D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 1

S/W Requirement – any operating System.

Title: Design and develop web application using HTML.

HTML –

- HTML stands for Hyper Text Markup Language.
- It is used to create the web pages.
- It is having pre-defined tags.
- It saves with either .htm or .html file extension.

HTML tags –

Tag name	**Description**
<!-- -->	This tag is used to apply comment in an HTML document.
<!DOCTYPE>	This tag is used to specify the version of HTML
<a>	It is termed as anchor tag and it creates a hyperlink or link.
<abbr>	It defines an abbreviation for a phrase or longer word.
<acronym>	It defines acronym for a word. **(Not supported in HTML5)**
<address>	It defines the author's contact information of the HTML article
<applet>	It defines an embedded Java applet. **(Not supported in HTML5)**
<area>	It defines the area of an image map.

<article>	It defines the self-contained content.
<aside>	It defines content aside from main content. Mainly represented as sidebar.
<audio>	It is used to embed sound content in HTML document.
<b>	It is used to make a text bold.
<base>	This tag defines the base URL for all relative URL within the document.
<basefont>	This tag is used to set default font, size and color for all elements of document. **(Not supported in HTML5)**
<bdi>	This tag is used to provide isolation for that part of text which may be formatted in different directions from its surrounding text.
<bdo>	It is used to override the current text direction.
<big>	This tag is used to make font size one level larger than its surrounding content. **(Not supported in HTML5)**
<blockquote>	It is used to define a content which is taken from another source.
<body>	It is used to define the body section of an HTML document.
<br>	It is used to apply single line break.
<button>	It is used to represent a clickable button
<canvas>	It is used to provide a graphics space within a web document.
<caption>	It is used to define a caption for a table.
<center>	It is used to align the content in center. **(Not supported in HTML5)**

<cite>	It is used to define the title of the work, book, website, etc.
<code>	It is used to display a part of programming code in an HTML document.
<col>	It defines a column within a table which represent common properties of columns and used with the <colgroup> element.
<colgroup>	It is used to define group of columns in a table.
<data>	It is used to link the content with the machine-readable translation.
<datalist>	It is used to provide a predefined list for input option.
<dd>	It is used to provide definition/description of a term in description list.
<del>	It defines a text which has been deleted from the document.
<details>	It defines additional details which user can either view or hide.
<dfn>	It is used to indicate a term which is defined within a sentence/phrase.
<dialog>	It defines a dialog box or other interactive components.
<dir>	It is used as container for directory list of files. **(Not supported in HTML5)**
<div>	It defines a division or section within HTML document.
<dl>	It is sued to define a description list.
<dt>	It is used to define a term in description list.
<em>	It is used to emphasis the content applied within this element.
<embed>	It is used as embedded container for external file/application/media, etc.

<fieldset>	It is used to group related elements/labels within a web form.
<figcaption>	It is used to add a caption or explanation for the <figure> element.
<figure>	It is used to define the self-contained content, and s mostly refer as single unit.
<font>	It defines the font, size, color, and face for the content. **(Not supported in HTML5)**
<footer>	It defines the footer section of a webpage.
<form>	It is used to define an HTML form.
<frame>	It defines a particular area of webpage which can contain another HTML file. **(Not supported in HTML5)**
<frameset>	It defines group of Frames. **(Not supported in HTML5)**
<h1> to <h6>	It defines headings for an HTML document from level 1 to level 6.
<head>	It defines the head section of an HTML document.
<header>	It defines the header of a section or webpage.
<hr>	It is used to apply thematic break between paragraph-level elements.
<html>	It represents root of an HTML document.
<i>	It is used to represent a text in some different voice.
<iframe>	It defines an inline frame which can embed other content.
<img>	It is used to insert an image within an HTML document.

<input>	It defines an input field within an HTML form.
<ins>	It represent text that has been inserted within an HTML document.
<isindex>	It is used to display search string for current document. **(Not supported in HTML5)**
<kbd>	It is used to define keyboard input.
<label>	It defines a text label for the input field of form.
<legend>	It defines a caption for content of <fieldset>
<li>	It is used to represent items in list.
<link>	It represents a relationship between current document and an external resource.
<main>	It represents the main content of an HTML document.
<map>	It defines an image map with active areas.
<mark>	It represents a highlighted text.
<marquee>	It is used to insert the scrolling text or an image either horizontally or vertically. **(Not supported in HTML5)**
<menu>	It is used for creating a menu list of commands.
<meta>	It defines metadata of an HTML document.
<meter>	It defines scalar measurement with known range or fractional value.
<nav>	It represents section of page to represent navigation links.

<noframes>	It provides alternate content to represent in browser which does not support the <frame> elements. **(Not supported in HTML5)**
<noscript>	It provides an alternative content if a script type is not supported in browser.
<object>	It is used to embed an object in HTML file.
<ol>	It defines an ordered list of items.
<optgroup>	It is used to group the options of a drop-down list.
<option>	It is used to define options or items in a drop-down list.
<output>	It is used as container element which can show result of a calculation.
<p>	It represents a paragraph in an HTML document.
<param>	It defines parameter for an <object> element
<picture>	It defines more than one source element and one image element.
<pre>	It defines preformatted text in an HTML document.
<progress>	It defines the progress of a task within HTML document.
<q>	It defines short inline quotation.
<rp>	It defines an alternative content if browser does not supports ruby annotations.
<rt>	It defines explanations and pronunciations in ruby annotations.
<ruby>	It is used to represent ruby annotations.
<s>	It render text which is no longer correct or relevant.

<samp>	It is used to represent sample output of a computer program.
<script>	It is used to declare the JavaScript within HTML document.
<section>	It defines a generic section for a document.
<select>	It represents a control which provides a menu of options.
<small>	It is used to make text font one size smaller than document?s base font size.
<source>>	It defines multiple media recourses for different media element such as <picture>, <video>, and <audio> element.
<span>	It is used for styling and grouping inline.
<strike>	It is used to render strike through the text. **(Not supported in HTML5)**
<strong>	It is used to define important text.
<style>	It is used to contain style information for an HTML document.
<sub>	It defines a text which displays as a subscript text.
<summary>	It defines summary which can be used with <details> tag.
<sup>	It defines a text which represent as superscript text.
<svg>	It is used as container of SVG (Scalable Vector Graphics).
<table>	It is used to present data in tabular form or to create a table within HTML document.
<tbody>	It represents the body content of an HTML table and used along with <thead> and <tfoot>.

<td>	It is used to define cells of an HTML table which contains table data
<template>	It is used to contain the client side content which will not display at time of page load and may render later using JavaScript.
<textarea>	It is used to define multiple line input, such as comment, feedback, and review, etc.
<tfoot>	It defines the footer content of an HTML table.
<th>	It defines the head cell of an HTML table.
<thead>	It defines the header of an HTML table. It is used along with <tbody> and <tfoot> tags.
<time>	It is used to define data/time within an HTML document.
<title>	It defines the title or name of an HTML document.
<tr>	It defines the row cells in an HTML table
<track>	It is used to define text tracks for <audio> and <video> elements.
<tt>	It is used to define teletype text. **(Not supported in HTML5)**
<u>	It is used to render enclosed text with an underline.
<ul>	It defines unordered list of items.
<var>	It defines variable name used in mathematical or programming context.
<video>	It is used to embed a video content with an HTML document
<wbr>	It defines a position within text where break line is possible.

Example –

Develop a login application

```
<html>

<form name="f1"method="post" action="login.php" id="f1">
<table>
<tr>
<td class="f1_label">User Name:</td><td><input type="text" name="username" value=""/>
</td>
</tr>
<tr>
<td class="f1_label">Password:</td><td>
<input type="password" name="password" value=""/>
</td>
</tr>
<tr>
<td>
 <input type="submit" name="login" value="Log In" style="font-size:18px;"/>
</td>
</tr>
</table>
</form>
</html>
```

Results –

1. Students should be able to develop static web pages for website by using HTML.

Observations –

1. Student should able to modify the code by using the 'placeholder' and 'required' attributes

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory	**Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 2

S/W Requirement – any operating System.

Title: Develop an application for demonstrating the use of CSS in HTML.

CSS –

Cascading Style Sheet (CSS) is used to set the style in web pages that contain HTML elements. It sets the background color, font-size, font-family, color, … etc property of elements on a web page.

There are three types of CSS which are given below:

- Inline CSS
- Internal or Embedded CSS
- External CSS

Inline CSS:

Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.

Example:

- html

```
<!DOCTYPE html>
<html>
   <head>
      <title>Inline CSS</title>
   </head>
  <body>
      <p style = "color:#009900; font-size:50px;
            font-style:italic; text-align:center;">
         GeeksForGeeks
      </p>
```

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

Output:

GeeksForGeeks

Internal or Embedded CSS:

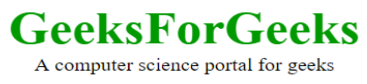
This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.

Example:

- html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Internal CSS</title>
        <style>
            .main {
                text-align:center;
            }
            .GFG {
                color:#009900;
                font-size:50px;
                font-weight:bold;
            }
            .geeks {
                font-style:bold;
                font-size:20px;
            }
        </style>
    </head>
    <body>
        <div class = "main">
            <div class ="GFG">GeeksForGeeks</div>
              <div class ="geeks">
            A computer science portal for geeks
          </div>
        </div>
      </body>
    </html>
```

Output:



GeeksForGeeks

A computer science portal for geeks

External CSS:

External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using **link** tag. This means that for each element, style can be set only once and that will be applied across web pages. **Example:** The file given below contains CSS property. This file save with .css extension. For Ex: **geeks.css**

```
body {
   background-color:powderblue;
}
.main {
   text-align:center;
}
.GFG {
   color:#009900;
   font-size:50px;
   font-weight:bold;
}
#geeks {
   font-style:bold;
   font-size:20px;
```

}

Below is the HTML file that is making use of the created external style sheet

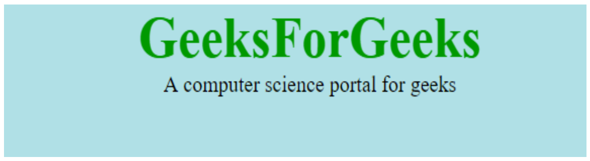
- **link** tag is used to link the external style sheet with the html webpage.
- **href** attribute is used to specify the location of the external style sheet file.

- html

```
<!DOCTYPE html>
<html>
   <head>
      <link rel="stylesheet" href="geeks.css"/>
   </head>

   <body>
      <div class = "main">
         <div class ="GFG">GeeksForGeeks</div>
         <div id ="geeks">
            A computer science portal for geeks
         </div>
      </div>
   </body>
</html>
```

Output:



Properties of CSS: Inline CSS has the highest priority, then comes Internal/Embedded followed by External CSS which has the least priority. Multiple style sheets can be defined on one page. If for an HTML tag, styles are defined in multiple style sheets then the below order will be followed.

- As Inline has the highest priority, any styles that are defined in the internal and external style sheets are overridden by Inline styles.

- Internal or Embedded stands second in the priority list and overrides the styles in the external style sheet.
- External style sheets have the least priority. If there are no styles defined either in inline or internal style sheet then external style sheet rules are applied for the HTML tags.

Results:

1. Students should develop static website by using CSS in HTML

Observations –

1. Student should able to modify the code by using various different property and value tags.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory	**Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 3

Title - Study and implementation of XML.

XML

- Xml (eXtensible Markup Language) is a mark up language.
- XML is designed to store and transport data.
- Xml was released in late 90's. it was created to provide an easy to use and store self describing data.
- XML became a W3C Recommendation on February 10, 1998.
- XML is not a replacement for HTML.
- XML is designed to be self-descriptive.
- XML is designed to carry data, not to display data.
- XML tags are not predefined. You must define your own tags.
- XML is platform independent and language independent.

Features and Advantages of XML

XML is widely used in the era of web development. It is also used to simplify data storage and data sharing.

The main features or advantages of XML are given below.

1) XML separates data from HTML

If you need to display dynamic data in your HTML document, it will take a lot of work to edit the HTML each time the data changes. With XML, data can be stored in separate XML files. This way you can focus on using HTML/CSS for display and layout, and be sure that changes in the underlying data will not require any changes to the HTML. With a few lines of JavaScript code, you can read an external XML file and update the data content of your web page.

2) XML simplifies data sharing

In the real world, computer systems and databases contain data in incompatible formats. XML data is stored in plain text format. This provides a software- and hardware-independent way of storing data. This makes it much easier to create data that can be shared by different applications.

3) XML simplifies data transport

One of the most time-consuming challenges for developers is to exchange data between incompatible systems over the Internet. Exchanging data as XML greatly reduces this complexity, since the data can be read by different incompatible applications.

4) XML simplifies Platform change

Upgrading to new systems (hardware or software platforms), is always time consuming. Large amounts of data must be converted and incompatible data is often lost. XML data is stored in text format. This makes it easier to expand or upgrade to new operating systems, new applications, or new browsers, without losing data.

5) XML increases data availability

Different applications can access your data, not only in HTML pages, but also from XML data sources. With XML, your data can be available to all kinds of "reading machines" (Handheld computers, voice machines, news feeds, etc), and make it more available for blind people, or people with other disabilities.

6) XML can be used to create new internet languages

A lot of new Internet languages are created with XML.

Here are some examples:

- **XHTML**
- **WSDL** for describing available web services
- **WAP** and **WML** as markup languages for handheld devices
- **RSS** languages for news feeds
- **RDF** and **OWL** for describing resources and ontology
- **SMIL** for describing multimedia for the web

HTML vs XML

There are many differences between HTML (Hyper Text Markup Language) and XML (eXtensible Markup Language). The important differences are given below:

No.	HTML	XML
1)	HTML is used to display data and focuses on how data looks.	XML is a software and hardware independent tool used to transport and store data. It focuses on what data is.
2)	HTML is a markup language itself.	XML provides a framework to define markup languages.
3)	HTML is not case sensitive.	XML is case sensitive.
4)	HTML is a presentation language.	XML is neither a presentation language nor a programming language.
5)	HTML has its own predefined tags.	You can define tags according to your need.
6)	In HTML, it is not necessary to use a closing tag.	XML makes it mandatory to use a closing tag.

7)	HTML is static because it is used to display data.	XML is dynamic because it is used to transport data.
8)	HTML does not preserve whitespaces.	XML preserve whitespaces.

For example -

