HTML

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

What is HTML?

HTML is the standard markup language for creating Web pages.

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

HTML Tags

HTML tags are element names surrounded by angle brackets:

<tagname>content here...</tagname>

- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- The end tag is written like the start tag, but with a forward slash inserted before the tag name

The <! DOCTYPE> Declaration

The <! DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.It must only appear once, at the top of the page (before any HTML tags).

The <! DOCTYPE> declaration is not case sensitive.

The <! DOCTYPE> declaration for HTML5 is: **<! DOCTYPE html>**

HTML Tags Chart

To use any of the following HTML tags, simply select the HTML code you'd like and copy and paste it into your web page.

Tag	Name	Code Example	Browser View
<!--	**comment**	**<!--**This can be viewed in the **HTML** part of a document**-->**	Nothing will show
<a -	**anchor**	** Visit Our Site****	Visit google
	bold	****Example****	**Example**
<big>	**big (text)**	**<big>**Example**</big>**	Example
<body>	**body of HTML document**	**<body>**The content of your **HTML** page**</body>**	Contents of your web page
 	line break	Keonics** **Hubli	Keonics Hubli
	center	** **This will center your contents** **	This will center your contents
<dd>	**definition description**	<dl> <dt>Definition Term</dt> **<dd>Definition of the term</dd>** <dt>Definition Term</dt> **<dd>Definition of the term</dd>** </dl>	Definition Term **Definition of the term** Definition Term **Definition of the term**
<dl>	**definition list**		

<dt>	**definition term**		
	emphasis	This is an ****Example**** of using the emphasis tag	This is an *Example* of using the emphasis tag
	font	**<font** face="Times New Roman" size="3" color="#ff0000">Example****	Example
<h1> <h2> <h3> <h4> <h5> <h6>	**heading 1** **heading 2** **heading 3** **heading 4** **heading 5** **heading 6**	** **Heading 1 Example** ** ** **Heading 2 Example** ** ** **Heading 3 Example** ** ** **Heading 4 Example** ** ** **Heading 5 Example** ** ** **Heading 6 Example** **	
<hr>	**horizontal rule**	**<hr />**	Contents of your web page Contents of your web page
<hr>	**horizontal rule**	**<hr** width="50%" size="3" />	Contents of your web page Contents of your web page
<hr>	**horizonta l rule**	**<hr** width="50%" size="3" noshade />	Contents of your web page Contents of your web page
<hr> (Internet Explorer)	**horizonta l rule**	**<hr** width="75%" color="#ff0000" size="4" />	Contents of your web page Contents of your web page
<html>	**hypertext markup language**	**<html>** <head> <meta> <title>Title of your web page</title> </head> <body>**HTML web page** contents </body> **</html>**	Contents of your web page
	italic	****Example****	*Example*
<img>	**image**	**<img** src="Earth.gif" width="41" height="41" border="0" alt="text describing the image" />	

1) <html>
<head>
<title>My First web page</title>
</head>
<body>
Welcome to HTML!
</body>
</html>

2) <html>
<head><title>back color</title> </head>
<body bgcolor="wheat">
Welcome to HTML!...
</body>
</html>

3) <html>
<head><title>text color</title></head>
<body text="red">
Welcome to HTML!...
</body>
</html>

4) <html>
<head><title>heading tags</title>
</head>
<body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
</body>
</html>

5) <html>
<head><title>font size tags</title>
</head>
<body>
Font size 1
Font size 2
Font size 3
Font size 4
Font size 5
Font size 6
Font size 7
</body>
</html>

6) <html>
<head><title>font color tags</title>
</head>
<body>
Font color is Red
Font color is Green
Font color is Blue
</body>
</html>

7) <html>
<head>
<title>font style tags</title>
</head>
<body>
this is Arial
This is Arial Black
This is Arial Narrow
This is Comics sans ms
This is Impact
This is Monotype Corsiva
This is Chiller
</body>
</html>

8) <html>
<head><title>Other font style tags</title></head>
<body>
Bold
Italic
<u>Underline</u>
<strike>Strike</strike>
<big>Big</big>
<small>Small</small>
Strong
<tt>Teletype</tt>
This is Maths 10²
We need H₂O
</body>
</html>

9) <html>
<head><title>Marked Text Example</title></head>
<body>
The <u>following </u><q>"word"</q> has b^{ee}n <mark>marked</mark> with ye_{ll}ow
</body>
</html>

10) <!DOCTYPE html>
<html>
<head> <!-- Document Header Starts -->
<title>This is document title</title>
</head> <!-- Document Header Ends -->
<body>
Document content goes here....observe use of comments in source code
</body>
</html>

11) <html>
<head>
<title>use of horizontal line</title>
</head>
<body>
<h1>This is heading 1</h1>
This is some text.
<hr>
<h2>This is heading 2</h2>
This is some other text.
<hr>
<h2>This is heading 3</h2>
This is some other text.
</body>
</html>

12) <html>
<body>
<h1 align="center">HTML</h1>
<hr color="red" align="center" width="100" size="10"/>
<h1 align="center">CSS</h1>
<hr color="blue" align="center" width="100" size="10"/>
<h1 align="center">JAVASCRIPT</h1>
<hr color="green" align="center" width="200"
size="10"/>
</body>
</html>

13) <html>
<head><title>Marked Text Example</title></head>
<body>
The following word has been
<mark>marked</mark> with yellow
</body>
</html>

14) <html>
<head><title>Paragraph tags</title>
</head>
<body>
Paragraph 1
Paragraph 2
Paragraph 3
</body>
</html>

15) <html>
<head>
<title>Paragraph tags</title></head>
<body>
Video provides a powerful way to
help you prove your point.1 <hr>
Video provides a powerful way to
help you prove your point. 2
Video provides a powerful way to
help you prove your point. 3
Video provides a powerful way to
help you prove your point. 3
</body>
</html>

16) <html>
<head>
<title>Paragraph with alignment tags</title></head>
<body>
Paragraph 1
Paragraph 2
Paragraph 3
</body>
</html>

17) <html>
<head>
<title>other Paragraph alignment</title>
</head>
<body>
displays some portion of text inline
<blockquote><h1> It should contain only block-level
elements within it, and not just plain
text.</h1></blockquote>
displays some portion of text block
</body>
</html>

18) <html>
<head><title>My web page</title></head>
<body>
<pre>
KEONICS COMPUTER TRAINING CENTRE
IT PARK
HOSUR
HUBBALLI
CONTACT:0836-2367675
</pre>
</body>
</html>

19) <html>
<head><title>order listing</title></head>
<body>
<H2>ORDERED LIST</H2>

C
C++
Java

<ol type="I">
C
C++
Java

<ol type="i">
C
C++
Java

<ol type="A">
C
C++
Java

<ol type="a">
C
C++
Java

</body>
</html>

20) <html>
<head>
<title>Unorder listing</title></head>
<body>
<H2>UNORDERED LIST</H2>

C
C++
Java

<ul type="square">
C
C++
Java

<ul type="circle">
C
C++

```
        <li>Java</li>
    </ul>
    <ul type="disc">
        <li>C</li>
        <li>C++</li>
        <li>Java</li>
    </ul>
  </body>
</html>
```

```
    21) <html>
   <head><title>Mixed listing</title></head>
  <body>
      <ol>
    <li>Programming subjects
    <ul type="square">
        <li>C</li>
        <li>C++</li>
        <li>Java</li>
    </ul>
    <li>DTP subjects
    <ul type="circle">
        <li>pagemaker</li>
        <li>Corel Draw</li>
        <li>Photoshop</li>
    </ul>
   </li>
  </li>
  </body>
</html>
```

```
    22) <html>
        <head><title>definition list</title></head>
<body>
<h2>Definition List</h2>
        <dl>
        <dt>COMPUTER</dt>
<dd>Computer is an electronic device accepts data from
user and process the data ,gives you required result as
output</dd> </dt>  </dl> <dl>
<dt>What is HTML?</dt><dd>HTML is the standard
markup language for creating     Web pages.
        </dd>   </dt>    </dl>
</body>
</html>
```

```
    23) <html>
   <head><title>Image</title></head>
  <body>
    <img src="1.jpg">
  </body>
</html>
```

```
    24) <html>
   <head> <title>Image</title></head>
  <body>
    <img src="3.jpg" height="300" width="300">
  </body>
</html>
```

```
    25) <html>
   <head> <title>Image </title> </head>
  <body>
    <img src="5.jpg" height="300" width="400"
border="4">
  </body> </html>
```

```
    26) <html>
   <head> <title>Background Image </title> </head>
  <body background="1.jpg" text="#4d0000">
     <center><h1>KEONICS COMPUTER
CENTRE</h1></center>
<center><h1>IT PARK</h1></center>
<center><h1>HOSUR HUBLI</h1></center>
  </body>
</html>
```

```
    27) <html>
   <head> <title>MOVING text</title></head>
  <body>
      <marquee direction="right" bgcolor="orange">
       <h1>KEONICS COMPUTER CENTRE</h1>
      </marquee>
  </body>
</html>
```

```
    28) <html>
   <head> <title>MOVING Image</title></head>
  <body>
      <marquee>
         <img src="1.jpg" height=150 width=150>
         <img src="2.jpg" height=150 width=150>
         <img src="3.jpg" height=150 width=150>
         <img src="4.jpg" height=150 width=150>
      </marquee>
  </body>
</html>
```

```
    29) <html>
   <head> <title>MOVING Image</title></head>
  <body>
      <marquee bgcolor=#00bfff behavior="alternate">
         <img src="2.jpg" height=150 width=150>
         <img src="3.jpg" height=150 width=150>
         <img src="4.jpg" height=150 width=150>
         <img src="5.jpg" height=150 width=150>
      </marquee>
  </body>
</html>
```

```
    30) <html>
   <head> <title>MOVING Image</title></head>
  <body>
      <marquee bgcolor="pink" direction="up"
behavior="alternate" width="150" height="720">
         <img src="2.jpg" height=150 width=150>
         <img src="3.jpg" height=150 width=150>
         <img src="4.jpg" height=150 width=150>
         <img src="5.jpg" height=150 width=150>
      </marquee>
  </body>
</html>
```

```
    31)  <html>
   <head> <title>linking page</title></head>
  <body>
      <a href="2.jpg">Click here To View Computer</a>
  </body>
</html>
```

```
    32) <html>
   <head><title>linking page</title></head>
  <body>
```

```
      <a href="3.jpg"><img src="1.jpg" height="100"
width="100"></a>
  </body>
</html>
```

