	Addition	$7+6 = 13$
-	Subtraction	$15-6 = 9$
*	Multiplication	$6*7 = 42$
/	Division	$9/3 = 3$
Mod	Modulus (remainder of a division)	$15 \text{ mod } 10 = 5$
^	Exponentiation (raises the number to the power of an exponent)	$2^4 = 16$

#### Assignment Operators

Operator	Description	Example
=	Assign	$a = 50$

#### Comparison Operators

Operator	Description	Example
=	Is equal to	$10 = 15 \Rightarrow$ Result is false
<>	Is not equal to	$7 <> 4 \Rightarrow$ Result is true
>	Greater than	$9 > 7 \Rightarrow$ Result is true

>=	Greater than or equal to	12 >= 12 => Result is true
<	Less than	17 < 29 => Result is true
<=	Less than or equal to	18 <= 18 => Result is true

## Logical Operators

Operator	Description
<b>And</b>	Performs a logical conjunction on two expressions (if both expressions evaluate to True, result is True. If either expression evaluates to False, result is False)
<b>Or</b>	Performs a logical disjunction on two expressions (if either or both expressions evaluate to True, result is True).
<b>Not</b>	Performs logical negation on an expression.
<b>Xor</b>	Performs a logical exclusion on two expressions (if one, and only one, of the expressions evaluates to True, result is True. However, if either of the expressions is Null, the result is also Null).

## Concatenation Operators

Operator	Description
<b>&amp;</b>	Concatenate two strings
<b>+</b>	Adds two numbers together

## Conditional Statements

Conditional statements are used to perform different operations based on different conditions. In VBScript there are four conditional statements

- If
- If...Then...Else
- If...Then...Elseif...Else
- Select Case

### If statement:

The **If statement** checks for a condition and if the condition is true it executes the expression.

```
<script type="text/vbscript">
Dim theNumber
theNumber = 7
If theNumber = 7 Then
```

```
        document.write("The number is " & theNumber)
End If
</script>
```

### **If...Then...Else statement:**

Consider an example where you want to execute a set of statements if a condition passes and if the condition fails we want to execute another set of statements.

For example, If a student's mark is greater than 40 then mark him as 'Pass' and print his name. Else mark him as "Fail".

```
<script type="text/vbscript">
Dim theNumber
theNumber = 7
If theNumber = 7 Then
    document.write("The number " & theNumber & "passed the condition ")
Else
    document.write("The number " & theNumber & "did not pass the condition ")
End If
</script>
```

### **If...Then...Elseif...Else statement:**

This is used when you want to execute different set of statements for different conditions.

For example, if a student has an amount of money which is greater than Rs 50 he uses an auto ; if he has an amount between Rs 20 to Rs 50 he uses the share auto; if he has between Rs 5 to Rs 20 he uses the public bus or else he walks to his school.

In the example given below, different messages are given out depending on the color that is specified.

```
<script type="text/vbscript">
Dim theNumber
theColor = "red"
If theColor = "red" Then
    document.write("Apple is " & theColor & " in color ")
Elseif theColor = "orange" Then
    document.write("Orange is " & theNumber & " in color ")
Elseif theColor = "yellow" Then
    document.write("Pineapple is " & theNumber & " in color ")
Else
    document.write("Orange is " & theNumber & " in color ")
End If
</script>
```

## Select Case:

The Select Case is similar to the If...Then...Elseif...Else. The Select Case is preferred in situations where we want to test against a single variable. Also the Select Case checks only for equal (=) value. It can't be used to check for greater than (>) values or lesser than (<) values.

```
<script type="text/vbscript">
  Dim hairColor
  hairColor = "Yellow"
  Select Case hairColor
  Case "Black"
    document.write("The cat is " & hairColor & " in color ")
  Case "Green"
    document.write("The parrot is " & hairColor & " in color ")
  Case "Yellow"
    document.write("The canary is " & hairColor & " in color ")
  Case Else
    document.write("Ah well, whatever...")
  End Select
</script>
```

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a simple program to: a. Add two numbers and display the results b. Compare two strings and display the results
2.	Declare a variable called marks. Check if marks are greater than 50. If they are > 50, Display "Pass" else display "Fail"
3	Accept a variable called schoolHouse. If the schoolhouse is 'Tansen' display 'Green shirt', if it is 'Meera' display 'Red shirt', if it is 'Kabir' display 'Yellow shirt' and if it is not any of these, then display 'Specify the correct school House you belong to'. Try this with both if -elseif - else statements and Select Case statements.