```
<?xml version="1.0" encoding="UTF-8"?>
<message>
<to>Students</to>
<from>Teacher</from>
<subject>Regarding assignment submission</subject>
<text>All students will have to submit assignment by tomorrow.</text>
</message>
```

Results:

1. Students should able to create the XML structure

Observations –

1. Student should able to create the XML document using elements and attributes of XML.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 4

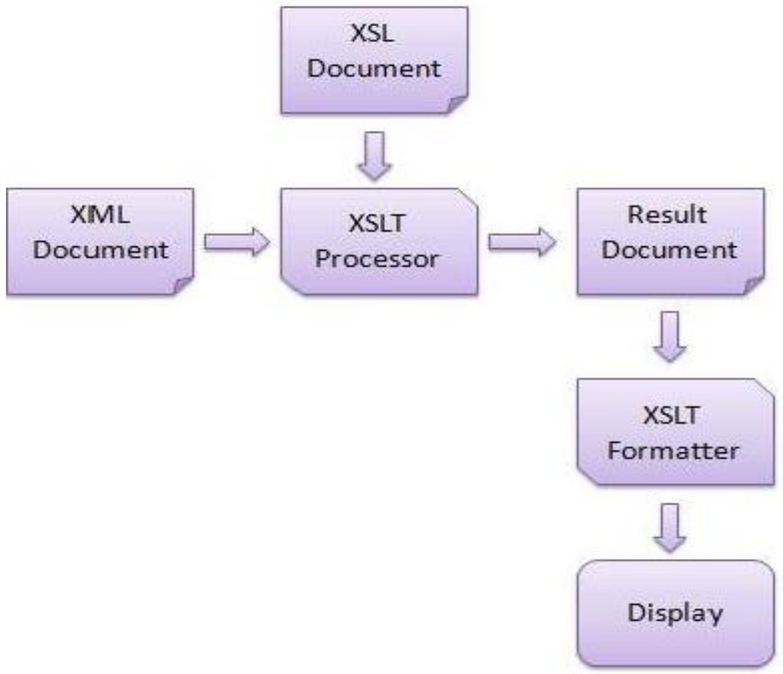
Title - Implementation of student application using XSLT and conditional statement.

XSLT –

XSLT stands for XSL Transformation. It is used to transform XML documents into other formats (like transforming XML into HTML).

The XSLT stylesheet is written in XML format. It is used to define the transformation rules to be applied on the target XML document. The XSLT processor takes the XSLT stylesheet and applies the transformation rules on the target XML document and then it generates a formatted document in the form of XML, HTML, or text format. At the end it is used by XSLT formatter to generate the actual output and displayed on the end-user.

Image representation:



Advantage of XSLT

- XSLT provides an easy way to merge XML data into presentation because it applies user defined transformations to an XML document and the output can be HTML, XML, or any other structured document.
- XSLT provides Xpath to locate elements/attribute within an XML document. So it is more convenient way to traverse an XML document rather than a traditional way, by using scripting language.
- XSLT is template based. So it is more resilient to changes in documents than low level DOM and SAX.
- By using XML and XSLT, the application UI script will look clean and will be easier to maintain.
- XSLT templates are based on XPath pattern which is very powerful in terms of performance to process the XML document.
- XSLT can be used as a validation language as it uses tree-pattern-matching approach.
- You can change the output simply modifying the transformations in XSL files.

For Example –

Employee.xml

```
<?xml version = "1.0"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
```

```
      <nickname>Zoya</nickname>
      <salary>25000</salary>
    </employee>
    <employee id = "056">
      <firstname>Peter</firstname>
      <lastname>Symon</lastname>
      <nickname>John</nickname>
      <salary>10000</salary>
    </employee>
  </class>
```

Employee.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!-- xsl stylesheet declaration with xsl namespace:
Namespace tells the xlst processor about which
element is to be processed and which is used for output purpose only
-->
<xsl:stylesheet version = "1.0"
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
<!-- xsl template declaration:
template tells the xlst processor about the section of xml
document which is to be formatted. It takes an XPath expression.
In our case, it is matching document root element and will
tell processor to process the entire document with this template.
-->
  <xsl:template match = "/">
    <!-- HTML tags
      Used for formatting purpose. Processor will skip them and browser
        will simply render them.
    -->
    <html>
      <body>
        <h2>Employee</h2>
        <table border = "1">
          <tr bgcolor = "#9acd32">
            <th>ID</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Nick Name</th>
            <th>Salary</th>
          </tr>
          <!-- for-each processing instruction
```

```
              Looks for each element matching the XPath expression
              -->
             <xsl:for-each select="class/employee">
                <tr>
                   <td>
                      <!-- value-of processing instruction
                      process the value of the element matching the XPath expression
                      -->
                      <xsl:value-of select = "@id"/>
                   </td>
                   <td><xsl:value-of select = "firstname"/></td>
                   <td><xsl:value-of select = "lastname"/></td>
                   <td><xsl:value-of select = "nickname"/></td>
                   <td><xsl:value-of select = "salary"/></td>
                </tr>
              </xsl:for-each>
           </table>
        </body>
     </html>
  </xsl:template>
</xsl:stylesheet>
```

XSLT <xsl:value-of> Element

The XSLT <xsl:value-of> element is used to extract the value of selected node. It puts the value of selected node as per XPath expression, as text.

```
<xsl:value-of
  select = Expression
  disable-output-escaping = "yes" | "no">
</xsl:value-of>
```

XSLT <xsl:for-each> Element

The XSLT <xsl:for-each> element is used to apply a template repeatedly for each node.

```
<xsl:for-each
  select = Expression>
</xsl:for-each>
```

XSLT <xsl:sort> Element

The XSLT <xsl:sort> element is used to specify a sort criteria on the nodes. It displays the output in sorted form. The <xml:sort> element is added inside the <xsl:for-each> element in the XSL file, to sort the output.

```
<xsl:sort
  select = string-expression
  lang = { nmtoken }
  data-type = { "text" | "number" | QName }
  order = { "ascending" | "descending" }
  case-order = { "upper-first" | "lower-first" } >
</xsl:sort>
```

XSLT <xsl:if> Element

The XSLT <xsl:if> element is used to specify a conditional test against the content of the XML file.

```
<xsl:if test="expression">
 ...some output if the expression is true...
</xsl:if>
```

XSLT <xsl:choose> Element

The XSLT <xsl:choose> element is used to specify a multiple conditional test against the content of nodes with the <xsl:otherwise> and <xsl:when> elements.

```
<xsl:choose>
 <xsl:when test="expression">
  ... some output ...
 </xsl:when>
 <xsl:otherwise>
```

```
     ... some output ....
    </xsl:otherwise>
  </xsl:choose>
```

Program –

Write the XML and XSLT for the student registration.

Results:

1. Students should create XML and apply the XSLT to it

Observations –

1. Student should able to apply the different conditional statements inside the XSLT.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 5

Title – Implementation of XPATH language expression for employee application.

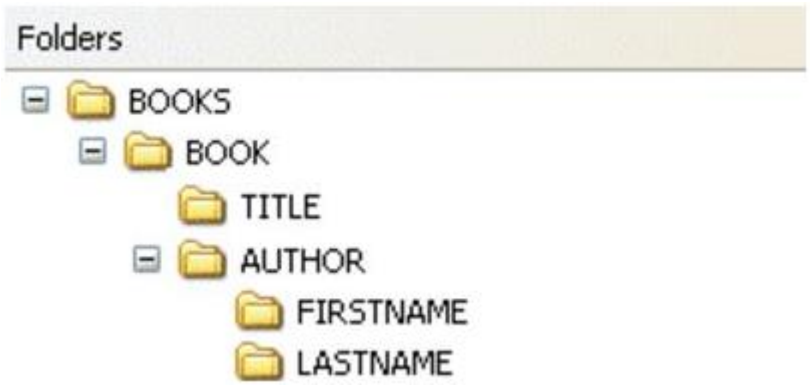
XPath –

XPath is an official recommendation of the World Wide Web Consortium (W3C). It defines a language to find information in an XML file. It is used to traverse elements and attributes of an XML document. XPath provides various types of expressions which can be used to enquire relevant information from the XML document.

- **Structure Definitions** − XPath defines the parts of an XML document like element, attribute, text, namespace, processing-instruction, comment, and document nodes
- **Path Expressions** − XPath provides powerful path expressions select nodes or list of nodes in XML documents.
- **Standard Functions** − XPath provides a rich library of standard functions for manipulation of string values, numeric values, date and time comparison, node and QName manipulation, sequence manipulation, Boolean values etc.
- **Major part of XSLT** − XPath is one of the major elements in XSLT standard and is must have knowledge in order to work with XSLT documents.
- **W3C recommendation** − XPath is an official recommendation of World Wide Web Consortium (W3C).

XPath Expression

XPath defines a pattern or path expression to select nodes or node sets in an XML document. These patterns are used by XSLT to perform transformations. The path expressions look like very similar to the general expressions we used in traditional file system.



XPath specifies seven types of nodes that can be output of the execution of the XPath expression.

- Root
- Element
- Text
- Attribute
- Comment
- Processing Instruction
- Namespace

We know that XPath uses a path expression to select node or a list of nodes from an XML document. A list of useful paths and expression to select any node/ list of nodes from an XML document:

Index	Expression	Description
1)	node-name	It is used to select all nodes with the given name "nodename"
2)	/	It specifies that selection starts from the root node.

3)	//	It specifies that selection starts from the current node that match the selection.
4)	.	Select the current node.
5)	..	Select the parent of the current node.
6)	@	Selects attributes.
7)	Student	Example - selects all nodes with the name "student".
8)	class/student	Example - selects all student elements that are children of class
9)	//student	Selects all student elements no matter where they are in the document

---

XPath Expression Example

Let's take an example to see the usage of XPath expression. Here, we use an xml file "employee.xml" and a stylesheet for that xml file named "employee.xsl". The XSL file uses the XPath expressions under **select** attribute of various XSL tags to fetchvalues of id, firstname, lastname, nickname andsalary of each employee node.