33)
```
<html>
  <head>
    <title>linking page</title></head>
  <body>
      <a href="2.html">Click here</a><br>
      <a href="file1.docx">My Word File</a><br>
      <a href="file1.xlsx">My Excel File</a><br>
      <a href="file1.pptx">My Powerpoint File</a>
  </body>
</html>
```

34)
```
<html>
<body>
<h1>html table display</h1>
<table border="2">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Naveen</td>
    <td>Vernekar</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Nikhil</td>
    <td>Shet</td>
    <td>25</td>
  </tr>
  <tr>
    <td>Neha</td>
    <td>Rao</td>
    <td>40</td>
  </tr>
</table>
</body>
</html>
```

35)
```
<html>
<body>
<h2>Cell that spans two columns</h2>
<p>To make a cell span more than one column, use the
colspan attribute.</p>
<table border="2" width="300" height="200">
  <tr >
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr align="center">
    <td>Anil</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
<tr align="center">
    <td>Raj</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
</body>
</html>
```

36)
```
<html>
<head>
<body>
<h2>Cell that spans two rows</h2>
<p>To make a cell span more than one row, use the
rowspan attribute.</p>
<table border="2" bgcolor="cyan" bordercolor="maroon"
width="300" height="200">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
</body>
</html>
```

37)
```
<html>
   <head><title>tables</title> </head>
  <body bgcolor="grey" text="white">
       <table border="1" cellspacing="10"
cellpadding="10" bordercolor="blue"
background="4.jpg" align="center">
          <Caption ><h1>STUDENT
DETAILS</h1></caption>
          <tr align="center" bgcolor="purple">
             <th  width="150">Roll No</th>
             <th width="250">Name of the Student</th>
          </tr>
          <tr  align="center" bgcolor="red">
             <td>101</td>
             <td>Ajay Jain</td>
          </tr>
          <tr  align="center" bgcolor="green">
             <td>102</td>
             <td>Bharat Jain</td>
          </tr>
          <tr bgcolor="blue">
             <th colspan="2"><marquee
behavior="alternate" >KEONICS
COMPUTER</marquee></th>
          </tr>
       </table>
  </body>
</html>
```

38)
```
<html>
      <head> <title>Label Text</title></head>
 <body>
<form>
<table align="center">
<tr><th>
   <label>Enter Your Name</label></th>
<td> <input type="text" placeholder="enter
name"/></td>
</tr>
```

```
<tr>
<th>
 <label> Contact Number</label></th>
<td>
   <input type="text" placeholder="contact
number"/></td>
</tr>
</table>
</form>
</body>
</html>
```

39)
```
<html>
      <head><title>password</title></head>
 <body>
<form>
<table align="center">
<tr>
<th>
  <label>Enter Your Name</label></th>
<td>   <input type="text" placeholder="enter
name"/></td>
</tr>
<tr>
<th>
 <label> Password</label></th>
<td>   <input type="password" placeholder="enter
password" /></td>
</tr>
</table>
</form>
</body>
</html>
```

40)
```
<html>
      <head> <title>checkbox</title></head>
 <body>
<form>
<table align="center">
<tr>
<th>   <label>Languages Known By You</label></th>
  <td><input type="checkbox" checked>Kannada <br>
  <input type="checkbox">English <br>
 <input type="checkbox" >Hindi </td>
</table>
</form>
</body>
</html>
```

41)
```
<html>
      <head> <title>radio button</title>
      </head>
 <body>
<form>
<table align="center">
<tr>
<th>
  <label>Select Gender</label></th>
<td>
  <input type="radio" name="gr" checked>Male <br>
  <input type="radio" name="gr" >Female <br>
  <input type="radio" name="gr" >Other <br>
</td>
</tr>
</table>
</form>
</body>
</html>
```

42)
```
<html>
      <head><title>Drop Down List</title></head>
 <body>
<form>
<table align="center">
<tr><td>
  <label>Select Your State</label></td>
<td>
  <select>
<option selected>Karnataka</option>
 <option>gujarat</option>
<option>Kerala</option>
<option>Maharashtra</option>
<option>Goa</option>
</select>
</td>
</tr>
</table>
</form>
</body>
</html>
```

43)
```
<html>
  <head><title>Login page</title></head>
 <body>
    <table border="1" bgcolor="pink">
      <tr>
        <th colspan="3">Login Up </th>
      </tr>
      <tr>
        <th>User Name</th>
            <td><input type="text"></td>
      </tr>
      <tr>
        <th>Password</th>
<td><input type="password" maxlength="8"></td>
      </tr>
      <tr>          <tr>
            <td colspan="2" align="center">
            <input type="Submit" value="Login">
            <input type="Reset" value="Clear">
          </td>
      </tr>
    </table>
 </body>
</html>
```

44)
```
<html>
 <head> <title>Sign up page</title></head>
 <body>
    <table border="1" bgcolor="pink">
      <tr>
        <th colspan="2">Sign Up </th>
      </tr>
      <tr>
        <th>First Name</th>
            <td><input type="text"></td>
      </tr>
      <tr>
        <th>Middle Name</th>
            <td><input type="text"></td>
      </tr>
      <tr>
```

```
            <th>Last Name</th>
                  <td><input type="text"></td>
         </tr>
         <tr>
            <th>Address</th>
                  <td>
<TextArea>hubli</TextArea>
</td>
         </tr>
         <tr>
            <th>Gender</th>
                  <td>
                  <input type="radio" name="gr1">Male
                  <input type="radio" name="gr1">Female
                  </td>
         </tr>
         <tr>
            <th>Language Known</th>
                  <td>
                  <input type="Checkbox">Kannada <br>
                  <input type="Checkbox">Hindi<br>
                  <input type="Checkbox">English<br>
                  </td>
         </tr>
         <tr>
            <th>Select Country</th>
                  <td>
                  <Select>
                    <option>India</option>
                    <option>Aus</option>
                    <option>England</option>
                        <option>USA</option>
                  </Select>
                  </td>
         </tr>
         <tr>
<td colspan="2" align="center">
                  <input type="Submit" value="Save">
                  <input type="Reset" value="Clear">
                </td>
  </tr>
      </table>
  </body>
</html>
```

45)
```
<frameset rows="20%,40%,20%,20%">
  <frame src="2.html">
  <frame src="4.html">
  <frame src="6.html">
 <frame src="8.html">
</frameset>
```

46)
```
<frameset cols="20%,20%,30%,30%">
  <frame src="29.html">
 <frame src="36.html">
  <frame src="25.html">
  <frame src="26.html">
 </frameset>
```

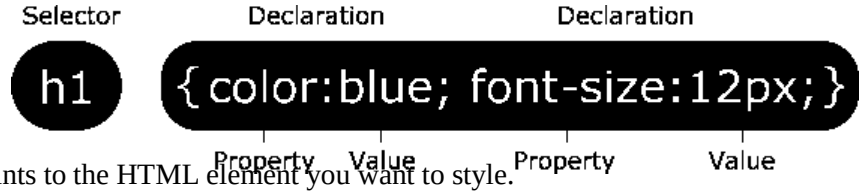
47)
```
<html>
  <head>
     <title>
          iframe
     </title>
  </head>
  <body>
     <h1>KEONICS</h1>
    <iframe src="25.html" align="right">

  </body>
</html>
```

48)
```
<html>
  <head>
     <title>
          iframe
     </title>
  </head>
  <body>
     <h1>KEONICS</h1>
    <iframe src="29.html"  align="right" height="500"
width="180"></iframe>
    <iframe src="25.html" align="bottom"  align="left"
height="200" width="300">
  </body>
</html>
```

49)
```
<html>
<head>
<title>mailto</title>
</html>
<body>
<a href="mailto:xyz123@gmail.com ">Mail Me</a>
</body>
</html>
```

What is CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External style sheets are stored in **CSS files**
- CSS Syntax
- A CSS rule-set consists of a selector and a declaration block:



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.

- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.
- In the following example all <p> elements will be center-aligned, with a red text color:

The element Selector

The element selector selects elements based on the element name.
You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

Example

```
p {
   text-align: center;
   color: red;
}
```

The id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by the id of the element.
The style rule below will be applied to the HTML element with id="para1":

Example

```
#para1 {
   text-align: center;
   color: red;
}
```

The class Selector

The class selector selects elements with a specific class attribute.
To select elements with a specific class, write a period (.) character, followed by the name of the class.
In the example below, all HTML elements with class="center" will be red and center-aligned:

Example

```
.center {
   text-align: center;
   color: red;
}
```

Grouping Selectors

If you have elements with the same style definitions, like this:

```
h1 {    text-align: center;
        color: red;  }
h2 {      text-align: center;
          color: red; }
p {       text-align: center;
         color: red;}
```

It will be better to group the selectors, to minimize the code.
To group selectors, separate each selector with a comma.
In the example below we have grouped the selectors from the code above:

Example

```
h1, h2, p {
    text-align: center;color: red; }
```

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date. Commentsare ignored by browsers.

A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

Example

```
p
{
    color: red;
    /* This is a single-line
    comment */text-align:
    center;}
/* This isa
multi-line
comment */
```

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

External Style Sheet

With an external style sheet, you can change the look of an entire website by changing just one file!

Each page must include a reference to the external style sheet file inside the <link> element. The <link> elementgoes inside the <head> section:

Example

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

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheetfile must be saved with a **.css extension.**

Here is how the "**mystyle.css"**

looks:

```
body {
    background-color: lightblue;
}
h1 {
    color: navy; margin-left: 20px;
}
```

Internal Style Sheet

An internal style sheet may be used if one single page has a unique style.

Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

Example

```
<head>
<style> body {
    background-color: linen;
}
h1 {
    color: maroon; margin-left: 40px;
}
</style>
</head>
```

Inline Styles

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

The example below shows how to change the color and the left margin of a <h1> element:

Example

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment starts with /* and ends with */. Comments can also span multiple lines:

```
Example p {color: red;
    /* This is a single-line comment */text-align: center;}
/* This is a multi-line comment */
```

1.css_ways

1) <html>
<head> <style>
/* illustration of inline css */
</style><title>style attribute</title>
</head>
<body>
<h1 style="font-family:Algerian;color:green;">WELCOME TO CSS</h1>
Paragraph 1
paragraph2
</body>
</html>

2) <html>
<head>
<title>My First web page</title>
</head>
<body STYLE="background-color:cyan;">
Welcome to CSS!...
<H1 STYLE="background-image:url(3.JPG);color:yellow;">IMAGE BACKGROUND</H1>
<h2 STYLE="background-color:FFDDFF;">COLOR BACKGROUND</H2>
</body>
</html>

3) <HTML>
<head> <style>
/* illustration of internal css and class selector */
.mykeonics
{ color:red;
font-size:60; }
.mykeonics1
{ background-color:cyan; }
</style></head>
<body>
using my class
<h1 class="mykeonics">heading1</h1>
bold text
<h2 class="mykeonics1">heading2</h2>
</body> </html>

4) <html>
<head>
<style>
/* Illustration of internal css anf id selector */
#myid
{ color:red;
font-size:40; }
#myid1
{ text-align:center; }
</style></head>
<body>
using my id
<h1 id="myid1">using my id1</h1>
</body>
</html>

5) <html>
<head>
<style>
/* illustration of internal css anf grouping selector */
p,b,i
{ color:red;
font-size:40; }
h2,h3
{ text-align:center; }
h1
{ text-align:right;
color:453628;
font-size:50; }
</style></head>
<body>
my paragraph <hr/>
bold text<hr/>
italic text<hr/>
<h1>heading1</h1>
<h2>heading2</h2>
<h3>heading3</h3>
</body>
</html>

2.css_text

6) <html>
<head><style>
h1 { text-align: center;}
h2 { text-align: left;}
h3 { text-align: right;}
</style></head>
<body>
<h1>Heading 1 (center)</h1>
<h2>Heading 2 (left)</h2>
<h3>Heading 3 (right)</h3>
The three headings above are aligned center, left and right.
</body></html>

7) <html>
<head><style>
h1 { text-decoration: overline;}
h2 { text-decoration: line-through;}
h3 { text-decoration: underline;}
</style></head><body>
<h1>This is heading 1(overline)</h1>
<h2>This is heading 2(line through)</h2>
<h3>This is heading 3(underline)</h3>
</body></html>

8) <html>
<head><style>
p.uppercase {text-transform: uppercase; }
p.lowercase {text-transform: lowercase; }
p.capitalize {text-transform: capitalize; }
</style></head><body>
This is some text.
This is some text.
This is some text.
</body></html>

9) <html>
<head><style>
h1 { letter-spacing: 3px; }
p{ word-spacing: 20px;
text-shadow:5px 6px red;}
h2 { letter-spacing: -3px; }
</style></head><body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
computer is an electronic device
</body></html>

10) <html>
<head><style>
p.small {line-height: 0.7; }
p.big { line-height: 2.8; }
</style></head><body>
This is a paragraph with a standard line-height.
The default line height in most browsers is about 110% to 120%.

This is a paragraph with a smaller line-height.
This is a paragraph with a smaller line-height.

This is a paragraph with a bigger line-height.
This is a paragraph with a bigger line-height.
</body></html>

11) <html>
<head><style>
p.ex1 { direction: rtl;}
</style></head><body>
This is the default text direction.
This is right-to-left text direction, which begins from right
</body></html>

12) <html>
<head><style>
h1 { text-shadow: 5px 4px red;}
</style> </head> <body>
<h1>Text-shadow effect</h1>
Note: Internet Explorer 9 and earlier do not support the text-shadow property.
</body> </html>

13) <html>
<head><style>
p { text-indent: 50px; }
</style></head><body>
In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.' In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.' In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'
</body></html>

14) <html> <head> <style>
h1{ color:Crimson ; /* any color value*/
letter-spacing:3px; /* any pixel value */
word-spacing:5px; /* any pixel value */
text-align:center; /* left center right justify */
text-decoration:underline;/* overline line-through*/
text-transform:capitalize; /* uppercase lowercase */
text-shadow: 3px 3px Gold ; /* horizontal_shadow vertical_shadow color */
h2{ text-indent: 25px;
direction:ltr; /* rtl */}
</style></head><body>
<h1>KEONICS COMPUTER TRAINING CENTRE</h1>
<h2> IT PARK HUBLI</h2>
</body> </html>

3.css_font

15) <html><head><style>
p.impact { font-family: IMPACT;}
p.an{ font-family: Arial NARROW;}
</style></head><body>
<h1>CSS font-family</h1>
This is a paragraph, shown impact font.
This is a paragraph, shown in the Arial Narrow font.
</body></html>

15) <html>
<head> <style>
h1 { font-size: 250%;}
h2 { font-size: 200%;}
p { font-size: 300%;}
h3{ font-size: 40px;}
</style></head><body>
<h1>This is heading 1 size 250%</h1>
<h2>This is heading 2 size 200%</h2>
This is a paragraph size 300%
<h3>This is heading 3 size 40px</h3>
</body></html>

16) <html>
<head> <style>
p.normal { font-style: normal;

```
}
p.italic {
   font-style: italic;
}
p.oblique {
   font-style: oblique;
}
</style>
</head>
<body>
<p class="normal">This is a paragraph, normal.</p>
<p class="italic">This is a paragraph, italic.</p>
<p class="oblique">This is a paragraph, oblique.</p>
</body> </html>
```

17)
```
<html>
<head> <style>
p.normal {font-variant: normal;}
p.small {    font-variant: small-caps;}
</style></head><body>
<p class="normal">My name is AMIT PATIL.</p>
<p class="small">My name is AMIT PATIL.</p>
</body></html>
```

18)
```
<html>
<head> <style>
p.normal { font-weight: normal;}
p.light {  font-weight: lighter;}
p.thick {    font-weight: bold;}
p.thicker {    font-weight: 900;}
</style></head><body>
<p class="normal">This is a paragraph. normal</p>
<p class="light">This is a paragraph. light</p>
<p class="thick">This is a paragraph . thick</p>
<p class="thicker">This is a paragraph. thicker</p>
</body> </html>
```

4.css_color

19)
```
<html> <body>
<h1 style="background-color:Tomato;">Tomato</h1>
<h1 style="background-color:Orange;">Orange</h1>
<h1 style="background-color:Blue;">Blue</h1>
<h1 style="background-
color:Green;">MediumSeaGreen</h1>
<h1 style="background-color:Gray;">Gray</h1>
<h1 style="background-color:SlateBlue;">Blue</h1>
<h1 style="background-color:Violet;">Violet</h1>
<h1 style="background-
color:LightGray;">LightGray</h1>
</body></html>
```

20)
```
<html> <body>
<h3 style="color:Tomato;">Hello World</h3>
<p style="color:Blue;">You can set the color of
text</p> <p style="color:Green;">You can set the
color of text</p> </body> </html>
```

21)
```
<html> <body>
<h1 style="border: 5px solid Tomato;">Hello
World</h1>
<h1 style="border: 4px solid DodgerBlue;">Hello
World</h1>
<h1 style="border: 3px solid Violet;">Hello
World</h1>
</body> </html>
```

22)
```
<html> <body>
<p>Same as color name "Tomato":</p>
<h1 style="background-color:rgb(255, 99,
71);">rgb(255, 99, 71)</h1> <h1 style="background-
color:#ff6347;">#ff6347</h1> <h1
style="background-color:hsl(9, 100%, 64%);">hsl(9,
100%, 64%)</h1> <p>Same as color name "Tomato",
but 50% transparent:</p><h1 style="background-
color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)-
><html><body>
<p>Same as color name "Tomato":</p>
<h1 style="background-color:rgb(255, 99,
71);">rgb(255, 99, 71)</h1>
<h1 style="background-color:#ff6347;">#ff6347</h1>
<h1 style="background-color:hsl(9, 100%,
64%);">hsl(9, 100%, 64%)</h1>
<p>Same as color name "Tomato", but 50%
transparent:</p><h1 style="background-
color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71,
0.5)</h1><h1 style="background-color:hsla(9, 100%,
64%, 0.5);">hsla(9, 100%, 64%, 0.5)HSL Colors
</h1><p>In addition to the predefined color names,
colors can be specified using RGB, HEX, HSL, or
even transparent colors using RGBA or HSLA color
values.</p></body></html>
</h1><h1 style="background-color:hsla(9, 100%,
64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h1>
<p>In addition to the predefined color names, colors
can be specified using RGB, HEX, HSL, or even
transparent colors using RGBA or HSLA color
values.</p> </body> </html>
```

23)
```
<html> <head> <style>
#p1 {background-color:rgba(255,0,0,0.3);}
#p2 {background-color:rgba(0,255,0,0.3);}
#p3 {background-color:rgba(0,0,255,0.3);}
#p4 {background-color:rgba(192,192,192,0.3);}
#p5 {background-color:rgba(255,255,0,0.3);}
#p6 {background-color:rgba(255,0,255,0.3);}
</style> </head> <body>
<p>RGB colors with opacity:</p>
<p id="p1">Red</p>
<p id="p2">Green</p>
<p id="p3">Blue</p>
<p id="p4">Grey</p>
<p id="p5">Yellow</p>
<p id="p6">Cerise</p>
</body> </html>
```

24)
```
<html> <head> <style>
#p1 {background-color:hsl(120,100%,50%);}
#p2 {background-color:hsl(120,100%,75%);}
#p3 {background-color:hsl(120,100%,25%);}
#p4 {background-color:hsl(120,60%,70%);}
#p5 {background-color:hsl(290,100%,50%);}
#p6 {background-color:hsl(290,60%,70%);}
</style> </head>
```

```
<body>
<p>HSL colors:</p>
<p id="p1">Green</p>
<p id="p2">Light green</p>
<p id="p3">Dark green</p>
<p id="p4">Pastel green</p>
<p id="p5">Violet</p>
<p id="p6">Pastel violet</p>
</body> </html>
```

25) <html> <head> <style>

```
#p1 {background-color:hsla(120,100%,50%,0.3);}
#p2 {background-color:hsla(120,100%,75%,0.3);}
#p3 {background-color:hsla(120,100%,25%,0.3);}
#p4 {background-color:hsla(120,60%,70%,0.3);}
#p5 {background-color:hsla(290,100%,50%,0.3);}
#p6 {background-color:hsla(290,60%,70%,0.3);}
</style> </head> <body>
<p>HSL colors with opacity:</p>
<p id="p1">Green</p>
<p id="p2">Light green</p>
<p id="p3">Dark green</p>
<p id="p4">Pastel green</p>
<p id="p5">Violet</p>
<p id="p6">Pastel violet</p>
</body> </html>
```