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of arithmetic operators and logical operators.
2. Explain the use of Select Case in VBScript.
3. Explain the condition statements in VBScript.

**Fill in the blanks:**

1. \_\_\_\_\_ is used to append two strings.
2. Select Case is used to compare only the \_\_\_\_\_ case.
3. Select Case is best preferred when we use a \_\_\_\_\_ variable.

## SESSION 24: LOOPS

### RELEVANT KNOWLEDGE

Loops are used to execute the same set of operations repeatedly for a specified number of times. VBScript has four types of looping statements.

- For...Next statement
- For Each...Next statement
- Do...Loop statement
- While...Wend statement

#### For...Next statement:

For...Next specifies a counter with its start and end values. The next statement increases the counter by 1 which is called step.

```
<html>
<body>
<script type="text/vbscript">
For i = 0 To 5
  document.write("The number is " & i & "<br />")
Next
</script>
</body>
</html>
```

← Executes 6 times

Notice that the loop executes for 6 times. Also here, by default the increment value is 1. You can use the step statement followed by the increment size to increment the loop variable to any number. For example,

```
For i = 2 to 10 step 2
```

will increment *i* by 2 for each iteration i.e. the loop iterates for *i* = 2,4,6,8,10

#### For Each...Next statement:

The “for each...next” statement is used when you iterate on an array. A block of code executes for each item in the array.

```
<script type="text/vbscript">
Dim cars(2)
cars(0)="Volvo"
cars(1)="Saab"
cars(2)="BMW"
For Each x In cars
  document.write(x & "<br />")
```

```
Next  
</script>
```

### Do...loop statement:

The do...loop statement is used when you do not know how many number of times the loop gets executed. The do...loop repeats a block of code while a condition is true or until a condition becomes true. The “while” and the “until” keywords are used to execute the loop in the following scenarios

- Check a condition using while and execute the block only if the condition is true
- Execute the loop at least once while the condition is false. Continue to execute the loop if the condition is true.
- Execute the loop until the condition becomes true.
- Execute the loop at least once until the condition becomes true.

Example: execute block only if condition is true:

```
Do while i > 5  
    Some code  
Loop
```

Notice that the code executes only if  $i > 5$ . Instead, If  $i$  is less than or equal to 5, then the code will not execute the loop at all.

Example: execute block **at least once** even if  $i < 5$ :

```
Do  
    Some code  
Loop while i > 5
```

Example: execute block until a counter( $n$ ) is reached (i.e. loop run  $n - 1$  times)

```
Do until i =10  
    Some code  
Loop
```

**Note:** In the above example the loop will execute *until*  $i$  becomes 10. If  $i = 10$  then the code will not execute.

Example: execute block  $n$  times (where  $n =$  counter).

```
Do  
    Some code  
Loop until i = 10
```

Note the code executes for  $i = 10$  also.

### While...Wend statement:

The while...wend statement executes the loop while the condition is true.

```
While i <= 8
    Some code
Wend
```

Notice the above code executes until  $i$  is less than or equal to 8.

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Write a program to: c. Display the numbers from 1 to 100 using for...next loop d. Display all even numbers from 2 to 30 using for...next loop
2.	Create an array with numbers 1 to 10 using for...next loop. Multiply each value of the array by 5 using for each...next loop and display the results.

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of loops in VBScript.
2. Explain the use of while and until statements in do...loop.

Fill in the blanks:

1. \_\_\_\_\_ and \_\_\_\_\_ are used as condition statements in do...loop.
2. Having the condition after the \_\_\_\_\_ statement makes the loop to execute at least once.
3. The last statement of while loop is the \_\_\_\_\_.

## SESSION 25: PROCEDURES AND FUNCTIONS

### RELEVANT KNOWLEDGE

Procedures are blocks of code written for a specific purpose. They can be reused in multiple places. This reduces the number of lines of code in your program. They get executed only when you call them specifically. There is a lot more clarity in your program when you use procedures and functions.