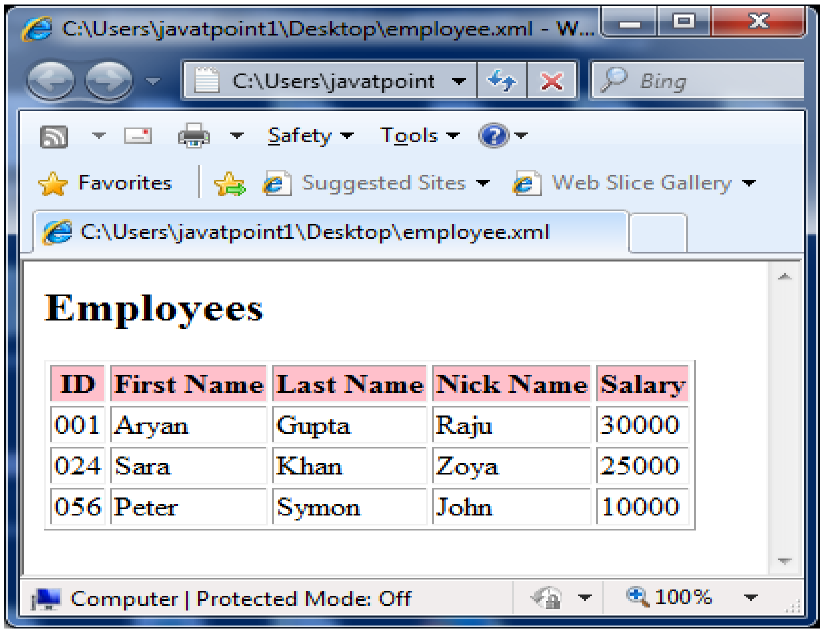
Employee.xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee.xsl"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
    <nickname>Zoya</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "056">
    <firstname>Peter</firstname>
    <lastname>Symon</lastname>
    <nickname>John</nickname>
    <salary>10000</salary>
  </employee>
</class>
```

Employee.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xsl:stylesheet version = "1.0">
  xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
        <h2> Employees</h2>
        <table border = "1>
          <tr bgcolor = "pink">
            <th> ID</th>
            <th> First Name</th>
            <th> Last Name</th>
            <th> Nick Name</th>
            <th> Salary</th>
          </tr>
          <xsl:for-each select = "class/employee">
            <tr>
              <td> <xsl:value-of select = "@id"/> </td>
              <td> <xsl:value-of select = "firstname"/> </td>
              <td> <xsl:value-of select = "lastname"/> </td>
              <td> <xsl:value-of select = "nickname"/> </td>
              <td> <xsl:value-of select = "salary"/> </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Output:


C:\Users\javatpoint1\Desktop\employee.xml - W...
C:\Users\javatpoint
Bing
Safety
Tools
Favorites
Suggested Sites
Web Slice Gallery
C:\Users\javatpoint1\Desktop\employee.xml
Computer | Protected Mode: Off
100%

Employees

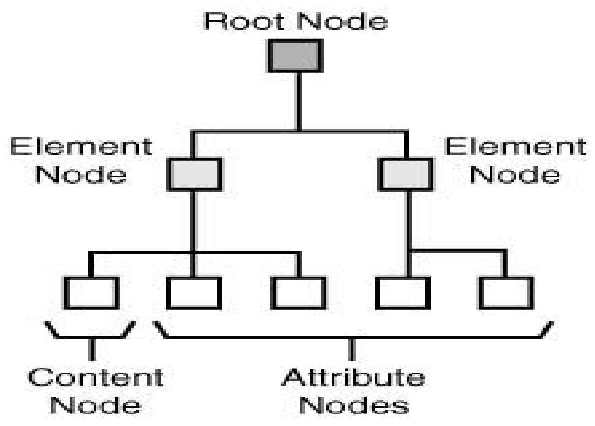
ID	First Name	Last Name	Nick Name	Salary
001	Aryan	Gupta	Raju	30000
024	Sara	Khan	Zoya	25000
056	Peter	Symon	John	10000

XPath Nodes

There are seven kinds of nodes in XPath:

1. Element
2. Attribute
3. Text
4. Namespace
5. Processing-instruction
6. Comment
7. Document nodes.

An XML document can be specified as a tree of nodes. The topmost element of the tree is called the root element.



Let's take an example of XML document to understand the different terminology of XPath nodes.

An XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<Library>
 <book>
  <title lang="en">Three Mistakes of My Life</title>
  <author>Chetan Bhagat</author>
  <year>2008</year>
  <price>110</price>
 </book>
</Library>
```

Nodes in the above XML document:

```
<library> (root element node)
<author>Chetan Bhagat</author> (element node)
lang="en" (attribute node)
```

Atomic values

Atomic values are used to specify the nodes with no children or parent. For example: In the above XML document, following are the atomic values:

Chetan Bhagat

"en"

Relationship of Nodes

Parent Node

Each element and attribute has a parent which is a top element of the respective element or attribute.

See this example:

In this example, the book element is the parent of the title, author, year, and price.

```
<book>
   <title lang="en">Three Mistakes of My Life</title>
   <author>Chetan Bhagat</author>
   <year>2008</year>
   <price>110</price>
</book>
```

Children Nodes

The children nodes can have zero, one or more children. In this example, the title, author, year, and price elements are all children of the book element.

```
<book>
   <title lang="en">Three Mistakes of My Life</title>
   <author>Chetan Bhagat</author>
   <year>2008</year>
   <price>110</price>
</book>
```

Siblings Nodes

The nodes having the same parent are known as siblings. In this example, the title, author, year, and price elements are all siblings.

```
<book>
   <title lang="en">Three Mistakes of My Life</title>
   <author>Chetan Bhagat</author>
```

```
    <year>2008</year>
    <price>110</price>
  </book>
```

Ancestors

A node's parent or parent's parent is specified as ancestor. In this example, the ancestors of the title element are the book element and the library element.

```
<Library>
  <book>
    <title lang="en">Three Mistakes of My Life</title>
    <author>Chetan Bhagat</author>
    <year>2008</year>
    <price>110</price>
  </book>
</Library>
```

Descendants

A descendent is specified as a node's children or children's children. In this example, descendants of the library element are the book, title, author, year, and price elements.

```
<Library>
  <book>
    <title lang="en">Three Mistakes of My Life</title>
    <author>Chetan Bhagat</author>
    <year>2008</year>
    <price>110</price>
  </book>
</Library>
```

XPath Syntax

The XPath expression uses a path notation like URLs, for addressing parts of an XML document. The expression is evaluated to yield an object of the node-set, Boolean, number, or string type. For example, the expression book/author will return a node-set of the <author> elements contained in the <book> elements, if such elements are declared in the source XML document.

In XPath, path expression is used to select nodes or node-sets in an XML document. The node is selected by following a path or steps.

Let's take an example to see the syntax of XPath. Here, we take an XML document.

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book>
  <title lang="en">Three Mistakes of My Life</title>
  <price>110</price>
</book>
<book>
  <title lang="en">Immortals of Meluha</title>
  <price>200</price>
</book>
</bookstore>
```

Selecting Nodes

Path expressions used for selecting nodes are:

Index	Expression	Description
1)	nodename	Selects all nodes with the name "nodename"
2)	/	Selects from the root node.
3)	//	Selects nodes in the document from the current node that match the selection no matter where they are.
4)	.	Selects the current node
5)	..	Selects the parent of the current node
6)	@	Selects attributes

See the path expressions and their details in the above example:

Path Expression	**Result**
bookstore	Selects all nodes with the name "bookstore"
/bookstore	Selects the root element bookstore. **Note:** if the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore.
//book	Selects all book elements no matter where they are in the document.
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element.
//@lang	Selects all attributes that are named lang.

Predicates

Predicates are used to find a specific node or a node that contains a specific value.

Predicates are always embedded in square brackets.

In the table below we have listed some path expressions with predicates and the result of the expressions:

Path Expression	**Result**
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element. **Note:** In IE 5,6,7,8,9 first node is[0], but according to

	W3C, it is [1]. To solve this problem in IE, set the selectionlanguage to XPath: in JavaScript: xml.setProperty("SelectionLanguage","XPath");
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element.
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element.
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element.
//title[@lang]	Selects all the title elements that have an attribute named lang.
//title[@lang='en']	Selects all the title elements that have a "lang" attribute with a value of "en".
/bookstore/book[price>100]	Selects all the book elements of the bookstore element that have a price element with a value greater than 100
/bookstore/book[price>100]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 100

Selecting Unknown Nodes

XPath wildcards are used to select unknown XML nodes.

Wildcard	Description
*	Matches any element node
@*	Matches any attribute node

node()	Matches any node of any kind

See this in above example:

Path Expression	Result
/bookstore/*	Selects all the child element nodes of the bookstore element
//*	Selects all elements in the document
//title[@*]	Selects all title elements which have at least one attribute of any kind

---

Selecting Several Paths

The | operator is used in XPath expression to select several paths. From the above example, we have listed some path expressions and result of the expressions.

Path Expression	Result
//book/title | //book/price	Selects all the title and price elements of all book elements
//title | //price	Selects all the title and price elements in the document
/bookstore/book/title | //price	Selects all the title elements of the book element of the bookstore element and all the price elements in the document

Program –

Apply different XPath expression for the XSLT

Results:

1. Students should able to write the different XPath expressions

Observations –

1. Student should able to modify the XPath expressions as per needs

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 6

Title: Write a code for validating the HTML form inputs using JavaScript.

Theory:

JavaScript is a lightweight, interpreted **programming** language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. **JavaScript** is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Why to Learn JavaScript

JavaScript is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning JavaScript:

- JavaScript is the most popular **programming language** in the world and that makes it a programmer's great choice. Once you learnt JavaScript, it helps you developing great front-end as well as back-end software's using different JavaScript based frameworks like jQuery, Node.JS etc.

- JavaScript is everywhere, it comes installed on every modern web browser and so to learn JavaScript you really do not need any special environment setup. For example, Chrome, Mozilla Firefox, Safari and every browser you know as of today, supports JavaScript.

- JavaScript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.

- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as JavaScript Programmer.

- Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills looks like in the job market.
- Great thing about JavaScript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

Applications of JavaScript Programming

JavaScript is one of the most widely used programming languages (Front-end as well as Back-end). It has its presence in almost every area of software development. I'm going to list few of them here:

Client-side validation - This is really important to verify any user input before submitting it to the server and JavaScript plays an important role in validating those inputs at front-end itself.

Manipulating HTML Pages - JavaScript helps in manipulating HTML page on the fly. This helps in adding and deleting any HTML tag very easily using JavaScript and modify your HTML to change its look and feel based on different devices and requirements.

User Notifications - You can use JavaScript to raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors.

Back-end Data Loading - JavaScript provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your website visitors.

Presentations - JavaScript also provides the facility of creating presentations which gives website look and feel. JavaScript provides RevealJS and BespokeJS libraries to build a web-based slide presentation.

Server Applications - Node JS is built on Chrome's JavaScript runtime for building fast and scalable network applications. This is an event-based library which helps in developing very sophisticated server applications including Web Servers.

Example:

```
<!DOCTYPE HTML>
<html>
<head>
<style>
      p {
          position: absolute;
         top: 50%;
         left: 50%;
        transform: translate(-50%, -50%);
        }
</style>
</head>
<body>
<p id="demo"></p>
  <script>
        var var1 = setInterval(inTimer, 1000);
        var fs = 5;
        var ids = document.getElementById("demo");
        function inTimer() {
        ids.innerHTML = 'TEXT GROWING';
       ids.setAttribute('style', "font-size: " + fs + "px; color: red");
       fs += 5;
      if(fs >= 50 ){
      clearInterval(var1);
      var2 = setInterval(deTimer, 1000);
        }
     }

   function deTimer() {
  fs -= 5;
 ids.innerHTML = 'TEXT SHRINKING';
 ids.setAttribute('style', "font-size: " + fs + "px; color: blue");
 if(fs === 5 ){
 clearInterval(var2);
}
}
</script>
</body>
</html>
```

Validating a form: The data entered into a form needs to be in the right format and certain fields need to be filled in order to effectively use the submitted form. Username, password, contact information are some details that are mandatory in forms and thus need to be provided by the user. Below is a code in HTML, CSS, and JavaScript to validate a form.

For example:

```
<html>
 <head>
        <script>
                function executefun() {
                        var name = document.forms.RegForm.Name.value;
                        var email =document.forms.RegForm.EMail.value;
                        var phone = document.forms.RegForm.Telephone.value;
                        var what = document.forms.RegForm.Subject.value;
                        var password = document.forms.RegForm.Password.value;
                        var address =  document.forms.RegForm.Address.value;
                        var regEmail=/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/g;
                                                    //Javascript reGex for Email Validation.
                       var regPhone=/^\d{10}$/;
                                            // Javascript reGex for Phone Number validation.
                        var regName = /\d+$/g;
                                                 // Javascript reGex for Name validation

                                 if (name == "" || regName.test(name)) {
                                        window.alert("Please enter your name properly.");
                                        name.focus();
                                        return false;
                                }

                                if (address == "") {
                                        window.alert("Please enter your address.");
                                        address.focus();
                                        return false;
                                }

                                if (email == "" || !regEmail.test(email)) {
                                        window.alert("Please enter a valid e-mail address.");
                                        email.focus();
                                        return false;
                                }

                                if (password == "") {
                                        alert("Please enter your password");
                                        password.focus();
                                        return false;
                                }

                                if(password.length <6){
                                        alert("Password should be atleast 6 character long");
                                        password.focus();
                                        return false;

                                }
                                if (phone == "" || !regPhone.test(phone)) {
                                        alert("Please enter valid phone number.");
                                        phone.focus();
                                        return false;
                                }

                                if (what.selectedIndex == -1) {
                                        alert("Please enter your course.");
                                        what.focus();
                                        return false;
                                }

                                return true;
                        }
                </script>

                <style>
                        div {
                                box-sizing: border-box;
                                width: 100%;
                                border: 100px solid black;
                                float: left;
                                align-content: center;
                                align-items: center;
                        }

                        form {
                                margin: 0 auto;
                                width: 600px;
                        }
                </style>
        </head>

        <body>
        <h1 style="text-align: center;">REGISTRATION FORM</h1>
        <form name="RegForm" onsubmit="return executefun()" method="post">
```

```
<p>Name: <input type="text" size="65" name="Name" /></p> <br />

<p>Address: <input type="text" size="65" name="Address" /></p> <br />

<p>E-mail Address: <input type="text" size="65" name="EMail" /></p><br />

<p>Password: <input type="text" size="65" name="Password" /></p> <br />

<p>Telephone: <input type="text" size="65" name="Telephone" /></p> <br />

<p>                                    SELECT YOUR COURSE
                                       <select type="text" value="" name="Subject">
                                                <option>BTECH</option>
                                                <option>BBA</option>
                                                <option>BCA</option>
                                                <option>B.COM</option>
                                                <option>GEEKFORGEEKS</option>
                                       </select>
</p><br /> <br />

<p>Comments: <textarea cols="55" name="Comment"> </textarea></p>

<p>  <input type="submit" value="send" name="Submit" />
      <input type="reset" value="Reset" name="Reset" />
</p>

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

Results:

1. Students should develop application using HTML and validate the input by using JavaScript

Observations –

1. Student should able to apply and validate the JavaScript for the static website with different client side events and scripts.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory	**Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 7

Title: Write a JavaScript to design a simple calculator

JavaScript

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

Features of JavaScript

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.