5.css_background

26) <html> <body>

```
<h2 style="background-color:red;color:white">
Red background-color
</h2> <h2 style="background-color:green;color:blue">
Green background-color </h2> <h2
style="background-color:blue;color:FFFFFF">
Blue background-color and white text color
</h2> <h2 style="background-color:orange">
Orange background-color </h2>
<h2 style="background-color:yellow">
Yellow background-color </h2>
<h2 style="background-color:CYAN">
Cyan background-color </h2>
<h2 style="background-color:black;color:yellow">
Black background-color and yellow text color
</h2> <h2 style="background-color:black;color:blue">
Black background-color and purple text color </h2>
</body> </html>
```

27) <html> <head> <style>

```
body {    background-image: url("1.jpg");
   background-repeat:repeat;        /* no-repeat */
   background-position: left top;
background-attachment: fixed;   /* scroll*/}
</style> </head> <body>
<h1>Hello World!</h1> <p>The background-image is
fixed. Try to scroll down the page.</p> <p>The
background-image is fixed. Try to scroll down the
page.</p> <p>The background-image is fixed. Try to
scroll down the page.</p>  <p>The background-image
is fixed. Try to scroll down the page.</p> <p>The
background-image is fixed. Try to scroll down the
page.</p> <p>The background-image is fixed. Try to
scroll down the page.</p> <p>The background-image
is fixed. Try to scroll down the page.</p>  <p>The
background-image is fixed. Try to scroll down the
page.</p> <p>The background-image is fixed. Try to
scroll down the page.</p> <p>The background-image
is fixed. Try to scroll down the page.</p> <p>The
background-image is fixed. Try to scroll down the
page.</p> <p>The background-image is fixed. Try to
scroll down the page.</p> <p>The background-image
is fixed. Try to scroll down the page.</p>  <p>The
background-image is fixed. Try to scroll down the
page.</p> <p>The background-image is fixed. Try to
scroll down the page.</p> <p>The background-image
is fixed. Try to scroll down the page.</p> <p>The
background-image is fixed. Try to scroll down the
page.</p> <p>The background-image is fixed. Try to
scroll down the page.</p><p>The background-image
is fixed. Try to scroll down the page.</p><p>The
background-image is fixed. Try to scroll down the
page.</p><p>If you do not see any scrollbars, try to
resize the browser window.</p></body></html>
```

28) <html>

```
<head>
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style>
</head>
<body>
<h2>The border-style Property</h2>
<p>This property specifies what kind of border to
display:</p>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>
</body>
</html>
```

JavaScript

What is JavaScript? JavaScript – Overview

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.
It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

Advantages of JavaScript

The merits of using JavaScript are −

- **Less server interaction** − You can validate user input before sending the page off to the server. Thissaves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they haveforgotten to enter something.
- **Increased interactivity** − You can create interfaces that react when the user hovers over them with amouse or activates them via the keyboard.
- **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components andsliders to give a Rich Interface to your site visitors.

JavaScript is the most popular and widely used client-side scripting language. Client-side scripting refers to scripts that run within your web browser. JavaScript is designed to add interactivity and dynamic effects to the web pages by manipulating the content returned from a web server

JavaScript was originally developed as **LiveScript by Netscape** in the mid 1990s. It was later renamedto JavaScript in 1995, and became an ECMA standard in 1997. Now JavaScript is the standard client- side scripting language for web-based applications, and it is supported by virtually all web browsers available today, such as Google Chrome, Mozilla Firefox, Apple Safari, etc.

JavaScript is officially maintained by ECMA (European Computer Manufacturers Association) as ECMAScript. ECMAScript 6 (or ES6) is the latest major version of the ECMAScript standard.

What You Can Do with JavaScript

- You can modify the content of a web page by adding or removing elements.
- You can change the style and position of the elements on a web page.
- You can monitor events like mouse click, hover, etc. and react to it.
- You can perform and control transitions and animations.
- You can create alert pop-ups to display info or warning messages to the user.
- You can perform operations based on user inputs and display the results.
- You can validate user inputs before submitting it to the server.

Example 1 Embedding
JavaScript

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Embedding JavaScript</title>
</head>
<body>
    <script>
    var greet = "Hello World!";
    document.write(greet); // Prints: Hello World!
    </script>
</body>
</html>
```

Example 2 Externatl
JavaScript

Hello.js

```
// A function to display a message
  function sayHello() {
     alert("Hello World!");
}

// Call function on click of the button
document.getElementById("myBtn").onclick = sayHello;
```

pro2.html

```
<!DOCTYPE html>
<html lang="en">
<head>
     <meta charset="UTF-8">
     <title>Including External JavaScript File</title>
</head>
<body>
     <button type="button" id="myBtn">Click Me</button>
     <script src="js/hello.js"></script>
</body>
</html>
```

Example 3

Placing JavaScript Inline

```
<!DOCTYPE html>
<html lang="en">
<head>
     <meta charset="UTF-8">
     <title>Inlining JavaScript</title>
</head>
<body>
     <button onclick="alert('Hello World!')">Click Me</button>
</body>
</html>
```

Difference between Client-side and Server-side Scripting

Client-side scripting languages such as JavaScript, VBScript, etc. are interpreted and executed by theweb browser, while server-side scripting languages such as PHP, ASP, Java, Python, Ruby, etc. runs on the web server and the output sent back to the web browser in HTML format.

Client-side scripting has many advantages over traditional server-side scripting approach. For example, you can use JavaScript to check if the user has entered invalid data in form fields and shownotifications for input errors accordingly in real-time before submitting the form to the web-server for final data validation and further processing in order to prevent unnecessary network bandwidth usages and the exploitation of server system resources.

JavaScript Syntax

A JavaScript consists of JavaScript statements that are placed within the <script></script> HTML tagsin a web page, or within the external JavaScript file having .js extension.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
     <meta charset="utf-8">
     <title>Example of JavaScript Statements</title>
</head>
<body>
     <script> var x = 5; var y = 10;
     var sum = x + y;
     document.write(sum); // Prints variable value
     </script>
</body>
</html>
```

JavaScript - Syntax

JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>** HTML tags in a web page.

```
<script ...>
  JavaScript code
</script>
```

You can place the **<script>** tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the **<head>** tags.
The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. Asimple syntax of your JavaScript will appear as follows.

```
<script language = "javascript" type = "text/javascript">
  JavaScript code
</script>
```

The script tag takes two important attributes −

- **Language** − This attribute specifies what scripting language you are using.
- **Type** − This attribute is what is now recommended to indicate the scripting language in use and its valueshould be set to **"text/javascript".**

First JavaScript Code

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

Comments in JavaScript

JavaScript supports both C-style and C++-style comments, Thus −

- Any text between a // and the end of a line is treated as a comment and is ignored by JavaScript.
- Any text between the characters /* and */ is treated as a comment. This may span multiple lines.
- JavaScript also recognizes the HTML comment opening sequence <!--. JavaScript treats this as a single-line comment, just as it does the // comment.
- The HTML comment closing sequence --> is not recognized by JavaScript so it should be written as //-->.
- JavaScript Variables

```
<script type = "text/javascript">
    var name;
    var address;
</script>
```

Note − Use the **var** keyword only for declaration or initialization, once for the life of any variable name in adocument. You should not re-declare same variable twice.

JavaScript Variable Scope

The scope of a variable is the region of your program in which it is defined. JavaScript variables have only twoscopes.

- **Global Variables** − A global variable has global scope which means it can be defined anywhere in yourJavaScript code.
- **Local Variables** − A local variable will be visible only within a function where it is defined. Functionparameters are always local to that function.

JavaScript Variable Names

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

- You should not use any of the JavaScript reserved keywords as a variable name.
- JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or anunderscore character. For example, **123test** is an invalid variable name but **_123test** is a valid one.
- JavaScript variable names are case-sensitive. For example, **Name** and **name** are two different variables.

JavaScript – Operators

JavaScript supports the following types of operators.

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators
- Conditional (or ternary) Operators

Case Sensitivity in JavaScript

JavaScript is case-sensitive. This means that variables, language keywords, function names, and other identifiers must always be typed with a consistent capitalization of letters.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Case Sensitivity</title>
</head>
<body>
    <script>
    var myVar = "Hello World!";
    console.log(myVar);
    console.log(MyVar);
    console.log(myvar);
    </script>
   <p><strong>Note:</strong> Check out the browser console by pressing the f12 key on the keyboard, you'll see a line something like this: "Uncaught ReferenceError: MyVar is not defined".</p>
</body>
</html>
```

JavaScript Variables

What is Variable?

Variables are fundamental to all programming languages. Variables are used to store data, like stringof text, numbers, etc. The data or value stored in the variables can be set, updated, and retrieved whenever needed. In general, variables are symbolic names for values.

You can create a variable with the var keyword, whereas the assignment operator (=) is used to assign value to a variable, like this: var varName = value;

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Creating Variables in JavaScript</title>
</head>
<body>
    <script>
    // Creating variables
    var name = "Peter Parker";
    var age = 21;
    var isMarried = false;

    // Printing variable values
    document.write(name + "<br>");
    document.write(age + "<br>");
    document.write(isMarried);
    </script>
</body>
</html>
```

Example 2

```
        <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Declaring Variables in JavaScript</title>
</head>
<body>
    <script>
```

```
    // Declaring Variable
    var userName;

    // Assigning value
    userName = "Clark Kent";
    // Printing variable values
    document.write(userName);
    </script>
</body>
</html>
```

Example 3

```
  <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Declaring Multiple Variables in JavaScript</title>
</head>
<body>
    <script>
    // Declaring multiple Variables
    var name = "Peter Parker", age = 21, isMarried = false;

    // Printing variable values
    document.write(name + "<br>");
    document.write(age + "<br>");
    document.write(isMarried);
    </script>
</body>
</html>
```

The let and const Keywords (ES6)

ES6 introduces two new keywords let and const for declaring variables.

The const keyword works exactly the same as let, except that variables declared using const keyword cannot be reassigned later in the code.