VBScript has two kinds of Procedures

1. Sub Procedures
2. Function Procedures

#### Sub Procedures:

Sub Procedures are series of statements written with Sub and End Sub. Sub procedures do not return any value. You can also pass arguments to a sub procedure.

Syntax	Example
<pre>Sub &lt;sub_name&gt; ()   'some statements End sub</pre>	<pre>&lt;script type="text/vbscript"&gt; Sub printMessage() Dim myMessage myMessage = "Hello World!"</pre>
<p><b>With Arguments</b></p> <pre>Sub &lt;sub_name&gt;(argument 1, argument 2)   'some statements End sub</pre>	<pre>'This will print out myMessage to the visitor document.write(myMessage) End Sub Call printMessage() &lt;/script&gt;</pre>

Notice that the sub procedure starts with Sub printMessage() and ends with End Sub. To call a sub procedure you can use the keyword “call” followed by the sub procedure name. or just type the procedure name.

Call printMessage() }  
printMessage() } Call a sub procedure in both the ways

#### Function Procedures

A function procedure also called a Function is similar to the sub procedure except for the fact that you can return a value using a function. The function starts with the key

word “Function” and ends with “End function”. You can pass any number of arguments to a function.

Syntax	Example
<pre>Function &lt;function_name&gt;()     `some statements End function</pre>	<pre>&lt;script type="text/vbscript"&gt; Dim message Function printMessage(name) Dim myMessage theMessage = "Hello" finalMessage = theMessage &amp; name End function Message = printMessage("Rahul") &lt;/script&gt;</pre>
<p><b>With Arguments</b></p> <pre>Function &lt;function_name&gt;(argument1, argument2)     `some statements End function</pre>	

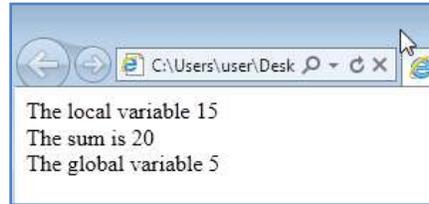
Here, the output will be Hello Rahul. Notice that you are passing a string “Rahul” as an argument to the function printMessage. The variable finalMessage is returned by concatenating the string which is passed and the variable “theMessage”.

### Variable Scoping Revisited

Remember the variable scope earlier. The scope or the lifetime of a variable is very useful when we use functions. You can declare a variable inside a function or outside. If declared inside the function the scope of the variable is within the function and cannot be accessed outside. If declared outside the variable scope is for the entire script.

```
<html>  
<body>  
<script type="text/vbscript">  
Dim a  
a=5  
sub add()  
    dim a  
    dim b  
    a=15  
    b=5  
    document.write("The local variable " & a & "<br>" )  
    document.write("The sum is " & a+b & "<br>" )  
end sub  
add()  
document.write("The global variable " & a & "<br>" )  
</script>  
</body>  
</html>
```

Notice that the variable is declared in two places - one outside the sub and another inside the sub with values 5 and 15 respectively. You can see from the output shown below that the value of variable a inside the sub is 15 and outside the sub is 5. The sum of a and b is 20.



## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Create a program using : a. Sub procedure to add five numbers. Pass the numbers as arguments to the sub procedure. Display the results b. Function to multiply two numbers. Pass the numbers as arguments and display the results
2.	Write a program to understand variable scoping using function

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of procedures.
2. Explain the difference between sub procedure and function procedure.

Fill in the blanks:

1. \_\_\_\_\_ reduces the number of lines in the program.
2. The scope of a variable declared inside a procedure cannot be accessed \_\_\_\_\_ the procedure.
3. \_\_\_\_\_ is a type of procedure which returns a value while a \_\_\_\_\_ procedure does not return a value.