6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

JavaScript Variable

A **JavaScript variable** is simply a name of storage location. There are two types of variables in JavaScript: local variable and global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

1. Name must start with a letter (a to z or A to Z), underscore(_), or dollar($) sign.
2. After first letter we can use digits (0 to 9), for example value1.
3. JavaScript variables are case sensitive, for example x and X are different variables.

```
var x = 10;
var _value="sonoo";
```

JavaScript Data Types

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

JavaScript is a **dynamic type language**; means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use **var** here to specify the data type. It can hold any type of values such as numbers, strings etc. For example:

```
var a=40;//holding number
var b="Rahul";//holding string
```

JavaScript primitive data types

There are five types of primitive data types in JavaScript. They are as follows:

Data Type	Description
String	represents sequence of characters e.g. "hello"

Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

JavaScript non-primitive data types

The non-primitive data types are as follows:

Data Type	**Description**
Object	represents instance through which we can access members
Array	represents group of similar values
RegExp	represents regular expression

JavaScript Operators

JavaScript operators are symbols that are used to perform operations on operands. For example:

```
var sum=10+20;
```

Here, + is the arithmetic operator and = is the assignment operator.

There are following types of operators in JavaScript.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Bitwise Operators
4. Logical Operators
5. Assignment Operators
6. Special Operators

JavaScript If-else

The **JavaScript if-else statement** is used *to execute the code whether condition is true or false*. There are three forms of if statement in JavaScript.

1. If Statement
2. If else statement
3. if else if statement

JavaScript If statement

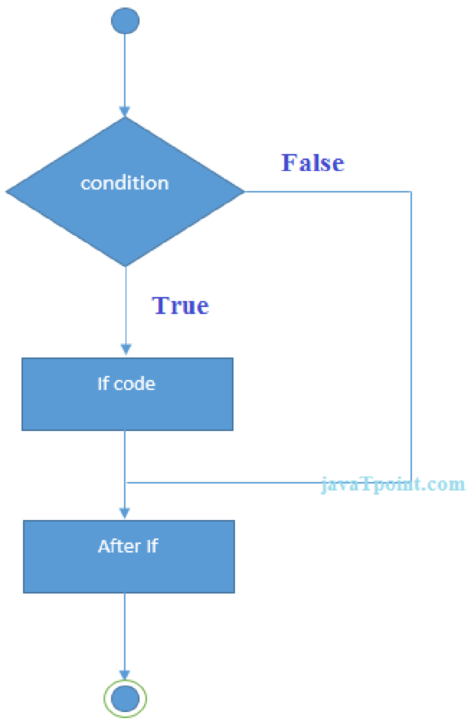
It evaluates the content only if expression is true. The signature of JavaScript if statement is given below.

if(expression){

//content to be evaluated

}

Flowchart of JavaScript If statement



Let's see the simple example of if statement in javascript.

<script>

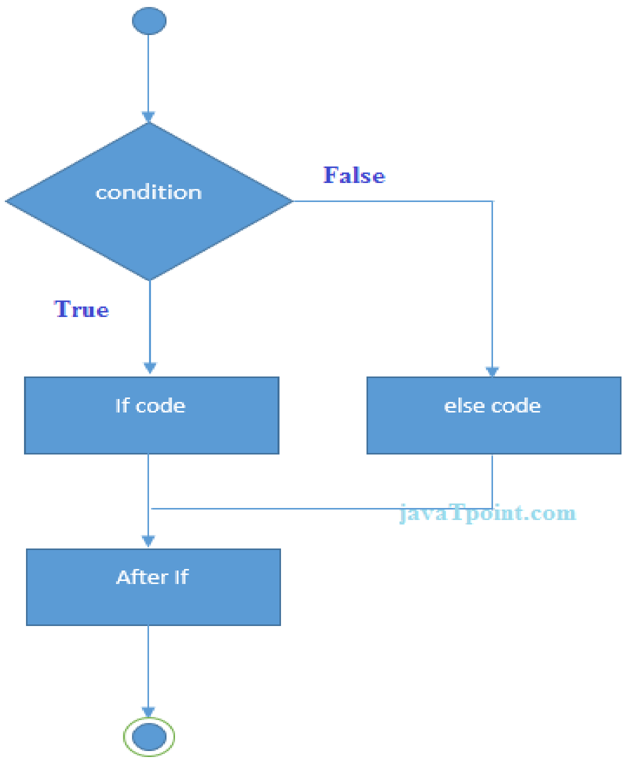
```
var a=20;
if(a>10){
document.write("value of a is greater than 10");
}
</script>
```

JavaScript If...else Statement

It evaluates the content whether condition is true of false. The syntax of JavaScript if-else statement is given below.

```
if(expression){
//content to be evaluated if condition is true
}
else{
//content to be evaluated if condition is false
}
```

Flowchart of JavaScript If...else statement



Let’s see the example of if-else statement in JavaScript to find out the even or odd number.

```
<script>
var a=20;
if(a%2==0){
document.write("a is even number");
}
else{
document.write("a is odd number");
}
</script>
```

JavaScript If...else if statement

It evaluates the content only if expression is true from several expressions. The signature of JavaScript if else if statement is given below.

```
if(expression1){
//content to be evaluated if expression1 is true
}
else if(expression2){
//content to be evaluated if expression2 is true
}
else if(expression3){
//content to be evaluated if expression3 is true
}
else{
//content to be evaluated if no expression is true
}
```

Let’s see the simple example of if else if statement in javascript.

```
<script>
var a=20;
if(a==10){
document.write("a is equal to 10");
}
else if(a==15){
document.write("a is equal to 15");
}
else if(a==20){
document.write("a is equal to 20");
}
else{
document.write("a is not equal to 10, 15 or 20");
}
</script>
```

JavaScript Switch

The **JavaScript switch statement** is used *to execute one code from multiple expressions*. It is just like else if statement that we have learned in previous page. But it is convenient than *if..else..if* because it can be used with numbers, characters etc.

The signature of JavaScript switch statement is given below.

```
switch(expression){
case value1:
 code to be executed;
 break;
case value2:
 code to be executed;
 break;
......

default:
 code to be executed if above values are not matched;
}
```

Let's see the simple example of switch statement in javascript.

```
<script>
```

```
var grade='B';
var result;
switch(grade){
case 'A':
result="A Grade";
break;
case 'B':
result="B Grade";
break;
case 'C':
result="C Grade";
break;
default:
result="No Grade";
}
document.write(result);
</script>
```

JavaScript Loops

The **JavaScript loops** are used *to iterate the piece of code* using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop
4. for-in loop

1) JavaScript For loop

The **JavaScript for loop** *iterates the elements for the fixed number of times*. It should be used if number of iteration is known. The syntax of for loop is given below.

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

```
{
    code to be executed
}
```

Let’s see the simple example of for loop in javascript.

```
<script>
for (i=1; i<=5; i++)
{
document.write(i + "<br/>")
}
</script>
```

2) JavaScript while loop

The **JavaScript while loop** *iterates the elements for the infinite number of times*. It should be used if number of iteration is not known. The syntax of while loop is given below.

```
while (condition)
{
    code to be executed
}
```

Let’s see the simple example of while loop in javascript.

```
<script>
var i=11;
while (i<=15)
{
document.write(i + "<br/>");
i++;
}
</script>
```

3) JavaScript do while loop

The **JavaScript do while loop** *iterates the elements for the infinite number of times* like while loop. But, code is *executed at least* once whether condition is true or false. The syntax of do while loop is given below.

```
do{
   code to be executed
}while (condition);
```

Let’s see the simple example of do while loop in javascript.

```
<script>
var i=21;
do{
document.write(i + "<br/>");
i++;
}while (i<=25);
</script>
```

JavaScript Functions

JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

1. **Code reusability**: We can call a function several times so it save coding.
2. **Less coding**: It makes our program compact. We don’t need to write many lines of code each time to perform a common task.

JavaScript Function Syntax

The syntax of declaring function is given below.

```
function functionName([arg1, arg2, ...argN]){
 //code to be executed
}
```

JavaScript Functions can have 0 or more arguments.

61.5M

JavaScript Function Example

Let’s see the simple example of function in JavaScript that does not has arguments.

```
<script>
function msg(){
alert("hello! this is message");
}
</script>
<input type="button" onclick="msg()" value="call function"/>
```

JavaScript Function Arguments

We can call function by passing arguments. Let’s see the example of function that has one argument.

```
<script>
function getcube(number){
alert(number*number*number);
}
</script>
<form>
<input type="button" value="click" onclick="getcube(4)"/>
</form>
```

Function with Return Value

We can call function that returns a value and use it in our program. Let’s see the example of function that returns value.

```
<script>
function getInfo(){
return "hello javatpoint! How r u?";
}
</script>
<script>
```

```
document.write(getInfo());
</script>
```

JavaScript Function Object

In JavaScript, the purpose of **Function constructor** is to create a new Function object. It executes the code globally. However, if we call the constructor directly, a function is created dynamically but in an unsecured way.

Syntax

new Function ([arg1[, arg2[, ...argn]],] functionBody)

Parameter

arg1, arg2, ... , argn - It represents the argument used by function.

functionBody - It represents the function definition.

JavaScript Function Methods

Let's see function methods with description.

Method	Description
apply()	It is used to call a function contains this value and a single array of arguments.
bind()	It is used to create a new function.
call()	It is used to call a function contains this value and an argument list.
toString()	It returns the result in a form of a string.

JavaScript Function Object Examples

Example 1

Let's see an example to display the sum of given numbers.

```
<script>
var add=new Function("num1","num2","return num1+num2");
document.writeln(add(2,5));
</script>
```

Example 2

Let's see an example to display the power of provided value.

```
<script>
var pow=new Function("num1","num2","return Math.pow(num1,num2)");
document.writeln(pow(2,3));
</script>
```

Results:

1. Students should able to develop the calculator using above concept.

Observations –

1. Student should able to apply the JavaScript for the static website with different client-side events and scripts.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 8

Title: Develop an application for demonstrating use of ReactJS.

ReactJS –

ReactJS is a simple, feature rich, component based JavaScript UI library. It can be used to develop small applications as well as big, complex applications. ReactJS provides minimal and solid feature set to kick-start a web application. React community compliments React library by providing large set of ready-made components to develop web application in a record time. React community also provides advanced concept like state management, routing, etc., on top of the React library.

React versions

The initial version, 0.3.0 of React is released on May, 2013 and the latest version, *17.0.1* is released on October, 2020. The major version introduces breaking changes and the minor version introduces new feature without breaking the existing functionality. Bug fixes are released as and when necessary. React follows the *Sematic Versioning (semver)* principle.

Features

The salient features of *React library* are as follows −

- Solid base architecture
- Extensible architecture
- Component based library
- JSX based design architecture
- Declarative UI library

Benefits

Few benefits of using *React library* are as follows −

- Easy to learn
- Easy to adept in modern as well as legacy application
- Faster way to code a functionality
- Availability of large number of ready-made component
- Large and active community

Applications

Few popular websites powered by *React library* are listed below −

- *Facebook*, popular social media application
- *Instagram*, popular photo sharing application
- *Netflix*, popular media streaming application
- *Code Academy*, popular online training application
- *Reddit*, popular content sharing application

Installation -

React provides CLI tools for the developer to fast forward the creation, development and deployment of the React based web application. React CLI tools depends on the Node.js and must be installed in your system. Hopefully, you have installed Node.js on your machine. We can check it using the below command −

```
node --version
```

You could see the version of Nodejs you might have installed. It is shown as below for me,

```
v14.2.0
```

If *Nodejs* is not installed, you can download and install by visiting https://nodejs.org/en/download/.

Toolchain

To develop lightweight features such as form validation, model dialog, etc., React library can be directly included into the web application through content delivery network (CDN). It is similar to using jQuery library in a web application. For moderate to big application, it is advised to write the application as multiple files and then use bundler such as webpack, parcel, rollup, etc., to compile and bundle the application before deploying the code.

React toolchain helps to create, build, run and deploy the React application. React toolchain basically provides a starter project template with all necessary code to bootstrap the application.

Some of the popular toolchain to develop React applications are −

- Create React App − SPA oriented toolchain
- Next.js − server-side rendering oriented toolchain
- Gatsby − Static content oriented toolchain

Tools required to develop a React application are −

- The *serve*, a static server to serve our application during development
- Babel compiler
- Create React App CLI

Let us learn the basics of the above mentioned tools and how to install those in this chapter.

The *serve* static server

The *serve* is a lightweight web server. It serves static site and single page application. It loads fast and consume minimum memory. It can be used to serve a React application. Let us install the tool using *npm* package manager in our system.

```
npm install serve -g
```

Let us create a simple static site and serve the application using *serve* app.