Example

```
  <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Declaring Variables with let and const Keywords in JavaScript</title>
</head>
<body>
    <script>
    // Declaring variables

    let name = "Harry Potter";let
    age = 11;
    let isStudent = true;

    // Printing variable values
    document.write(name + "<br>");
    document.write(age + "<br>");
    document.write(isStudent + "<br>");

    // Declaring constant
    const PI = 3.14;

    // Printing constant value
    document.write(PI); // 3.14
```

```
    // Trying to reassignPI
    = 10; // error
    </script>
  <p><strong>Note:</strong> Please check out the browser console by pressing the f12 key on the
keyboard.</p>
</body>
</html>
```

JavaScript Generating Output

Generating Output in JavaScript

There are certain situations in which you may need to generate output from your JavaScript code. For example, you might want to see the value of variable, or write a message to browser console tohelp you debug an issue in your running JavaScript code, and so on.
In JavaScript there are several different ways of generating output including writing output to the browser window or browser console, displaying output in dialog boxes, writing output into an HTML element, etc.

Writing Output to Browser Console

You can easily outputs a message or writes data to the browser console using the console.log() method.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Writing into the Browser's Console with JavaScript</title>
</head>
<body>
    <script>
    // Printing a simple text message console.log("Hello
    World!"); // Prints: Hello World!

    // Printing a variable value
    var x = 10;
    var y = 20;
    var sum = x + y;
    console.log(sum); // Prints: 30
    </script>
  <p><strong>Note:</strong> Please check out the browser console by pressing the f12 key on the
keyboard.</p>
</body>
</html>
```

Displaying Output in Alert Dialog Boxes

You can also use alert dialog boxes to display the message or output data to the user. An alert dialogbox is created using the alert() method.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Writing into an Alert Dialog Box with JavaScript</title>
</head>


<body>
    <script>
    // Displaying a simple text message alert("Hello
    World!"); // Outputs: Hello World!

    // Displaying a variable value
    var x = 10;
    var y = 20;
    var sum = x + y; alert(sum);
    // Outputs: 30
```

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

Writing Output to the Browser Window

You can use the document.write() method to write the content to the current document only whilethat document is being parsed.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Writing into an Browser Window with JavaScript</title>
</head>
<body>
    <script>
    // Printing a simple text message document.write("Hello
    World!"); // Prints: Hello World!

    // Printing a variable value
    var x = 10;
    var y = 20;
    var sum = x + y;
    document.write(sum); // Prints: 30
    </script>
</body>
</html>
```

Inserting Output Inside an HTML Element

You can also write or insert output inside an HTML element using the element's innerHTML property. However, before writing the output first we need to select the element using a methodsuch as getElementById()

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Writing into an HTML Element with JavaScript</title>
</head>
<body>
    <p id="greet"></p>
    <p id="result"></p>

    <script>
    // Writing text string inside an element
    document.getElementById("greet").innerHTML = "Hello World!";

    // Writing a variable value inside an elementvar
    x = 10;
    var y = 20;
    var sum = x + y;
    document.getElementById("result").innerHTML = sum;
    </script>
</body>
</html>
```

JavaScript Data Types

Data Types in JavaScript

Data types basically specify what kind of data can be stored and manipulated within a program.

There are six basic data types in JavaScript which can be divided into three main categories: primitive (or primary), composite (or reference), and special data types. String, Number, and Boolean are primitive data types. Object, Array, and Function (which are all types of objects) are composite data types. Whereas Undefined and Null are special data types.

Primitive data types can hold only one value at a time, whereas composite data types can hold collections of values and more complex entities.

The String Data Type

The string data type is used to represent textual data (i.e. sequences of characters). Strings are created using single or double quotes surrounding one or more characters

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript String Data Type</title>
</head>
<body>
    <script>
    // Creating variables
    var a = 'Hi there!'; // using single quotes var b = "Hi there!";  // using double quotes

    // Printing variable values
    document.write(a + "<br>");
    document.write(b);
    </script>
</body>
</html>
```

Example 2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Including Quotes inside the JavaScript String</title>
</head>
<body>
    <script>

    // Creating variables
    var a = "Let's have a cup of coffee.";var b = 'He said "Hello" and left.'; var c = 'We\'ll never give up.';

    // Printing variable values document.write(a + "<br>");document.write(b + "<br>");document.write(c);
    </script>
</body>
</html>
```

The Number Data Type

The number data type is used to represent positive or negative numbers with or without decimalplace, or numbers written using exponential notation e.g. 1.5e-4 (equivalent to 1.5x10-4).

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Number Data Type</title>
</head>
<body>
    <script>
    // Creating variablesvar a = 25;
    var b = 80.5; var c = 4.25e+6;var d = 4.25e-6;

    // Printing variable values
    document.write(a + "<br>");
    document.write(b + "<br>");
    document.write(c + "<br>");
    document.write(d);
    </script>
</body>
</html>
```

The Number data type also includes some special values which are: Infinity, -Infinity and NaN. Infinity represents the mathematical Infinity ∞, which is greater than any number. Infinity is the result of dividing a nonzero number by 0,
Example 2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Infinity</title>
</head>
<body>
    <script> document.write(16 / 0); document.write("<br>");document.write(-16 / 0);document.write("<br>");
    document.write(16 / -0);
    </script>
</body>
</html>
```

While NaN represents a special Not-a-Number value. It is a result of an invalid or an undefined mathematical operation, like taking the square root of -1 or dividing 0 by 0,
Example 3

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript NaN</title>
</head>
<body>
    <script>
    document.write("Some text" / 2);
    document.write("<br>");
    document.write("Some text" / 2 + 10);
    document.write("<br>");
    document.write(Math.sqrt(-1));
    </script>
</body>
</html>
```

The Boolean Data Type

The Boolean data type can hold only two values: true or false. It is typically used to store values likeyes (true) or no (false), on (true) or off (false)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Boolean Data Type</title>
</head>
<body>
    <script>
    // Creating variables
    var isReading = true;  // yes, I'm reading
    var isSleeping = false; // no, I'm not sleeping

    // Printing variable values
    document.write(isReading + "<br>");
    document.write(isSleeping);
    </script>
</body>
    </html>
```

Example 2

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="utf-8">
    <title>JavaScript Comparisons</title>
</head>
<body>
    <script>
    var a = 2, b = 5, c = 10;

    document.write(b > a) // Output: true
    document.write("<br>");
    document.write(b > c) // Output: false
    </script>
</body>
</html>
```

The Undefined Data Type

The undefined data type can only have one value-the special value undefined. If a variable has been declared, but has not been assigned a value, has the value undefined.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Undefined Data Type</title>
</head>
<body>
    <script>

    // Creating variables
    var a;
    var b = "Hello World!"

    // Printing variable values
    document.write(a + "<br>");
    document.write(b);
    </script>
</body>
</html>
```

The Null Data Type

This is another special data type that can have only one value-the null value. A null value means that there is no value. It is not equivalent to an empty string ("") or 0, it is simply nothing.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Null Data Type</title>
</head>
<body>
    <script> var a
    = null;
    document.write(a + "<br>"); // Print: null

    var b = "Hello World!"
    document.write(b + "<br>"); // Print: Hello World!

    b = null;
    document.write(b) // Print: null
    </script>
</body>
</html>
```

The Object Data Type

The object is a complex data type that allows you to store collections of data.

An object contains properties, defined as a key-value pair. A property key (name) is always a string,but the value can be any data type, like strings, numbers, booleans, or complex data types like arrays,

function and other objects.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>JavaScript Object Data Type</title>
</head>
<body>
    <script>
    var emptyObject = {};
    var person = {"name": "Clark", "surname": "Kent", "age": "36"};

    // For better readingvar
    car = {
       "modal": "BMW X3",
       "color": "white",
       "doors": 5
    }

    // Print variables values in browser's console
    console.log(person);
    console.log(car);
    </script>
   <p><strong>Note:</strong> Check out the browser console by pressing the f12 key on the keyboard.</p>
</body>
</html>
```

The Array Data Type

An array is a type of object used for storing multiple values in single variable. Each value (also calledan element) in an array has a numeric position, known as its index, and it may contain data of any data type-numbers, strings, booleans, functions, objects, and even other arrays. The array index starts from 0, so that the first array element is arr[0] not arr[1].

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Array Data Type</title>
</head>
<body>
    <script>
    // Creating arrays
    var colors = ["Red", "Yellow", "Green", "Orange"];
    var cities = ["London", "Paris", "New York"];

    // Printing array values
    document.write(colors[0] + "<br>");  // Output: Red

    document.write(cities[2]);  // Output: New York
    </script>
</body>
</html>
```

The Function Data Type

The function is callable object that executes a block of code. Since functions are objects, so it is possible to assign them to variables

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Function Data Type</title>
```

```
</head>
<body>
    <script>
    var greeting = function(){
       return "Hello World!";
    }
    // Check the type of greeting variable
    document.write(typeof greeting) // Output: function
    document.write("<br>");
    document.write(greeting());      // Output: Hello World!
    </script>
</body>
</html>
```

Example 2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Function Passed as Argument to Other Function</title>
</head>
<body>
    <script>
    function createGreeting(name){
       return "Hello, " + name;
    }
    function displayGreeting(greetingFunction, userName){
       return greetingFunction(userName);
    }
    var result = displayGreeting(createGreeting, "Rashmi");document.write(result); // Output: Hello, Rashmi
    </script>
</body>
</html>
```

The typeof Operator

The typeof operator can be used to find out what type of data a variable or operand contains. It canbe used with or without parentheses (typeof(x) or typeof x).

The typeof operator is particularly useful in the situations when you need to process the values of different types differently, but you need to be very careful, because it may produce unexpected result in some cases

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript typeof Operator</title>
</head>
<body>
    <script>
    // Numbers
    document.write(typeof 15 + "<br>"); // Prints: "number"
    document.write(typeof 42.7 + "<br>"); // Prints: "number"
    document.write(typeof 2.5e-4 + "<br>"); // Prints: "number"
    document.write(typeof Infinity + "<br>");  // Prints: "number"
    document.write(typeof NaN + "<br>");  // Prints: "number". Despite being "Not-A-Number"

    // Strings
    document.write(typeof " + "<br>"); // Prints: "string" document.write(typeof
    'hello' + "<br>");  // Prints: "string"
  document.write(typeof '12' + "<br>"); // Prints: "string". Number within quotes is
document.write(typeof string

    // Booleans
    document.write(typeof true + "<br>"); // Prints: "boolean"
```

```
    document.write(typeof false + "<br>");  // Prints: "boolean"

    // Undefined
    document.write(typeof undefined + "<br>"); // Prints: "undefined"
    document.write(typeof undeclaredVariable + "<br>"); // Prints: "undefined"

    // Null
    document.write(typeof Null + "<br>");  // Prints: "object"

    // Objects
    document.write(typeof {name: "John", age: 18} + "<br>");  // Prints: "object"

    // Arrays
    document.write(typeof [1, 2, 4] + "<br>");  // Prints: "object"

    // Functions
    document.write(typeof function(){});  // Prints: "function"
    </script>
</body>
</html>
```