## SESSION 26: OVERVIEW OF IIS & APACHE WEB SERVER

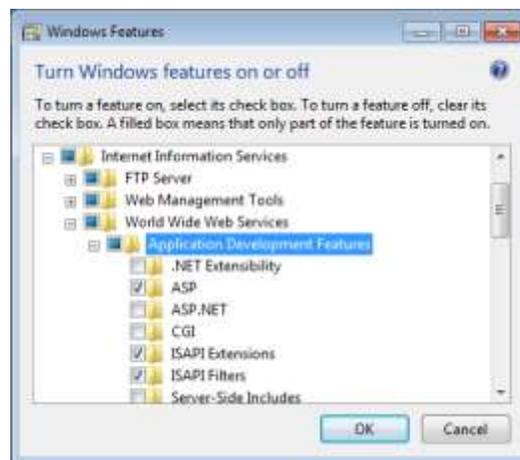
### RELEVANT KNOWLEDGE

#### Internet Information Services

IIS or Internet Information Services is a suite of products for web applications. IIS includes a web server (hence popularly referred to as IIS), ftp server, news server and SMTP service.

IIS is included in both client and server operating systems from Microsoft. IIS shipped with client operating system is meant to be used for testing or internal development purposes. IIS is required for processing server-side scripts included in web pages using ASP. To install IIS, do the following:

1. Select **Start > Control Panel > Programs and Features**.
2. Select **Turn Windows features on or off**.
3. Select and expand **Internet Information Services**, select **ASP** (figure below).



4. Select **OK**.
5. Once it is installed, you can test by typing the URL. For example, type `http://IPADDRESS` and press **Enter**. (Replace IPADDRESS with the IP address of the computer where IIS is installed).

#### Publishing Content

Once you install IIS, it creates the following folder structure `\inetpub\wwwroot` (typically in C:). This folder is used as the starting point of the website hosted on this computer. You can copy or publish web pages into this folder and web pages in this folder are served through the web server.

For example, if you have a web page index.htm or index.html, copy the file in C:\inetpub\wwwroot (should be like C:\inetpub\wwwroot\index.htm). To access this web page through a web server, type http://IPADDRESS/index.htm).

**Note:** You can also create subfolders and place content inside them. For example, C:\inetpub\wwwroot\SITENAME.

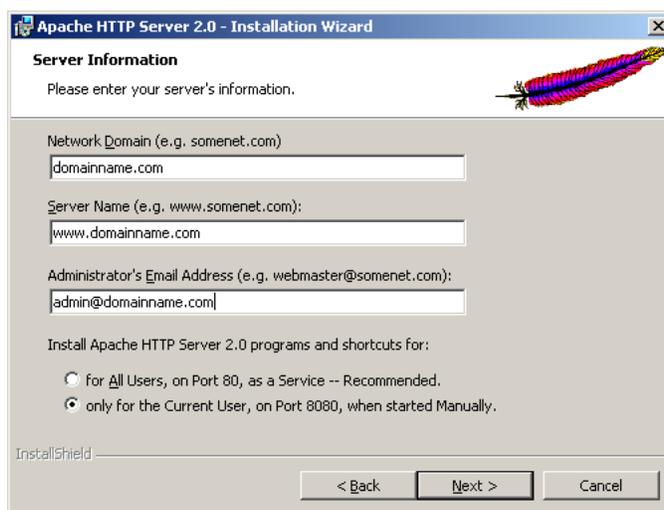
To learn more about IIS, use the following resources:

- [http://en.wikipedia.org/wiki/Internet\\_Information\\_Services](http://en.wikipedia.org/wiki/Internet_Information_Services)
- <http://windows.microsoft.com/en-in/windows-vista/install-internet-information-services-iis-7-0>

## Apache Web Server

Apache Web Server or Apache HTTP Server is the most widely used web server software hosting millions of websites across the globe. Apache web server can be run on Windows though Linux is the most popular operating system used. To use, Apache Web Server, do the following:

1. Go to <https://httpd.apache.org/>.
2. Download Apache HTTP Server.
3. Install Apache HTTP Server by:
  - a. Double-click *httpd-2.0.64-win32-x86-no\_ssl.msi*.
  - b. Click **Next**.
  - c. Select **I accept the terms in the license agreement**.
  - d. Click **Next** twice. Use the following details on Server Information (figure below).