Open a command prompt and go to your workspace.

```
cd /go/to/your/workspace
```

Create a new folder, *static_site* and change directory to newly created folder.

```
mkdir static_site
cd static_site
```

Next, create a simple webpage inside the folder using your favorite html editor.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Static website</title>
  </head>
  <body>
    <div><h1>Hello!</h1></div>
  </body>
</html>
```

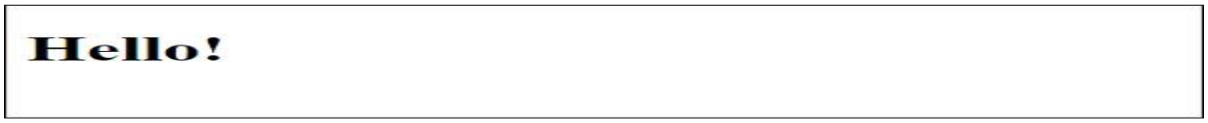
Next, run the *serve* command.

```
serve .
```

We can also serve single file, *index.html* instead of the whole folder.

```
serve ./index.html
```

Next, open the browser and enter *http://localhost:5000* in the address bar and press enter. serve application will serve our webpage as shown below.


Hello!

The serve will *serve* the application using default port, 5000. If it is not available, it will pick up a random port and specify it.

```
  │ Serving!                                   │
  │                                            │
  │ - Local: http://localhost:57311            │
  │ - On Your Network: http://192.168.56.1:57311 │
```

```
│                                                    │
│ This port was picked because 5000 is in use. │
│                                                    │
│ Copied local address to clipboard!
```

Babel compiler

Babel is a JavaScript compiler which compiles many variant (es2015, es6, etc.,) of JavaScript into standard JavaScript code supported by all browsers. React uses JSX, an extension of JavaScript to design the user interface code. Babel is used to compile the JSX code into JavaScript code.

To install Babel and it's React companion, run the below command −

```
npm install babel-cli@6 babel-preset-react-app@3 -g
...
...
+ babel-cli@6.26.0
+ babel-preset-react-app@3.1.2
updated 2 packages in 8.685s
```

Babel helps us to write our application in next generation of advanced JavaScript syntax.

Create React App toolchain

Create React App is a modern CLI tool to create single page React application. It is the standard tool supported by React community. It handles babel compiler as well. Let us install *Create React App* in our local system.

```
> npm install -g create-react-app
+ create-react-app@4.0.1
added 6 packages from 4 contributors, removed 37 packages and updated 12 packages in 4.693s
```

Updating the toolchain

React Create App toolchain uses the react-scripts package to build and run the application. Once we started working on the application, we can update the react-script to the latest version at any time using *npm* package manager.

```
npm install react-scripts@latest
```

React library is built on a solid foundation. It is simple, flexible and extensible. As we learned earlier, React is a library to create user interface in a web application. React's primary purpose is to enable the developer to create user interface using pure JavaScript. Normally, every user interface library introduces a new template language (which we need to learn) to design the user interface and provides an option to write logic, either inside the template or separately.

Instead of introducing new template language, React introduces three simple concepts as given below −

React elements

JavaScript representation of HTML DOM. React provides an API, ***React.createElement*** to *create React Element*.

JSX

A JavaScript extension to design user interface. JSX is an XML based, extensible language supporting HTML syntax with little modification. JSX can be compiled to React Elements and used to create user interface.

React component

React component is the primary building block of the React application. It uses React elements and JSX to design its user interface. React component is basically a JavaScript class (extends the ***React.component*** class) or pure JavaScript function. React component has properties, state management, life cycle and event handler. React component can be able to do simple as well as advanced logic.

Let us learn more about components in the React Component chapter.

Workflow of a React application

Let us understand the workflow of a React application in this chapter by creating and analyzing a simple React application.

Open a command prompt and go to your workspace.

```
cd /go/to/your/workspace
```

Next, create a folder, *static_site* and change directory to newly created folder.

```
mkdir static_site
cd static_site
```

Example

Next, create a file, *hello.html* and write a simple React application.

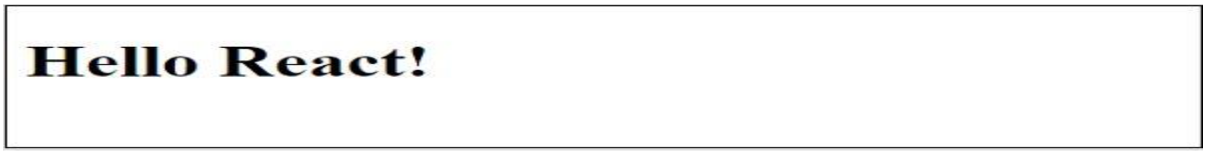
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>React Application</title>
  </head>
  <body>
    <div id="react-app"></div>
    <script src="https://unpkg.com/react@17/umd/react.development.js" crossorigin></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"
crossorigin></script>
    <script language="JavaScript">
      element = React.createElement('h1', {}, 'Hello React!')
      ReactDOM.render(element, document.getElementById('react-app'));
    </script>
  </body>
</html>
```

Next, serve the application using serve web server.

```
serve ./hello.html
```

Output

Next, open your favorite browser. Enter **http://localhost:5000** in the address bar and then press enter.


Hello React!

Let us analyse the code and do little modification to better understand the React application.

Here, we are using two API provided by the React library.

React.createElement

Used to create React elements. It expects three parameters −

- Element tag
- Element attributes as object
- Element content - It can contain nested React element as well

ReactDOM.render

Used to render the element into the container. It expects two parameters −

- React Element OR JSX
- Root element of the webpage

Nested React element

As ***React.createElement*** allows nested React element, let us add nested element as shown below –

Example

```
<script language="JavaScript">
   element = React.createElement('div', {}, React.createElement('h1', {}, 'Hello React!'));
   ReactDOM.render(element, document.getElementById('react-app'));
</script>
```

Output

It will generate the below content −

```
<div><h1> Hello React!</h1></div>
```

Use JSX

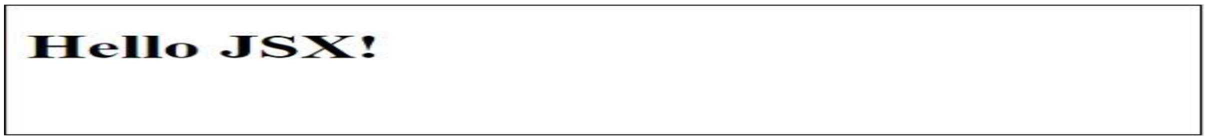
Next, let us remove the React element entirely and introduce JSX syntax as shown below −

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>React Application</title>
  </head>
  <body>
    <div id="react-app"></div>
    <script src="https://unpkg.com/react@17/umd/react.development.js" crossorigin></script>
    <script                    src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"
crossorigin></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
    <script type="text/babel">
      ReactDOM.render(
        <div><h1>Hello React!</h1></div>,
        document.getElementById('react-app')
      );
   </script>
  </body>
</html>
```

Here, we have included babel to convert JSX into JavaScript and added *type="text/babel"* in the script tag.

```
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
<script type="text/babel">
  ...
  ...
</script>
```

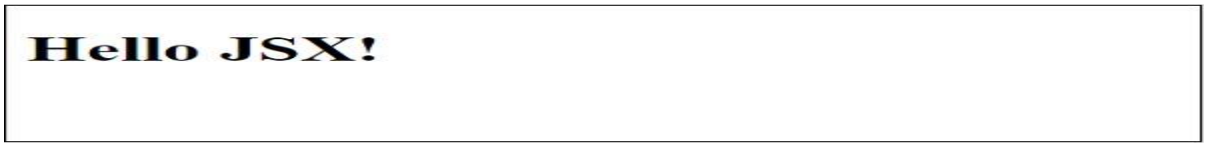
Next, run the application and open the browser. The output of the application is as follows −


Hello JSX!

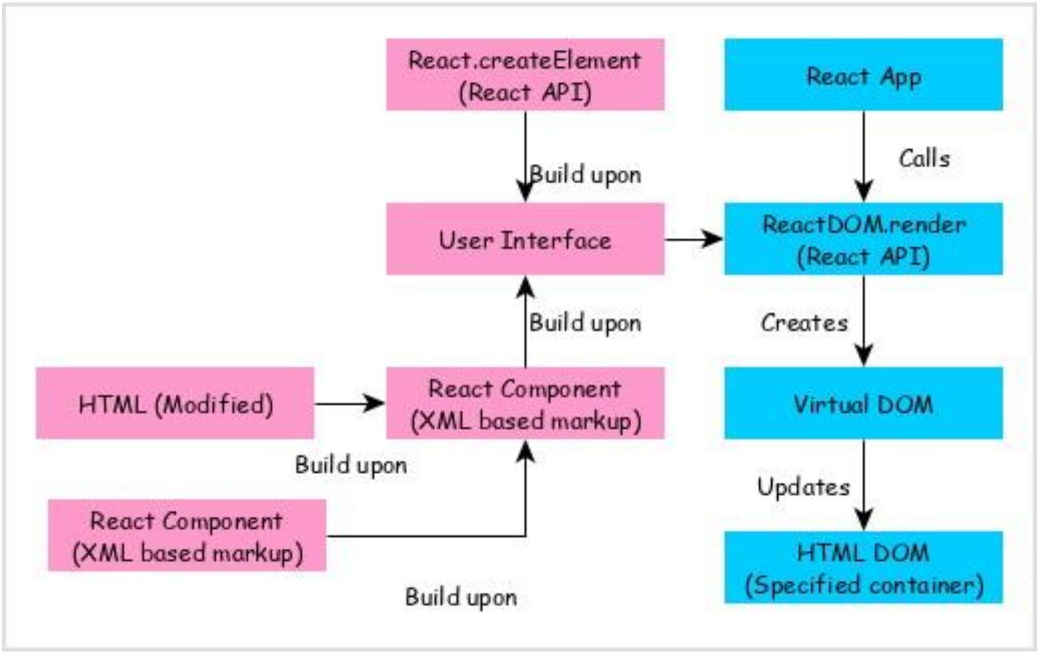
Next, let us create a new React component, Greeting and then try to use it in the webpage.

```
<script type="text/babel">
  function Greeting() {
    return <div><h1>Hello JSX!</h1></div>
  }
  ReactDOM.render(<Greeting />, document.getElementById('react-app') );
</script>
```

The result is same and as shown below −


Hello JSX!

By analyzing the application, we can visualize the workflow of the React application as shown in the below diagram.


React.createElement (React API)
React App
Calls
Build upon
User Interface
ReactDOM.render (React API)
Build upon
Creates
HTML (Modified)
React Component (XML based markup)
Virtual DOM
Build upon
Updates
React Component (XML based markup)
HTML DOM (Specified container)
Build upon

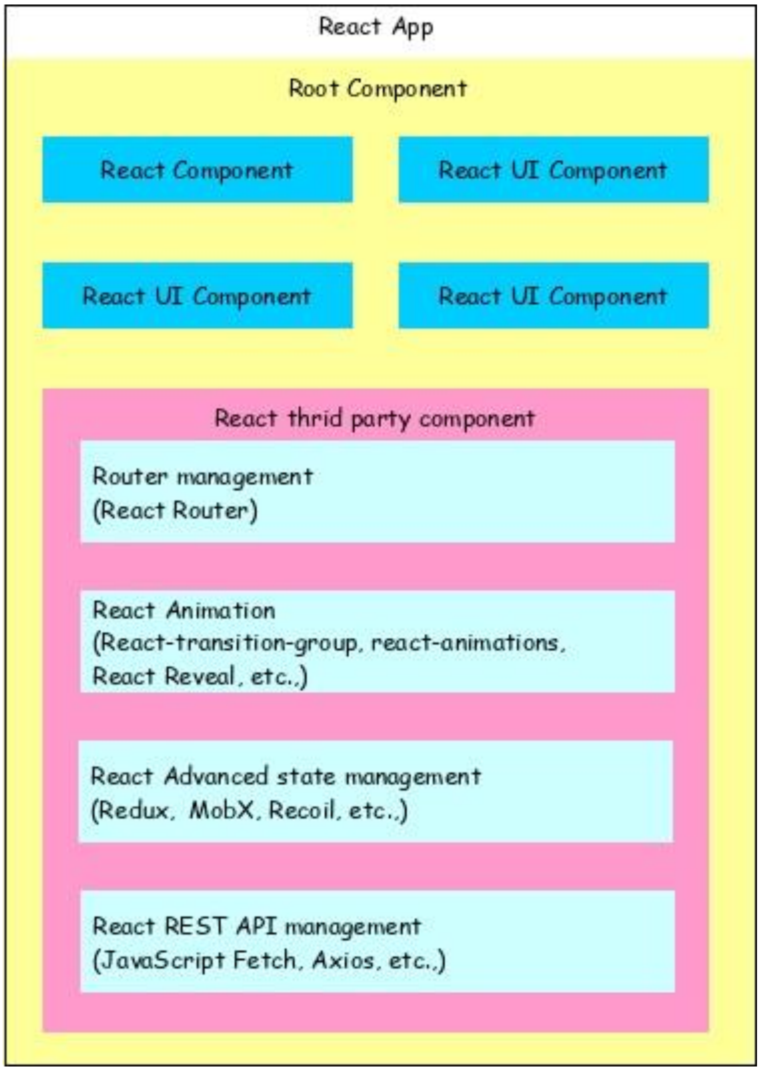
React app calls ***ReactDOM.render*** method by passing the user interface created using React component (coded in either JSX or React element format) and the container to render the user interface.

ReactDOM.render processes the JSX or React element and emits Virtual DOM.

Virtual DOM will be merged and rendered into the container.

Architecture of the React Application

React library is just UI library and it does not enforce any particular pattern to write a complex application. Developers are free to choose the design pattern of their choice. React community advocates certain design pattern. One of the patterns is Flux pattern. React library also provides lot of concepts like Higher Order component, Context, Render props, Refs etc., to write better code. React Hooks is evolving concept to do state management in big projects. Let us try to understand the high level architecture of a React application.


React App
Root Component
React Component
React UI Component
React UI Component
React UI Component
React thrid party component
Router management
(React Router)
React Animation
(React-transition-group, react-animations,
React Reveal, etc.,)
React Advanced state management
(Redux, MobX, Recoil, etc.,)
React REST API management
(JavaScript Fetch, Axios, etc.,)

- React app starts with a single root component.
- Root component is build using one or more component.
- Each component can be nested with other component to any level.
- Composition is one of the core concepts of React library. So, each component is build by composing smaller components instead of inheriting one component from another component.
- Most of the components are user interface components.
- React app can include third party component for specific purpose such as routing, animation, state management, etc.

Results:

1. Students should able to develop the sample program using React

Observations –

1. Student should able to apply the React for the website

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 9

S/W Requirement – WAMP/LAMP/XAMPP

Title: Study of PHP basics along with different modern tool installation and configuration.

Theory:

The **PHP Hypertext Preprocessor (PHP)** is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web-based software applications

The origins of PHP date back to 1995 when an independent software development contractor named Rasmus Lerdorf developed a Perl/CGI script that enabled him to know how many visitors were reading his online résumé. His script performed two tasks: logging visitor information, and displaying the count of visitors to the web page. Because the Web at the time was still a fledgling technology, tools such as these were non-existent. Thus, Lerdorf's script generated quite a bit of interest. Lerdorf began giving away his toolset, dubbed Personal Home Page (PHP).

Key advantages of learning PHP:

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

Applications of PHP

As mentioned before, PHP is one of the most widely used language over the web. I'm going to list few of them here:

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Basic PHP Syntax

A PHP script can be placed anywhere in the document.A PHP script starts with **<?php** and ends with **?>**:

```
<?php
// PHP code goes here
?>
```

The default file extension for PHP files is ".php".A PHP file can be embedded in HTML tags, and some PHP scripting code.

```
<html>
<body>
<h1>My first PHP page</h1>
<?php
echo "Hello World!";
          ?>
</body>
</html>
```

Installation of WAMP server:

WAMP Server is a server which is used to host PHP pages. PHP is a server-side scripting language developed by Rasmus Lerdorf.

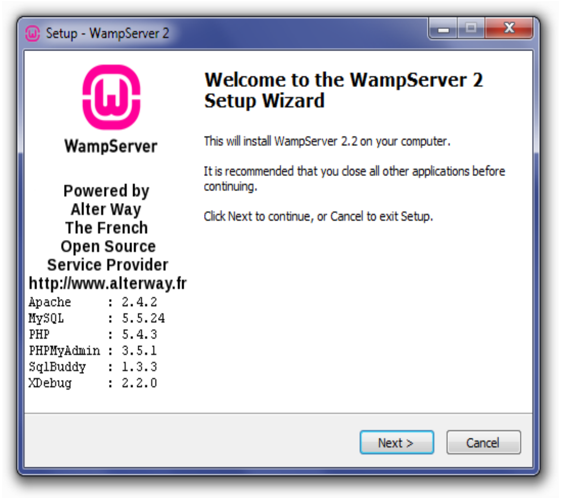
WAMP stands for – Window Apache MySQL and PHP.

Developed by – Romain Bourdon

Operating system – Windows

Now the steps for WAMP Server Installation as –

1. To start the installation process, you need to open the folder where you saved the file, and double-click the installer file. A security warning window will open, asking if you are sure you want to run this file. Click Run to start the installation process.



Welcome to the WampServer 2 Setup Wizard

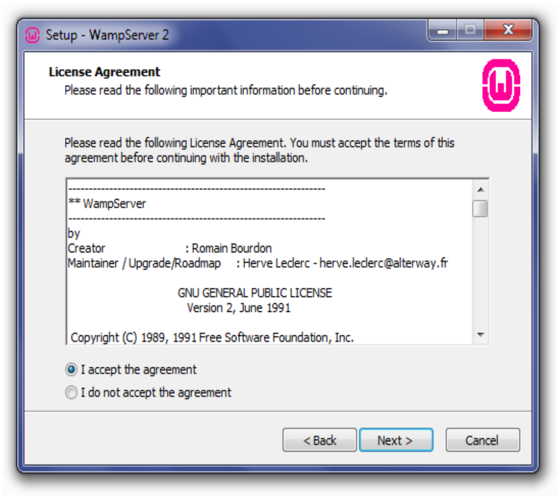
This will install WampServer 2.2 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Powered by Alter Way The French Open Source Service Provider http://www.alterway.fr

```
Apache     : 2.4.2
MySQL      : 5.5.24
PHP        : 5.4.3
PHPMyAdmin : 3.5.1
SqlBuddy   : 1.3.3
XDebug     : 2.2.0
```



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

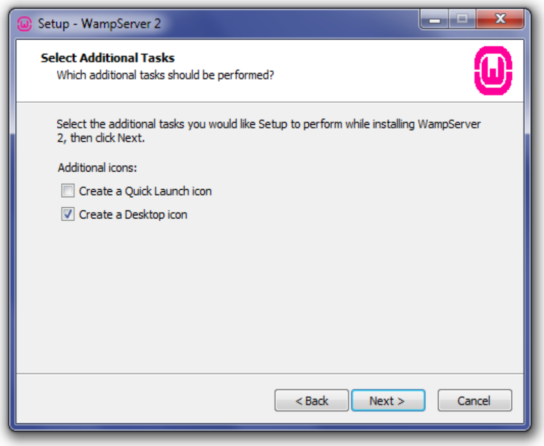
```
------------------------------------------------
** WampServer
------------------------------------------------
by
Creator                  : Romain Bourdon
Maintainer / Upgrade/Roadmap   : Herve Leclerc - herve.leclerc@alterway.fr

              GNU GENERAL PUBLIC LICENSE
                 Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
```

[✓ I accept the agreement]
I do not accept the agreement

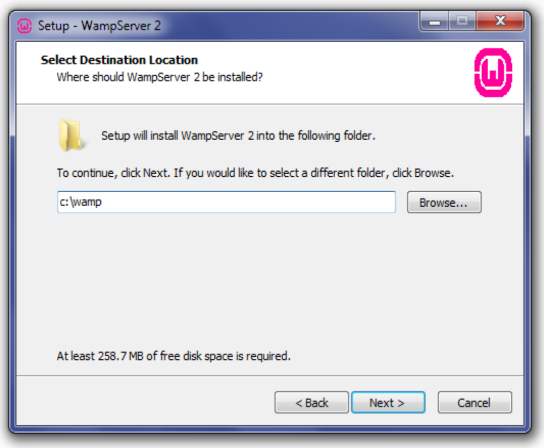
2. The next screen you are presented with is the License Agreement. Read the agreement, check the radio button next to I accept the agreement, then click Next to continue the installation.


Setup - WampServer 2

Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing WampServer 2, then click Next.

Additional icons:
- [ ] Create a Quick Launch icon
- [x] Create a Desktop icon


Setup - WampServer 2

Select Destination Location
Where should WampServer 2 be installed?

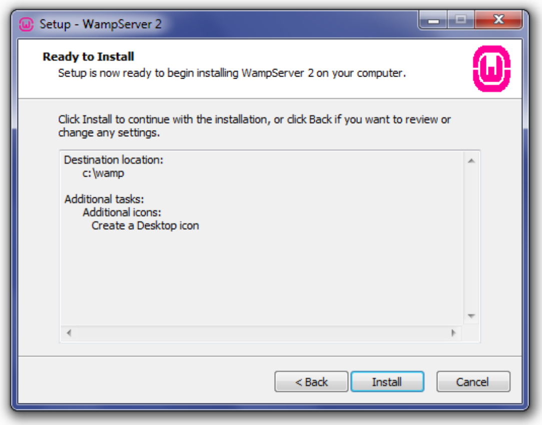
Setup will install WampServer 2 into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

c:\wamp

At least 258.7 MB of free disk space is required.

3. Next you will see the Select Destination Location screen. Unless you would like to install WampServer on another drive, you should not need to change anything. Click Next to continue

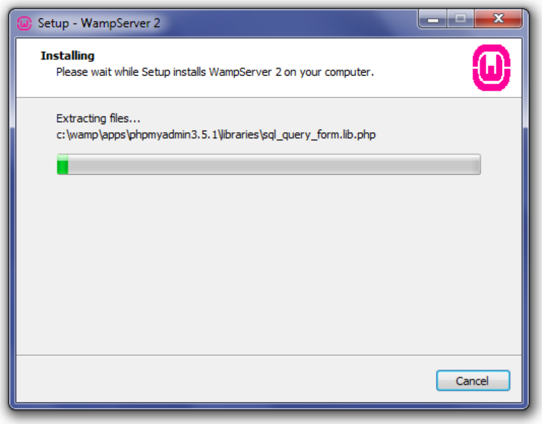

Setup - WampServer 2

Ready to Install
Setup is now ready to begin installing WampServer 2 on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
    c:\wamp

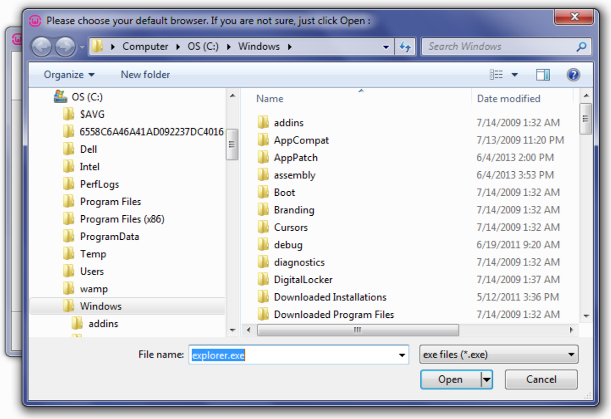
Additional tasks:
    Additional icons:
        Create a Desktop icon


Setup - WampServer 2

Installing
Please wait while Setup installs WampServer 2 on your computer.

Extracting files...
c:\wamp\apps\phpmyadmin3.5.1\libraries\sql_query_form.lib.php

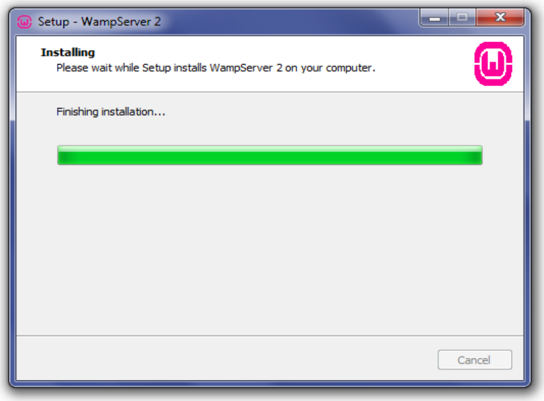
4.The next screen you are presented with is the Select Additional Tasks screen. You will be able to select whether you would like a Quick Launch icon added to the taskbar or a Desktop icon created once installation is complete. Make your selections, then click Next to continue


Please choose your default browser. If you are not sure, just click Open :

Computer ▸ OS (C:) ▸ Windows ▸

Name	Date modified
addins	7/14/2009 1:32 AM
AppCompat	7/13/2009 11:20 PM
AppPatch	6/4/2013 2:00 PM
assembly	6/4/2013 3:53 PM
Boot	7/14/2009 1:32 AM
Branding	7/14/2009 1:32 AM
Cursors	7/14/2009 1:32 AM
debug	6/19/2011 9:20 AM
diagnostics	7/14/2009 1:32 AM
DigitalLocker	7/14/2009 1:37 AM
Downloaded Installations	5/12/2011 3:36 PM
Downloaded Program Files	7/14/2009 1:32 AM

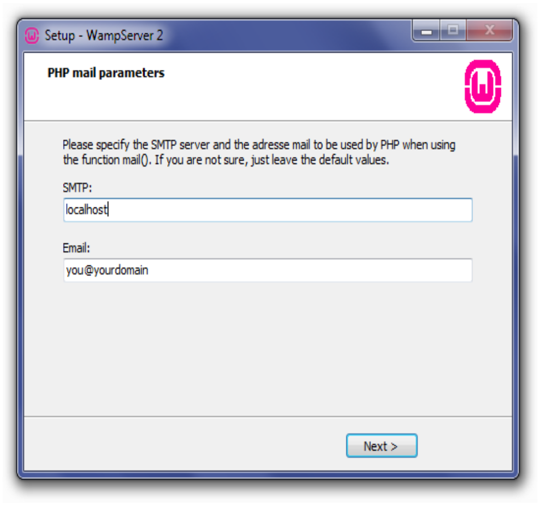
File name: explorer.exe   exe files (*.exe)


Setup - WampServer 2

Installing
Please wait while Setup installs WampServer 2 on your computer.

Finishing installation...

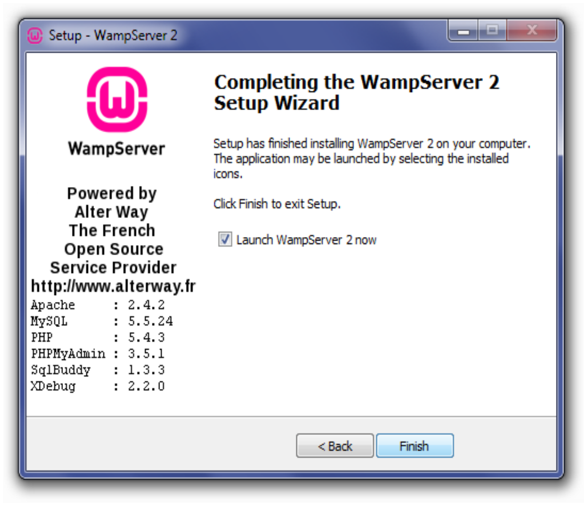
5. Next you will see the Ready To Install screen. You can review your setup choices, and change any of them by clicking Back to the appropriate screen, if you choose to. Once you have reviewed your choices, click Install to continue.



Please specify the SMTP server and the adresse mail to be used by PHP when using the function mail(). If you are not sure, just leave the default values.

SMTP: localhost

Email: you@yourdomain



Completing the WampServer 2 Setup Wizard

Setup has finished installing WampServer 2 on your computer. The application may be launched by selecting the installed icons.

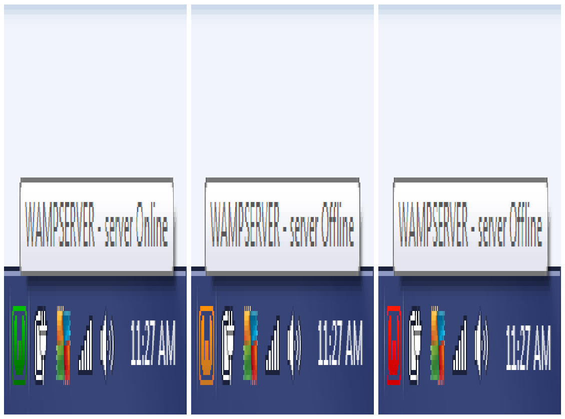
Click Finish to exit Setup.

[✓] Launch WampServer 2 now

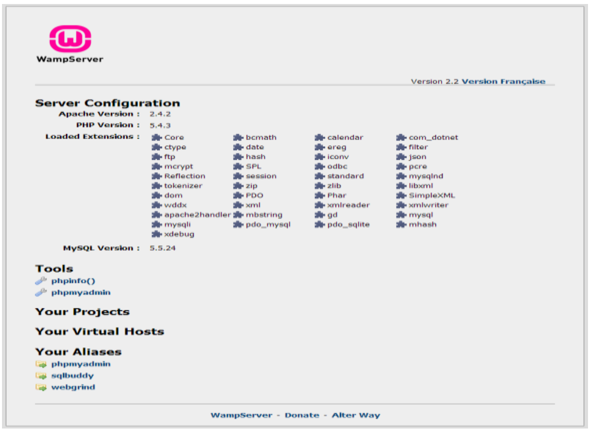
Powered by Alter Way The French Open Source Service Provider http://www.alterway.fr