JavaScript Operators

What are Operators in JavaScript

Operators are symbols or keywords that tell the JavaScript engine to perform some sort of actions. For example, the addition (+) symbol is an operator that tells JavaScript engine to add two variablesor values, while the equal-to (==), greater-than (>) or less-than (<) symbols are the operators that tells JavaScript engine to compare two variables or values

JavaScript Arithmetic Operators

The arithmetic operators are used to perform common arithmetical operations, such as addition, subtraction, multiplication etc

Operator	Description	Example	Result
+	Addition	$x + $y	Sum of $x and $y
-	Subtraction	$x - $y	Difference of $x and $y.
*	Multiplication	$x * $y	Product of $x and $y.
/	Division	$x / $y	Quotient of $x and $y
%	Modulus	$x % $y	Remainder of $x divided by $y

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Arithmetic Operators</title>
</head>
<body>
    <script> var
    x = 10;var y
    = 4;
    document.write(x + y); // Prints: 14
    document.write("<br>");
    document.write(x - y); // Prints: 6
    document.write("<br>");
    document.write(x * y); // Prints: 40
    document.write("<br>");
    document.write(x / y); // Prints: 2.5
    document.write("<br>");
    document.write(x % y); // Prints: 2
    </script>
</body>
</html>
```

JavaScript Assignment Operators

The assignment operators are used to assign values to variables.

Operator	Description	Example	Is The Same As
=	Assign	x = y	x = y
+=	Add and assign	x += $	x = x + y
-=	Subtract and assign	x -= y	x = x - y
*=	Multiply and assign	x *= y	x = x * y
/=	Divide and assign quotient	x /= y	x = x / y
%=	Divide and assign modulus	x %= y	x = x % y

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Assignment Operators</title>
</head>
<body>
    <script>
    var x;   // Declaring Variable

    x = 10;
    document.write(x + "<br>"); // Prints: 10

    x = 20;
    x += 30;
    document.write(x + "<br>"); // Prints: 50
    x = 50;
    x -= 20;
    document.write(x + "<br>"); // Prints: 30

    x = 5;
    x *= 25;
    document.write(x + "<br>"); // Prints: 125

    x = 50;
    x /= 10;
    document.write(x + "<br>"); // Prints: 5

    x = 100;
    x %= 15;
    document.write(x); // Prints: 10
    </script>
</body>
</html>
```

JavaScript String Operators

There are two operators which can also used be for strings.

Operator	Description	Example	Result
+	Concatenation	str1 + str2	Concatenation of str1 and str2
+=	Concatenation assignment	str1 += str2	Appends the str2 to the str1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript String Operators</title>
</head>
<body>
    <script>
    var str1 = "Hello"; var
    str2 = " World!";
    document.write(str1 + str2 + "<br>"); // Outputs: Hello World!
    str1 += str2;
```

```
    document.write(str1); // Outputs: Hello World!
    </script>
</body>
</html>
```

JavaScript Incrementing and Decrementing Operators

The increment/decrement operators are used to increment/decrement a variable's value.

Operator	Name	Effect
++x	Pre-increment	Increments x by one, then returns x
x++	Post-increment	Returns x, then increments x by one
--x	Pre-decrement	Decrements x by one, then returns x
x--	Post-decrement	Returns x, then decrements x by one

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Incrementing and Decrementing Operators</title>
</head>
<body>
    <script>
    var x; // Declaring Variable

    x = 10;
    document.write(++x);
    document.write("<p>" + x + "</p>");

    x = 10;
    document.write(x++);
    document.write("<p>" + x + "</p>");x
    = 10;
    document.write(--x);
    document.write("<p>" + x + "</p>");x
    = 10;
    document.write(x--);
    document.write("<p>" + x + "</p>");
    </script>
</body>
</html>
```

JavaScript Logical Operators

The logical operators are typically used to combine conditional statements.

Operator	Name	Example	Result
&&	And	x && y	True if both x and y are true
||	Or	x || y	True if either x or y is true
!	Not	!x	True if x is not true

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Logical Operators</title>
</head>
<body>
    <script>
    var year = 2020;

    // Leap years are divisible by 400 or by 4 but not 100
    if((year % 400 == 0) || ((year % 100 != 0) && (year % 4 == 0))){
       document.write(year + " is a leap year.");
    } else{
       document.write(year + " is not a leap year.");
    }
```

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

JavaScript Comparison Operators

The comparison operators are used to compare two values in a Boolean fashion.

Operator	Name	Example	Result
==	Equal	x == y	True if x is equal to y
===	Identical	x === y	True if x is equal to y, and they are of the same type
!=	Not equal	x != y	True if x is not equal to y
!==	Not identical	x !== y	True if x is not equal to y, or they are not of the same type
<	Less than	x < y	True if x is less than y
>	Greater than	x > y	True if x is greater than y
>=	Greater than or equal to	x >= y	True if x is greater than or equal to y
<=	Less than or equal to	x <= y	True if x is less than or equal to y

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Comparison Operators</title>
</head>
<body>
    <script> var
    x = 25;var y
    = 35;
    var z = "25";

    document.write(x == z); // Prints: true
    document.write("<br>");

    document.write(x === z); // Prints: false
    document.write("<br>");

    document.write(x != y); // Prints: true
    document.write("<br>");

    document.write(x !== z); // Prints: true
    document.write("<br>");

    document.write(x < y); // Prints: true
    document.write("<br>");

    document.write(x > y); // Prints: false
    document.write("<br>");

    document.write(x <= y); // Prints: true
    document.write("<br>");

    document.write(x >= y);  // Prints: false
    </script>

</body>
</html>
```

JavaScript Events

An event is something that happens when user interact with the web page, such as when he clickeda link or button, entered text into an input box or textarea, made selection in a select box, pressed key on the keyboard, moved the mouse pointer, submits a form, etc. In some cases, the Browser itself can trigger the events, such as the page load and unload events.

When an event occur, you can use a JavaScript event handler (or an event listener) to detect them and perform specific task or set of tasks. By convention, the names for event handlers always beginwith the word "on", so an event handler for the click event is called onclick, similarly an event handler for the load event is called onload, event handler for the blur event is called onblur, and soon.

There are several ways to assign an event handler. The simplest way is to add them directly to thestart tag of the HTML elements using the special event-handler attributes

Mouse Events

The Click Event (onclick)

The click event occurs when a user clicks on an element on a web page. Often, these are form elements and links. You can handle a click event with an onclick event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Click Event</title>
</head>
<body>
    <button type="button" onclick="alert('You have clicked a button!');">Click Me</button>
    <a href="#" onclick="alert('You have clicked a link!');">Click Me</a>
</body>
</html>
```

The Contextmenu Event (oncontextmenu)

The contextmenu event occurs when a user clicks the right mouse button on an element to open a context menu. You can handle a contextmenu event with an oncontextmenu event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Contextmenu Event</title>
</head>
<body>

  <button type="button" oncontextmenu="alert('You have right-clicked a button!');">Right Click on Me</button>
    <a href="#" oncontextmenu="alert('You have right-clicked a link!');">Right Click on Me</a>
</body>
</html>
```

The Mouseover Event (onmouseover)

The mouseover event occurs when a user moves the mouse pointer over an element.You can handle the mouseover event with the onmouseover event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Mouseover Event</title>
</head>
<body>
  <button type="button" onmouseover="alert('You have placed mouse pointer over a button!');">Place Mouse Over Me</button>
  <a href="#" onmouseover="alert('You have placed mouse pointer over a link!');">Place Mouse Over Me</a>
</body>
</html>
```

The Mouseout Event (onmouseout)

The mouseout event occurs when a user moves the mouse pointer outside of an element.You can handle the mouseout event with the onmouseout event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
```

```
    <title>JavaScript Handling the Mouseout Event</title>
</head>
<body>
   <button type="button" onmouseout="alert('You have moved out of the button!');">Place Mouse Inside Me and Move Out</button>
   <a href="#" onmouseout="alert('You have moved out of the link!');">Place Mouse Inside Me and Move Out</a>
</body>
</html>
```

Keyboard Events

A keyboard event is fired when the user press or release a key on the keyboard. Here're some most important keyboard events and their event handler.

The Keydown Event (onkeydown)

The keydown event occurs when the user presses down a key on the keyboard. You can handle the keydown event with the onkeydown event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Keydown Event</title>
</head>
<body>
    <input type="text" onkeydown="alert('You have pressed a key inside text input!')">
    <hr>
    <textarea cols="30" onkeydown="alert('You have pressed a key inside textarea!')"></textarea>
        <p><strong>Note:</strong> Try to enter some text inside input box and textarea.</p>
</body>
</html>
```

The Keyup Event (onkeyup)

The keyup event occurs when the user releases a key on the keyboard. You can handle the keyup event with the onkeyup event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Keyup Event</title>
</head>
<body>
    <input type="text" onkeyup="alert('You have released a key inside text input!')">
    <hr>
    <textarea cols="30" onkeyup="alert('You have released a key inside textarea!')"></textarea>
        <p><strong>Note:</strong> Try to enter some text inside input box and textarea.</p>
</body>
</html>
```

The Keypress Event (onkeypress)

The keypress event occurs when a user presses down a key on the keyboard that has a character value associated with it. For example, keys like Ctrl, Shift, Alt, Esc, Arrow keys, etc. will not generate a keypress event, but will generate a keydown and keyup event.

You can handle the keypress event with the onkeypress event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Keypress Event</title>
</head>
<body>
    <input type="text" onkeypress="alert('You have pressed a key inside text input!')">
    <hr>
```

```
    <textarea cols="30" onkeypress="alert('You have pressed a key inside textarea!')"></textarea>
    <p><strong>Note:</strong> Try to enter some text inside input box and textarea.</p>
</body>
</html>
```

Form Events

A form event is fired when a form control receive or loses focus or when the user modify a form control value such as by typing text in a text input, select any option in a select box etc. Here're some most important form events and their event handler.

The Focus Event (onfocus)

The focus event occurs when the user gives focus to an element on a web page.You can handle the focus event with the onfocus event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Focus Event</title>
</head>
<body>
    <script>
      function highlightInput(elm){
        elm.style.background = "yellow";
      }
    </script>
    <input type="text" onfocus="highlightInput(this)">
    <button type="button">Button</button>
</body>
</html>
```

The Blur Event (onblur)

The blur event occurs when the user takes the focus away from a form element or a window.You can handle the blur event with the onblur event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Blur Event</title>
</head>
<body>
    <input type="text" onblur="alert('Text input loses focus!')">
    <button type="button">Submit</button>
        <p><strong>Note:</strong> First click inside the text input box then click outside to see howit works.</p>
</body>
</html>
```

The Change Event (onchange)

The change event occurs when a user changes the value of a form element.You can handle the change event with the onchange event handler.