The screenshot shows the 'Apache HTTP Server 2.0 - Installation Wizard' window. The title bar includes the text 'Apache HTTP Server 2.0 - Installation Wizard'. The main window has a header 'Server Information' and a sub-header 'Please enter your server's information.' Below this, there are three text input fields: 'Network Domain (e.g. somenet.com)' with 'domainname.com' entered, 'Server Name (e.g. www.somenet.com):' with 'www.domainname.com' entered, and 'Administrator's Email Address (e.g. webmaster@somenet.com):' with 'admin@domainname.com' entered. At the bottom, there are two radio button options: 'for All Users, on Port 80, as a Service -- Recommended.' (which is unselected) and 'only for the Current User, on Port 8080, when started Manually.' (which is selected). At the very bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'. A small feather icon is visible in the top right corner of the dialog box.

- i. Type a domain name, for example, example.com.

- ii. Type the server name, for example, www.example.com.
  - iii. Type the email address, for example, admin@example.com.
  - iv. Select for **All users, on Port 80, as a service - Recommended**.
  - e. Click **Next**.
  - f. Select **Custom**.
  - g. Select **Next**.
  - h. Select **Install**.
  - i. Select **Finish**.
4. You need to test the web server once it is installed.
  5. Open the web browser and type http://IPADDRESS. For example: http://192.168.2.2. Default home page should be displayed (figure below).



## Publishing Content

Once it is installed, you can view or modify the path to default content folder. Configuration for Apache HTTP Server is stored in httpd.conf located in C:\Program Files\Apache Group\Apache2\conf folder. Open the httpd.conf using a text editor such as notepad and locate “DocumentRoot” section to view the default content folder.

Publish the files to the default content folder and view through the web browser (use http://IPADDRESS).

To learn about Apache HTTP Server, use the following resources:

- [http://en.wikipedia.org/wiki/Apache\\_HTTP\\_Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server)
- <http://www.apache.org/>

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Install IIS and publish sample content. View using a web server (http).
2.	Install Apache Web Server. Publish and view sample content (http).

## ASSESSMENT

Answer the following questions:

1. Explain the procedure to install IIS.
2. Explain the procedure to install Apache HTTP Server.
3. Explain the procedure to publish content on web servers.

Fill in the blanks:

1. Acronym for IIS \_\_\_\_\_.
2. IIS can be installed through \_\_\_\_\_.
3. Apache Web Server configuration file \_\_\_\_\_.

## SESSION 27: OVERVIEW OF ASP

### RELEVANT KNOWLEDGE

Active Server Pages or ASP is a server side scripting technology introduced by Microsoft. Learning ASP is easy as the technology is based on VBScript. ASP web pages contain HTML, XML or Scripts and have the file extension .asp.

When a request is sent from the browser, an ASP page is passed to the ASP engine of the web server such as IIS (Internet Information Services) and once processed, the result is returned to the browser for display. Unlike HTML that displays only static content (content that does not change unless modified by the developer), ASP allows you to add, edit or change content dynamically on the web page. Also ASP code is not displayed in the browser thus making it secure.

VBScript is the *default scripting language* for ASP and is the most preferred language for developers who work with ASP. You can also JScript, Microsoft's version of JavaScript or PerlScript for developing Active Server Pages.

### Writing your first ASP Code

You can use a simple text editor such as notepad to create ASP code.

To create the code, do the following:

- a. Open the text editor and type the following code:

```
<!DOCTYPE html>
<html>
<body>
<%
response.write("My first ASP program")
%>
</body>
</html>
```

- b. Save this as ASP file. For example, "intro.asp".
- c. Now open the web browser and type the complete URL. For example: <http://192.168.2.2/intro.asp>. Notice the results displayed (figure below).



d. Right-click, select View Source. Notice the code displayed.

In this case, the web server has already processed and returned only what is required for the browser to display. Thus, for any server-side scripting, the code is executed at the server's end and browsers just need to display only the result (no support is required at browser's end for ASP!).

Today ASP is used in most websites that require dynamic web pages such as search engines, online shopping portals, recruitment portals, etc. Though you had a high-level overview of ASP, use the following resources for learning ASP:

- [https://en.wikipedia.org/wiki/Active\\_Server\\_Pages](https://en.wikipedia.org/wiki/Active_Server_Pages)
- <http://www.w3schools.com/asp/>
- <http://msdn.microsoft.com/en-us/library/aa286483.aspx>

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Discuss the benefits of ASP. Use the resources included in this session.

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of ASP.