```
Apache     : 2.4.2
MySQL      : 5.5.24
PHP        : 5.4.3
PHPMyAdmin : 3.5.1
SqlBuddy   : 1.3.3
XDebug     : 2.2.0
```

6. WampServer will begin extracting files to the location you selected



WAMPSERVER - server Online
11:27 AM
WAMPSERVER - server Offline
11:27 AM
WAMPSERVER - server Offline
11:27 AM



Server Configuration

```
Apache Version :    2.4.2
PHP Version :       5.4.3
Loaded Extensions : Core            bcmath      calendar    com_dotnet
                    ctype           date        ereg        filter
                    ftp             hash        iconv       json
                    mcrypt          SPL         odbc        pcre
                    Reflection      session     standard    mysqlnd
                    tokenizer       zip         zlib        libxml
                    dom             PDO         Phar        SimpleXML
                    wddx            xml         xmlreader   xmlwriter
                    apache2handler  mbstring    gd          mysql
                    mysqli          pdo_mysql   pdo_sqlite  mhash
                    xdebug
MySQL Version :     5.5.24
```

Tools
phpinfo()
phpmyadmin

Your Projects

Your Virtual Hosts

Your Aliases
phpmyadmin
sqlbuddy
webgrind

WampServer - Donate - Alter Way

PHP Variable is nothing it is just name of the memory location. A Variable is simply a container i.e. used to store both numeric and non-numeric information.

Rules for Variable declaration

- Variables in PHP starts with a **dollar($)** sign, followed by the name of the variable.
- The variable name must begin with a letter or the underscore character.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and_)
- A variable name should not contain space

Assigning Values to Variables

Assigning a value to a variable in PHP is quite east: use the equality(=) symbol, which also to the PHP's assignment operators. This assign value on the right side of the equation to the variable on the left.

E.g.
```
 <?php
$str = "Hello";
echo $str;
 ?>
```

Control Statements

When we run a program, the statements are executed in the order in which they appear in the program. Also each statement is executed only once. But in many cases we may need a statement or a set of statements to be executed a fixed no of times or until a condition is satisfied. Also we may want to skip some statements based on testing a condition. For all these we use control statements. Control statements are of two types – branching and looping.

a) BRANCHING:
It is to execute one of several possible options depending on the outcome of a logical test, which is carried at some particular point within a program.

b) LOOPING:
It is to execute a group of instructions repeatedly, a fixed no of times or until a specified condition is satisfied.

a) BRANCHING

1. **if else statement**

It is used to carry out one of the two possible actions depending on the outcome of a logical test. The else portion is optional.

The syntax is
```
if (expression)
statement1 [if there is no else part]
                    Or
 if (expression)
Statement 1
 else
Statement 2
```

Here expression is a logical expression enclosed in parenthesis. If expression is true ,statement 1 or statement 2 is a group of statements, they are written as a block using the braces { }

2. **Ladder if statement**

Inorder to create a situation in which one of several courses of action is executed we use ladder – if statements.

The syntax is

```
If (expression1)
Statement 1
else if (expression2)
Statement 2
 else if (expression3)
Statement 3
  ..................
else
Statement n
```

3. **Switch Statement**

It is used to execute a particular group of statements to be chosen from several available options. The selection is based on the current value of an expression with the switch statement.

The syntax is:

```
switch(expression)
{
           case value1:
                       Statement 1
                      break;
          case value 2:
                      Statement 2
                     break;
                      .......
                      ........
           default:
                 Statement n
}
```

All the option are embedded in the two braces { }.Within the block each group is written after the label case followed by the value of the expression and a colon. Each group ends with ‘break’ statement. The last may be labeled ‘default’. This is to avoid error and to execute the group of statements in default if the value of the expression does not match value1, value2,........

b) LOOPING

1. **The while statement**

This is to carry out a set of statements to be executed repeatedly until some condition is satisfied.

The syntax is:

```
While (expression)
{
   statement
}
```

The statement is executed so long as the expression is true. Statement can be simple or compound.

2. do while statement

This is also to carry out a set of statements to be executed repeatedly so long as a condition is true.

The syntax is:

```
do
{
   statement
}while(expression);
```

3. for loop

It is the most commonly used looping statement in C.

The syntax is:

```
for(expression1;expression2;expression}
 {
          statement


  }
```

Here expression1 is to initialize some parameter that controls the looping action. Expression2 is a condition and it must be true to carry out the action. Expression3 is a unary expression or an assignment expression.

PHP Functions

- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute immediately when a page loads.
- A function will be executed by a call to the function.

User Defined Function in PHP

```
 function functionName()
{
   code to be executed;
}
```

Example

```
<?php
        function writeMsg()
       {
          echo "Hello world!";
        }

       writeMsg(); // call the function
  ?>
```

Functions with Parameters

```
function function_name($firstarg, $secondarg)
      {
     // Code to be executed
      }
```

Example

```
<?php
            // Defining function
function getSum($num1, $num2)
{
$sum = $num1 + $num2;
echo "Sum of the two numbers $num1 and $num2 is : $sum";
}
            // Calling function
getSum(10, 20);
?>
```

Results:

1. Thus, student must understand the basic concepts of PHP programming and structure of PHP program.
2. Student should be able to write the basic PHP program structure.

Observations:

1. Students have to study the PHP and its basic programming structure.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory	**Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 10

S/W Requirement – WAMP/LAMP/XAMPP

Title: Implementation of PHP Array with different functions.

Theory:

An array is a special variable, which can hold more than one value at a time. Arrays are complex variables that allow us to store more than one value or a group of values under a single variable name. In PHP, the array() function is used to create an array

syntax : *array(val1,val2,..);*

e.g. *$arr = **array**(10,11,12,13,14,15);*

*$colors = **array**("Red", "Green", "Blue");*

Outputting an Array

```
<?php
$colors = array("Red", "Green", "Blue");
echo "Colors are" . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

In PHP, there are three types of arrays:

- **Indexed arrays** - Arrays with a numeric index
- **Associative arrays** - Arrays with named keys
- **Multidimensional arrays** - Arrays containing one or more arrays

PHP Indexed Arrays

1 . The index can be assigned automatically (index always starts at 0),

```
<?php
    $colors = array("Red", "Green", "Blue");
?>
```

2. The indexes are assigned manually

```
<?php
        $colors[0] = "Red";
        $colors[1] = "Green";
        $colors[2] = "Blue";
?>
```

PHP Associative array

- In associative array index(key) can initialized according to Your own requirement.
- An array in PHP is actually an ordered map. A map is a type that associates values to keys.
- In this association use (=>) sign to define index and values.

E.g .

```
<?php
// Define an associative array
$ages = array("Peter"=>22, "Clark"=>32, "John"=>28);
echo “Peter is ".$ages[" Peter "]."Years old";
?>
```

PHP Multidimensional Arrays

- **Syntax**

  array(***array***(*val1, val1, val2..*), ***array***(*val1,val2,val3..*))
- E.g

```
$num=array (
    array(10,12,14),
    array(15,30,30),
    array(10,10,10)
 );
```

Array Functions:

The number of functions are provided by PHP which are capable of performing complex array-manipulation tasks.

1. For checking the variable is array or not:
   is_array(): The is_array() function determines whether a variable is an array, returning TRUE if it is and FALSE otherwise.
   Syntax: boolean is_array(mixed variable)

2. Printing Array for Testing Purpose:
   print_r(): The print_r() function accepts a variable and sends its contents to standard output, returning TRUE on success and FALSE otherwise.

Syntax: boolean print_r(mixed variable [, boolean return])

3. Merging Arrays:
   array_merge(): The array_merge() function merges arrays together, resulting a single, unified array.
   Syntax: array array_merge(array array1, array array2 [, array arrayN])

4. Recursively Appending Arrays:
   array_merge_recursive(): The array_merge_recursive() function is identical to the array_merge() function. The difference is array_merge() will overwrite the pre-existing key/value pair while array_merge_recursive() will form a new array with the pre-existing key as its name.
   Syntax: array array_merge_recursive(array array1, array array2 [, array arrayN])

5. Combining Two Arrays:
   array **array_combine**(array key, array value)

6. Slicing an Array:
   array **array_slice**(array *array*, int *offset* [, int *length* [, boolean *preserve_keys*]])

7. Splicing an Array:
   array **array_splice**(array *array*, int *offset* [, int *length* [, array *replacement*]])

8. Calculating an Array Intersection:
   array **array_intersect**(array *array1*, array *array2* [, *arrayN*])

9. Calculating Associative Array Intersections:
   array **array_intersect_assoc**(array *array1*, array *array2* [, arrayN])

10. Calculating Array Differences:
    array **array_diff**(array *array1*, array *array2* [, arrayN])

11. Calculating Associative Array Differences:
    array **array_diff_assoc**(array *array1*, array *array2* [, array *arrayN*])

In addition to all these array manipulation functions we have some other useful array functions like: **array_rand(),shuffle(),array_sum()** and **array_chunk()** used for performing important array related transformations.

Results:

1. Thus, student must understand the basic concepts of arrays and its functions in PHP programming.
2. Student should be able to write the PHP program using arrays and its functions.

Observations:

1. Students have studied concepts of arrays and its functions in PHP programming.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory	**Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 11

S/W Requirement – WAMP/LAMP/XAMPP

Title: Implementation object-oriented concepts in PHP.