```
   <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Change Event</title>
</head>
<body>
    <select onchange="alert('You have changed the selection!');">
      <option>Select</option>
      <option>Male</option>
      <option>Female</option>
    </select>
```

```
        <p><strong>Note:</strong> Select any option in select box to see how it works.</p>
</body>
</html>
```

The Submit Event (onsubmit)

The submit event only occurs when the user submits a form on a web page. You can handle the submit event with the onsubmit event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Submit Event</title>
</head>
<body>
  <form action="action.php" method="post" onsubmit="alert('Form data will be submitted to the server!');">
      <label>First Name:</label>
      <input type="text" name="first-name" required>
      <input type="submit" value="Submit">
    </form>
</body>
</html>
```

Document/Window Events

Events are also triggered in situations when the page has loaded or when user resize the browser window, etc. Here're some most important document/window events and their event handler.

The Load Event (onload)

The load event occurs when a web page has finished loading in the web browser. You can handle the load event with the onload event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Load Event</title>
</head>
<body onload="window.alert('Page is loaded successfully!');">
    <h1>This is a heading</h1>
    <p>This is paragraph of text.</p>
</body>
</html>
```

The Resize Event (onresize)

The resize event occurs when a user resizes the browser window. The resize event also occurs in situations when the browser window is minimized or maximized.
You can handle the resize event with the onresize event handler.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>JavaScript Handling the Resize Event</title>
</head>
<body>
    <p id="result"></p>
    <script>
      function displayWindowSize(){
        var w = window.outerWidth;
        var h = window.outerHeight;
        var txt = "Window size: width=" + w + ", height=" + h;
        document.getElementById("result").innerHTML = txt;
      }
      window.onresize = displayWindowSize;
    </script>
    <p><strong>Note:</strong> Resize the browser window to see how the resize event works.</p>
</body></html>
```

JavaScript Strings

In this tutorial you will learn how to create and manipulate strings in JavaScript.

What is String in JavaScript

A string is a sequence of letters, numbers, special characters and arithmetic values or combination ofall. Strings can be created by enclosing the string literal (i.e. string characters) either within single quotes (') or double quotes (")

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>Escaping Quotes inside JavaScript Strings</title>
</head><body><script>
    // Creating variablesvar
    str1 = 'it\'s okay';
    var str2 = "He said \"Goodbye\"";
    var str3 = 'She replied \'Calm down, please\';
    // Printing variable values
    document.write(str1 + "<br>");
    document.write(str2 + "<br>");
    document.write(str3);
    </script></body></html>
```

Getting the Length of a String

The length property returns the length of the string, which is the number of characters contained inthe string. This includes the number of special characters as well, such as \t or \n.

```
<!DOCTYPE html>
<html lang="en"><head>
    <meta charset="utf-8">
    <title>Get String Length in JavaScript</title>
</head><body><script>
    var str1 = "This is a paragraph of text.";
    document.write(str1.length + "<br>");
    var str2 = "This is a \n paragraph of text.";
    document.write(str2.length);
</script></body></html>
```

Finding a String Inside Another String

You can use the indexOf() method to find a substring or string within another string. This method returns the index or position of the first occurrence of a specified string within a string.

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Find the Position of Substring within a String</title>
</head><body><script>
    var str = "If the facts don't fit the theory, change the facts.";var
    pos = str.indexOf("facts");
    document.write(pos); // 0utputs: 7
    </script></body></html>
```

Similarly, you can use the lastIndexOf() method to get the index or position of the last occurrence ofthe specified string within a string, like this:

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Find the Position of Substring within a String</title>
</head><body><script>
    var str = "If the facts don't fit the theory, change the facts.";var
    pos = str.lastIndexOf("facts");
    document.write(pos); // 0utputs: 46
    </script></body></html>
```

Both the indexOf(), and the lastIndexOf() methods return -1 if the substring is not found. Both methods also accept an optional integer parameter which specifies the position within the string atwhich to start the search.

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8"><title>JavaScript Find the Position of Substring within a String</title>
</head><body><script>
```

```
    var str = "If the facts don't fit the theory, change the facts.";
    // Searching forwards
    var pos1 = str.indexOf("facts", 20);
    document.write(pos1 + "<br>"); // 0utputs: 46
    // Searching backwards
    var pos2 = str.lastIndexOf("facts", 20);
    document.write(pos2); // 0utputs: 7
    </script></body></html>
```

Searching for a Pattern Inside a String

You can use the search() method to search a particular piece of text or pattern inside a string.
Like indexOf() method the search() method also returns the index of the first match, and returns -1 ifno matches were found, but unlike indexOf() method this method can also take a regular expression as its argument to provide advanced search capabilities.

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Search Text or Pattern inside a String</title>
</head><body><script>
    var str = "Color red looks brighter than color blue.";
    // Case sensitive search
    var pos1 = str.search("color");
    document.write(pos1 + "<br>"); // 0utputs: 30
    // Case insensitive search using regexpvar
    pos2 = str.search(/color/i);
    document.write(pos2); // 0utputs: 0
    </script></body></html>
```

Extracting a Substring from a String

You can use the slice() method to extract a part or substring from a string.
This method takes 2 parameters: start index (index at which to begin extraction), and an optionalend index (index before which to end extraction), like str.slice(startIndex, endIndex).

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Slice Out a Portion of a String</title>
</head><body><script>
    var str = "The quick brown fox jumps over the lazy dog.";
    var subStr = str.slice(4, 15);
    document.write(subStr); // Prints: quick brown
    </script></body></html>
```

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Slice Strings Using Negative Indexes</title>
</head><body><script>
    var str = "The quick brown fox jumps over the lazy dog.";
       document.write(str.length);
    document.write(str.slice(-28, -19) + "<br>"); // Prints: fox jumps
    document.write(str.slice(31)); // Prints: the lazy dog.
    </script></body></html>
```

You can also use the substring() method to extract a section of the given string based on start andend indexes, like str.substring(startIndex, endIndex). The substring() method is very similar to the slice() method, except few differences:

- If either argument is less than 0 or is NaN, it is treated as 0.
- If either argument is greater than str.length, it is treated as if it were str.length.
- If startIndex is greater than endIndex, then substring() will swap those two arguments; for example, str.substring(5, 0) == str.substring(0, 5).

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Extract substring from a String</title>
</head><body><script>
```

```
    var str = "The quick brown fox jumps over the lazy dog.";
    document.write(str.substring(4, 15) + "<br>"); // Prints: quick brown
    document.write(str.substring(9, 0) + "<br>"); // Prints: The quick
    document.write(str.substring(-28, -19) + "<br>"); // Prints nothing
    document.write(str.substring(31)); // Prints: the lazy dog.
    </script></body></html>
```

Extracting a Fixed Number of Characters from a String

JavaScript also provide the substr() method which is similar to slice() with a subtle difference, the second parameter specifies the number of characters to extract instead of ending index, like str.substr(startIndex, length). If length is 0 or a negative number, an empty string is returned.

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Extract Fixed Number of Characters from a String</title>
</head><body><script>
    var str = "The quick brown fox jumps over the lazy dog.";
    document.write(str.substr(4, 15) + "<br>"); // Prints: quick brown fox
    document.write(str.substr(-28, -19) + "<br>"); // Prints nothing
    document.write(str.substr(-28, 9) + "<br>"); // Prints: fox jumps
    document.write(str.substr(31)); // Prints: the lazy dog.
    </script></body></html>
```

Replacing the Contents of a String

You can use the replace() method to replace part of a string with another string. This method takes two parameters a regular expression to match or substring to be replaced and a replacement string,like str.replace(regexp|substr, newSubstr).

This replace() method returns a new string, it doesn't affect the original string that will remain unchanged.

```
    <!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Replace Part of a String with another String</title>
</head><body><script>
    var str = "Color red looks brighter than color blue.";
    var result = str.replace("color", "paint");
    document.write(result); // 0utputs: Color red looks brighter than paint blue.
    </script></body></html>
```

Example 2

```
    <!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Replace Part of a String with another String</title>
</head><body><script>
    var str = "Color red looks brighter than color blue.";
    var result = str.replace(/color/i, "paint");
    document.write(result); // 0utputs: paint red looks brighter than color blue.
    </script></body> </html>
```

Example 3

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Replace All Occurrences of a Substring in a String</title>
</head><body><script>
    var str = "Color red looks brighter than color blue.";
    var result = str.replace(/color/ig, "paint");
    document.write(result); // 0utputs: paint red looks brighter than paint blue.
    </script></body></html>
```

Converting a String to Uppercase or Lowercase

You can use the toUpperCase() method to convert a string to uppercase

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Convert a String to Uppercase Characters</title>
</head><body><script>
    var str = "Hello World!";
    var result = str.toUpperCase();
```

```
    document.write(result); // Prints: HELLO WORLD!
    </script></body></html>
```
toLowerCase()

Example 4

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Convert a String to Lowercase Characters</title>
</head><body><script>
    var str = "Hello World!";
    var result = str.toLowerCase();
    document.write(result); // Prints: hello world!
    </script></body></html>
```

Concatenating Two or More Strings

You can concatenate or combine two or more strings using the + and += assignment operators.

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Join Two or More Strings</title>
</head><body><script>
    var hello = "Hello"; var
    world = "World";
    var greet = hello + " " + world;
    document.write(greet + "<br>"); // Prints: Hello World
    var wish = "Happy"; wish
      += " Birthday";
    document.write(wish); // Prints: Happy Birthday
    </script></body></html>
```

Accessing Individual Characters from a String

You can use the charAt() method to access individual character from a string, like str.charAt(index). The index specified should be an integer between 0 and str.length - 1. If no index is provided the firstcharacter in the string is returned, since the default is 0.