Fill in the blanks:

1. Acronym for ASP \_\_\_\_\_.

## SESSION 28: OVERVIEW OF PHP

### RELEVANT KNOWLEDGE

Hypertext Preprocessor (originally known as Personal Home Page) or PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is one of the most widely used languages on the web today; this is due to the fact that PHP is considered as a free, open-source alternative to competitive technology such as ASP (Active Server Pages). PHP resembles the syntax of C programming language but made simpler and easier than C.

PHP runs on many platforms including Windows, MAC, Unix, etc. PHP is compatible with web servers such as IIS, Apache, etc. PHP supports using a variety of databases such as MySQL, Microsoft SQL, Oracle, Microsoft Access, ODBC compliant databases, etc. however, you find that MySQL as the most commonly used database in most websites. Like any other Server-side scripting language, requests are processed by the web server and only the results are sent to the browser for display.

#### Overview of WAMP

WAMP refers to Windows, Apache Web Server, MySQL and PHP programs packaged together for ease of installation. Instead of going through strenuous procedure for installing and configuring each component individually, you can use WAMP to simply the whole task. To use WAMP, do the following:

1. Go to [www.wampserver.com](http://www.wampserver.com)
2. You will find a single package with a file name like `wampserver2.2e-php5.3.13-httpd2.2.22-mysql5.5.24-32b.exe`.
3. Download WAMP.
4. Follow the on screen instructions to complete the installation.
5. Select **Start > Programs > WampServer** (Notice the icon in the system tray)

#### Writing your first PHP Code

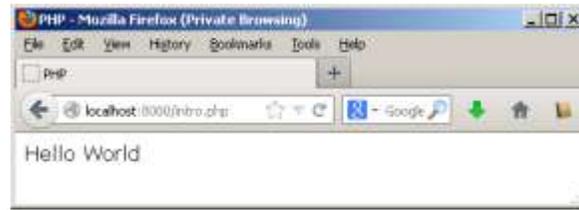
You can use a simple text editor such as notepad to create PHP code.

To create the code, do the following:

- a. Open the text editor and type the following code:

```
<!DOCTYPE html>
<Title>PHP</Title>
<?php
echo 'Hello World';
?>
```

- b. Save this as PHP file (store in root folder). For example, “intro.php”.
- c. Now open the web browser and type the complete URL. For example: <http://192.168.2.2/intro.php>. Notice the results displayed.



Though you had a high-level overview of PHP, use the following resources for learning PHP:

- <http://en.wikipedia.org/wiki/PHP>
- <http://www.w3schools.com/php/>
- <http://in.php.net/>

## EXERCISE

Perform the following activities till you are confident:

S.No.	Activities
1.	Discuss the benefits of PHP. Use the resources included in this session.

## ASSESSMENT

Answer the following questions:

1. Explain the purpose of PHP.

Fill in the blanks:

1. Acronym for PHP \_\_\_\_\_.

## LIST OF CONTRIBUTORS

### **Advisors:**

1. Mr. Ajay Mohan Goel, Director - Skills College, Wadhvani Foundation.
2. Mr. Austin Thomas, Director - Skills College Initiative, Wadhvani Foundation.
3. Prof. R.B. Shivagunde, Joint Director, PSSCIVE, Bhopal.
4. Dr. Vinay Swarup Mehrotra, Head, Curriculum Development and Evaluation Centre, PSSCIVE, Bhopal.

### **Subject Matter Experts:**

1. Ms. Sonia Kakkar, Wadhvani Foundation.
2. Mr. Karthik Chandru, Wadhvani Foundation.
3. Ms. Toral Veecumsee, Wadhvani Foundation.
4. Mr. Ajay Goel, Wadhvani Foundation.
5. Mr. Austin Thomas, Wadhvani Foundation.

### **Editing:**

1. Ms. Sonia Kakkar, Wadhvani Foundation.
2. Dr. Vinay Swarup Mehrotra, Head, Curriculum Development and Evaluation Centre, PSSCIVE, Bhopal.

### **Coordination:**

1. Ms. Rekha Menon, Wadhvani Foundation.
2. Mr. Ajay Goel, Wadhvani Foundation.
3. Mr. Austin Thomas, Wadhvani Foundation.