Theory:

Inheritance

One of the main advantages of object-oriented programming is the ability to reduce code duplication with **inheritance**. Code duplication occurs when a programmer writes the same code more than once, a problem that **inheritance** strives to solve. In inheritance, we have a parent class with its own methods and properties, and a child class (or classes) that can use the code from the parent. By using inheritance, we can create a reusable piece of code that we write only once in the parent class, and use again as much as we need in the child classes. Inheritance allows us to write the code only once in the parent, and then use the code in both the parent and the child classes. In order to declare that one class inherits the code from another class, we use **extends** keyword. Let's see the general case:

```
    class Parent
    {
          // The parent's class code
     }

class Child extends Parent
 {
  // The  child can use the parent's class code
}
```

Example

```
//The parent class
class Car
 {
  // Private property inside the class
  private $model;
  //Public setter method
  public function setModel($model)
  {
    $this -> model = $model;
  }
   public function hello()
  {
    return "beep! I am a <i>" . $this ->model . "</i><br />";
  }
}

 //The child class inherits the code from the parent class
class SportsCar extends Car
 {
  //No code in the child class
}

 //Create an instance from the child class
$sportsCar1 = new SportsCar();

// Set the value of the class’ property.
// For this aim, we use a method that we created in the parent
$sportsCar1 ->setModel('Mercedes Benz');

//Use another method that the child class inherited from the parent class
echo $sportsCar1 -> hello();
```

Interface

Interfaces resemble abstract classes in that they include abstract methods that the programmer must define in the classes that inherit from the interface. In this way, interfaces contribute to code organization because they commit the child classes to abstract methods that they should implement. The use of interfaces becomes very helpful when we work in a team of programmers and want to ensure that all the programmers write the methods that they should work on, or even in the case of a single programmer that wants to commit himself to write certain methods in the child classes.

How to declare and implement an interface?

To declare an interface with the **interface** keyword and, the class that inherits from an interface with the implements keyword. Let's see the general case:

```
Interface interfaceName
{
                    // abstract methods
                   }

class Child implements interfaceName
{
                     // defines the interface methods and may have its own code
                   }
```

Example

```
interface Car
{
  public function setModel($name);

  public function getModel();
}

Class minicar implements Car {
  private $model;

  public function setModel($name)
  {
    $this -> model = $name;
  }

  Public function getModel()
  {
    return $this -> model;
  }
}
```

We can implement a number of interfaces in the same class.

```
interface Vehicle
 {
  public function setHasWheels($bool);

 public function getHasWheels();
}
class minicar implements Car, Vehicle {
  private $model;
  private $hasWheels;

  public function setModel($name)
  {
    $this -> model = $name;
  }

  public function getModel()
  {
    return $this -> model;
  }

  Public function setHasWheels($bool)
  {
    $this ->hasWheels = $bool;
  }

  public function getHasWheels()
  {
  return ($this ->hasWheels)? "has wheels" : "no wheels";
  }
}
```

Results:

1. Thus, student must understand the basic concepts of OOP (Inheritance and Interface) in PHP.
2. Student should be able to write the PHP program using basic OOP concepts like Inheritance and Interface.

Observations:

1. Students have studied the OOP concepts in PHP programming.

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 12

S/W Requirement – WAMP/LAMP/XAMPP

Title: Implementation of form and extract the user input using $_GET or $_POST

.

Theory:

The PHP superglobals $_GET and $_POST are used to collect form-data. The form request may be get or post. To retrieve data from get request, we need to use $_GET, for post request $_POST.

PHP Get Form

Get request is the default form request. The data passed through get request is visible on the URL browser so it is not secured. You can send limited amount of data through get request.

Let's see a simple example to receive data from get request in PHP.

File: form1.html

```
<form action="welcome.php" method="get">
Name: <input type="text" name="name"/>
<input type="submit" value="visit"/>
</form>
```

File: welcome.php

```
<?php
$name=$_GET["name"];//receiving name field value in $name variable
echo "Welcome, $name";
?>
```

PHP Post Form

Post request is widely used to submit form that have large amount of data such as file upload, image upload, login form, registration form etc.The data passed through post request is not visible on the URL browser so it is secured. You can send large amount of data through post request.

Let's see a simple example to receive data from post request in PHP.

File: form1.html

```
<form action="login.php" method="post">
<table>
<tr><td>Name:</td><td> <input type="text" name="name"/></td></tr>
<tr><td>Password:</td><td> <input type="password" name="password"/></td></tr>
<tr><td colspan="2"><input type="submit" value="login"/>  </td></tr>
</table>
</form>
```

File: login.php

```
<?php
$name=$_POST["name"];//receiving name field value in $name variable
$password=$_POST["password"];//receiving password field value in $password variable

echo "Welcome: $name, your password is: $password";
?>
```

Results:

1. Thus, student must understand the basic concepts of how to get the users data entered in the form in PHP.

Observations:

1. Students have studied how to extract the data from the form in PHP.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 13

S/W Requirement – WAMP/LAMP/XAMPP

Title: Design and develop an application for uploading various types of files using PHP.
.

Theory:

A PHP script can be used with a HTML form to allow users to upload files to the server. Initially files are uploaded into a temporary directory and then relocated to a target destination by a PHP script.

The process of uploading a file follows these steps −

- The user opens the page containing a HTML form featuring a text files, a browse button and a submit button.
- The user clicks the browse button and selects a file to upload from the local PC.
- The full path to the selected file appears in the text filed then the user clicks the submit button.
- The selected file is sent to the temporary directory on the server.
- The PHP script that was specified as the form handler in the form's action attribute checks that the file has arrived and then copies the file into an intended directory.
- The PHP script confirms the success to the user.

Creating an upload form

```
<?php
  if(isset($_FILES['image'])){
    $errors= array();
    $file_name = $_FILES['image']['name'];
    $file_size =$_FILES['image']['size'];
    $file_tmp =$_FILES['image']['tmp_name'];
    $file_type=$_FILES['image']['type'];
    $file_ext=strtolower(end(explode('.',$_FILES['image']['name'])));
```

```
      $expensions= array("jpeg","jpg","png");

      if(in_array($file_ext,$expensions)=== false){
         $errors[]="extension not allowed, please choose a JPEG or PNG file.";
      }

      if($file_size > 2097152){
         $errors[]='File size must be excately 2 MB';
      }

      if(empty($errors)==true){
         move_uploaded_file($file_tmp,"images/".$file_name);
         echo "Success";
      }else{
         print_r($errors);
      }
   }
?>

<html>
<body>

<form action="" method="POST" enctype="multipart/form-data">
<input type="file" name="image" />
<input type="submit"/>
</form>

</body>
</html>
```

Creating an upload script

There is one global PHP variable called $_FILES. This variable is an associate double dimension array and keeps all the information related to uploaded file. So if the value assigned to the input's name attribute in uploading form was file, then PHP would create following five variables –

- **$_FILES['file']['tmp_name']** − the uploaded file in the temporary directory on the web server.

- **$_FILES['file']['name']** − the actual name of the uploaded file.
- **$_FILES['file']['size']** − the size in bytes of the uploaded file.
- **$_FILES['file']['type']** − the MIME type of the uploaded file.
- **$_FILES['file']['error']** − the error code associated with this file upload.

PHP's File-Upload Functions:

PHP offers two functions specifically intended to aid in the file upload process, is_uploaded_file() and move_uploaded_file().

- is_uploaded_file() function determines whether a file specified by the input parameter filename is uploaded using the POST method. Its prototype follows:
  boolean is_uploaded_file(string *filename*)
- move_uploaded_file() function provides a convenient means for moving an uploaded file from the temporary directory to a final location. Its prototype follows: boolean move_uploaded_file(string *filename*, string *destination*)

Results:

1. Thus, student must understand the basic concepts of uploading various types of files in PHP.
2. Student should be able to upload different types of files in PHP.

Observations:

1. Students have studied how to upload different types of files in PHP.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 14

S/W Requirement – WAMP Server

Title: Design and develop a program for storing and retrieving the information from session and cookies using PHP

Theory:

A session is a way to store information (in variables) to be used across multiple pages.

What is a PHP Session?

When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.So; Session variables hold information about one single user, and are available to all pages in one application.

Start a PHP Session

A session is started with the session_start() function.

```
<?php
session_start();
?>
```

Destroy a PHP Session

To remove all global session variables and destroy the session, use session_unset() and session_destroy()

```
<?php
```

```
// remove all session variables
session_unset();
// destroy the session
session_destroy();
?>
```

For Example – PHP Login Form with Sessions

To set the values to the session –

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

To access the value of session variables –

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
```

```
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

A **cookie** in PHP is a small file with a maximum size of 4KB that the web server stores on the client computer. They are typically used to keep track of information such as a username that the site can retrieve to personalize the page when the user visits the website next time. A cookie can only be read from the domain that it has been issued from. Cookies are usually set in an HTTP header but JavaScript can also set a cookie directly on a browser.

Setting Cookie In PHP: To set a cookie in PHP, the setcookie() function is used. The setcookie() function needs to be called prior to any output generated by the script otherwise the cookie will not be set.

Syntax:

setcookie(name, value, expire, path, domain, security);

Parameters: The setcookie() function requires six arguments in general which are:

- Name: It is used to set the name of the cookie.
- Value: It is used to set the value of the cookie.
- Expire: It is used to set the expiry timestamp of the cookie after which the cookie can’t be accessed.
- Path: It is used to specify the path on the server for which the cookie will be available.
- Domain: It is used to specify the domain for which the cookie is available.
- Security: It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

Below are some operations that can be performed on Cookies in PHP:

- Creating Cookies: Creating a cookie named Auction_Item and assigning the value Luxury Car to it. The cookie will expire after 2 days(2 days * 24 hours * 60 mins * 60 seconds).
- For accessing a cookie value, the PHP $_COOKIE superglobal variable is used.

```
<!DOCTYPE html>
<?php
```

```
	setcookie("Auction_Item", "Luxury Car", time() + 2 * 24 * 60 * 60);
?>
<html>
<body>
	<?php
	if (isset($_COOKIE["Auction_Item"]))
	{
		echo "Auction Item is a " . $_COOKIE["Auction_Item"];
	}
	else
	{
		echo "No items for auction.";
	}
	?>
	<p>
		<strong>Note:</strong>
		You might have to reload the page
		to see the value of the cookie.
	</p>

</body>
</html>
```

Programs –

1. Create the login application using session management and cookies concept.
2. Create the student registration form to store the student information into the session variables and access it from session variables and cookies.

Results:

1. Thus, student must understand the concepts of session management and cookies.
2. Student should able to write the programs using session management and cookies.

Observations:

1. Student should able to study and write the session management for multiple web pages.

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 15

S/W Requirement – WAMP/LAMP/XAMPP

Title: Design and Develop an application for handling different database operations (Insert, Update, Display) using PHP.

.

Theory:

PHP MySQL Database

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after co-founder Monty Widenius's daughter: My

PHP 5 and later can work with a MySQL database using:

- **MySQLi extension** (the "i" stands for improved)

Connection with Database:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
```

```
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully"; ?>
```

Create a MySQL Table Using MySQLi

The **CREATE TABLE** statement is used to create a table in MySQL.

```
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";
```

Insert Data Into MySQL Using MySQLi

To insert data into a MySQL table, you would need to use the SQL INSERT INTO command. You can insert data into the MySQL table by using the mysql> prompt or by using any script like PHP.

some syntax rules to follow:

- The SQL query must be quoted in PHP
- String values inside the SQL query must be quoted
- Numeric values must not be quoted
- The word NULL must not be quoted

The INSERT INTO statement is used to add new records to a MySQL table:

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

Delete Data From a MySQL

If you want to delete a record from any MySQL table, then you can use the SQL command DELETE FROM.

DELETE FROM table_name WHERE some_column = some_value

Update Data In a MySQL

There may be a requirement where the existing data in a MySQL table needs to be modified. You can do so by using the SQL UPDATE command. This will modify any field value of any MySQL table.

UPDATE table_name SET column1=value, column2=value2,... WHERE some_column=some_value

For Example –

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
   echo "Database created successfully";
} else {
   echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
```

Programs –

1. Create the login application and store the data in the database.
2. Display the student data stored in database on the browser using the select Query

Results:

1. Thus, student must understand the concepts of creating the database and connecting the database with the application

Observations:

1. Student should able to study and implement the programs for implementing various database operations.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav

D. Y. Patil college of Engineering & Technology, Kolhapur

Department: Computer Science & Engg.

Subject: Web Technology Laboratory **Class: TY (A, B, C) (C.S.E.)-II**

Assignment No.: 16

S/W Requirement – WAMP/LAMP/XAMPP

Title: Design and develop the mini project for solving the different real time problems using Web Technologies in the group of 4-5 students.

.

Guidelines for Mini Project

- Step 1:
  Website development start with some types of a survey for Example Business details, Business goals, who are the Target audience and Competition.
- Step 2:
  Select the topic for your website and gather information about business.
- Step 3:
  After you have gathered information about the business create the sitemaps and wireframe. A sitemap is made with the information collected in the first stage. The aim of a site map is to create a website user-friendly and create a structure of a site. A wireframe provides a visual description of a site and decide what functions and feature you want in a site. This function includes login, email subscription, admin, live chat, and so much more.
- Step 4:
  The design phase introduces your branding and complete the look and feel of your website. The things to keep in mind while designing are a theme, color contract, where to place text, images, videos, etc. Create rough design of web pages before code writing starts.
- Step 5:
  After design phase select the platforms to developing website for example select front end like HTML, CSS, Javascript, React and back end language as PHP and start the code writing.
- Step 6:
  After completion of project check the functionality of all the elements on your website ensure everything works properly.
- Step 7:
  Display the results.

Results:

1. Thus, student must understand the concepts of mini project.

Observations:

1. Student should able to study and implement.

Prepared By:

Prof. K. T. Mane, Prof. A. S. Yadav, Prof. R. S. Jadhav