```
<!DOCTYPE html>
<html lang="en">
<head><meta charset="utf-8">
    <title>JavaScript Extract a Single Character from a String</title>
</head><body><script>
    var str = "Hello World!"; document.write(str.charAt() + "<br>"); // Prints: H document.write(str.charAt(6) +
    "<br>"); // Prints: W
    document.write(str.charAt(30) + "<br>"); // Prints nothingdocument.write(str.charAt(str.length - 1)); // Prints: !
    </script></body></html>
```

Splitting a String into an Array

The split() method can be used to splits a string into an array of strings, using the syntax str.split(separator, limit). The seperator argument specifies the string at which each split should occur, whereas the limit arguments specifies the maximum length of the array.If separator argument is omitted or not found in the specified string, the entire string is assigned tothe first element of the array.

```
<!DOCTYPE html><html lang="en"><head><meta charset="utf-8">
    <title>JavaScript Split a String into an Array</title></head><body><script>
    var fruitsStr = "Apple, Banana, Mango, Orange, Papaya";
    var fruitsArr = fruitsStr.split(", ");
    document.write(fruitsArr[0] + "<br>"); // Prints: Apple
    document.write(fruitsArr[2] + "<br>"); // Prints: Mango
    document.write(fruitsArr[fruitsArr.length - 1]); // Prints: Papaya
    document.write("<hr>");
    // Loop through all the elements of the fruits arrayfor(var i in fruitsArr) {document.write("<p>" + fruitsArr[i] + "</p>");
    }</script></body></html>
```

Example 2

```
<!DOCTYPE html><html lang="en">
<head><meta charset="utf-8"> <title>JavaScript Split a String Into an Array of Characters</title>
</head><body><script> var str = "INTERSTELLAR";var strArr = str.split(""); document.write(strArr[0] + "<br>"); //
Prints: I document.write(strArr[1] + "<br>"); // Prints: N
    document.write(strArr[strArr.length - 1]); // Prints: Rdocument.write("<hr>"); // Prints: N
    // Loop through all the elements of the characters array and print themfor(var i in strArr) {
      document.write("<br>" + strArr[i]);}</script></body></html>
```

PHP

What is PHP

- PHP stands for Hypertext Preprocessor.
- PHP is an interpreted language, i.e., there is no need for compilation.
- PHP is a server-side scripting language.
- PHP is faster than other scripting languages, for example, ASP and JSP.

Web Development

PHP is widely used in web development nowadays. PHP can develop dynamic websites easily. But you must have the basic the knowledge of following technologies for web development as well.

- HTML
- CSS
- JavaScript

Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

PHP Features

There are given many features of PHP.

- **Performance**: Script written in PHP executes much faster then those scripts written in other languages such as JSP & ASP.
- **Open Source Software**: PHP source code is free available on the web, you can developed all the version of PHP according to your requirement without paying any cost.
- **Platform Independent**: PHP are available for WINDOWS, MAC, LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.
- **Compatibility**: PHP is compatible with almost all local servers used today like Apache, IIS etc.
- **Embedded**: PHP code can be easily embedded within HTML tags and script.

Install PHP

To install PHP, we will suggest you to install AMP (Apache, MySQL, PHP) software stack. It is available for all operating systems. There are many AMP options available in the market that are given below:

- **WAMP** for Windows
- **LAMP** for Linux
- **MAMP** for Mac
- **SAMP** for Solaris
- **XAMPP** (Cross, Apache, MySQL, PHP, Perl) for Cross Platform: It includes some other components too such as FileZilla, OpenSSL, Webalizer, Mercury Mail etc.
-

PHP Example

It is very easy to create a simple PHP example. To do so, create a file and write HTML tags + PHP code and save this file with .php extension.
All PHP code goes between php tag. A syntax of PHP tag is given below:

```
<?php
//your code here
?>
```

Example:

```
<html>
<body>
<?php
echo "<h2>Hello First PHP</h2>";
?>
</body>
</html>
```

PHP echo: printing string

```
<?php
```

```
echo "Hello by PHP echo";
?>
```

Run a PHP program in XAMPP Server

PHP program can be run under various like WAMP, XAMPP etc.

- **WAMP Server**: this server is a web development platform which helps in creating dynamic web applications.
- **XAMPP Server**: It is a free open source cross-platfrom web server package.

I am using XamppServer to run my program, you can download it . After downloading, just follow the following step to start xampp server:

Step1

Install XAMPP

Step2

Assume you installed xampp in C Drive.

Go to: **C:\xampp\htdocs**

Create your own folder, name it for example as **tutorialspoint**.

Step3

Now create your first php program in xampp and name it as **"add.php"**:

Step4

```
<html>
<head><title>Addition php</title></head>
<body>

<?php
   ?>
   print "<h2>php program to add two numbers...</h2><br />";
$val1 = 20;
$val2 = 20;
$sum = $val2 + $val2; /* Assignment operator */echo
"Result(SUM): $sum";

   </body>
   </html>
```

Now double click on "**XAAMP CONTROL PANEL**" on desktop and START "**Apache**" (icon also appears on the bottom)

Step5

Type **localhost** on your browser and press enter:

It will show the following:

Step6

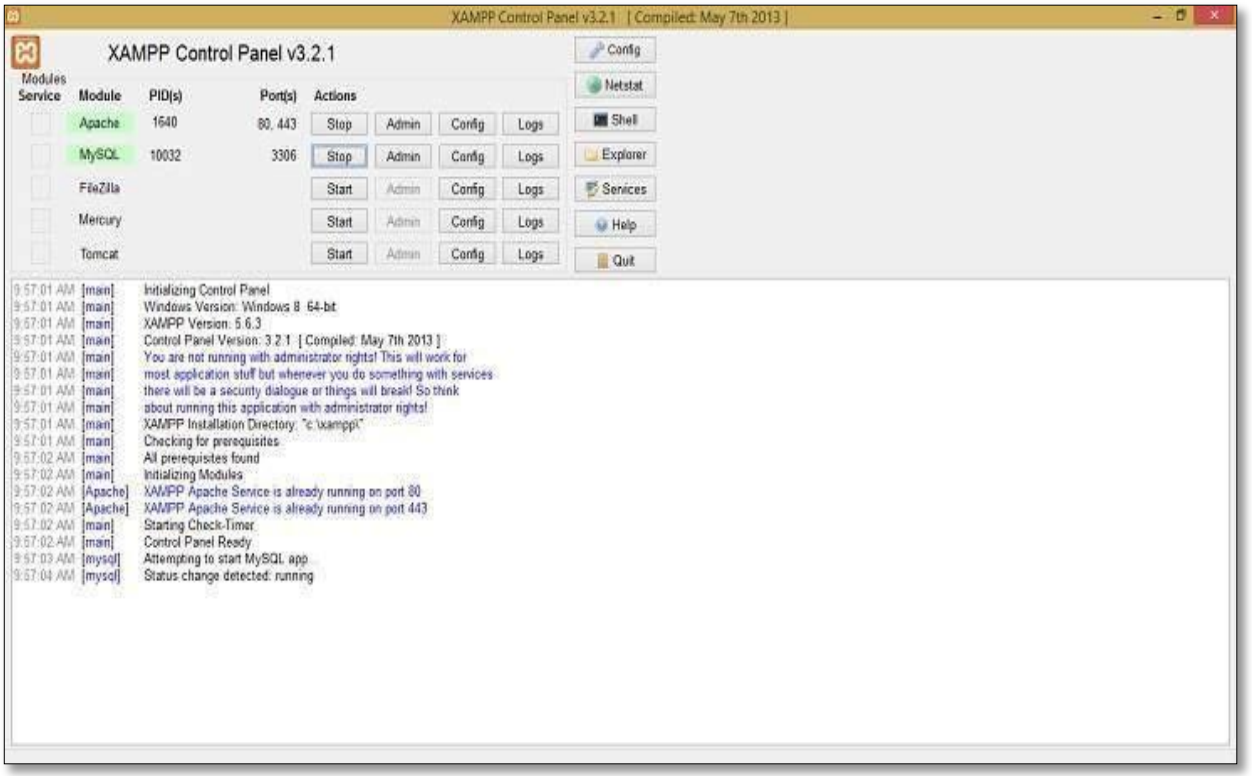
Now type the following on browser:

http://localhost/

Step7

Click on "**add.php**" and it will show the following:

The RESULT is 40 by adding both the values.This way you can run your php program in XAMPP server…




```
    ['submit']))
    {
        $number1 = $_POST['number1'];
        $number2 = $_POST['number2'];
        $sum =  $number1+$number2;
echo "The sum of $number1 and $number2is: ".$sum;
}
?>
</body>
</html>
```

PHP PROGRAMS

1) Area of triangle

```
<?php
$base = 10;
 $height = 15;
 echo "area with base $base and height $height= "
. ($base * $height) / 2;
 ?>
```

2) pattern printing

```
<?php
for($i=0;$i<=5;$i++){
for($j=5-$i;$j>=1;$j--){echo
"* ";
}echo "<br>";}?>
```

3) pattern printing

```
<?php
for($i=0;$i<=5;$i++){
for($j=1;$j<=$i;$j++){echo
"* ";
}
echo "<br>";
}
?>
```

4)
```
<?php
$num=16;
echo "$num * 1 = $num <br/>";
echo "$num * 2 = ".($num*2)." <br/>";
echo "$num * 3 = ".($num*3)." <br/>";
echo "$num * 4 = ".($num*4)." <br/>";
echo "$num * 5 = ".($num*5)." <br/>";
echo "$num * 6 = ".($num*6)." <br/>";
echo "$num * 7 = ".($num*7)." <br/>";
echo "$num * 8 = ".($num*8)." <br/>";
echo "$num * 9 = ".($num*9)." <br/>";
echo "$num * 10 = ".($num*10)." <br/>";
?>
```

5)
```
<?php /* Area of Circle*/
 $radius=5;
 echo "Area of Circle is : </br>";
 $area=3.142 * ($radius * $radius);
 print $area;
?>
```

6)
```
<?php /* Area of Reactangle*/
 $width=5;
 $height=5;
 echo "Area of Reactangle is : </br>";
 $area=$width * $height;
 print $area;
?>
```

7)
```
<?php
$age=17;
if($age>18)
        print "Your Age is $age and Eligible for
Voting";
else
        print "Minimum age is 18. and your age is $age.
Not Eligible for Voting";
?>
```

8)
```
<?php
$num1=10;
$num2=2;
$result=$num1/$num2;
echo "$result";
?>
```

9)
```
<?php
$num=10;
echo $num++."<br/>";
echo --$num."<br/>";
echo $num;
?>
```

10)
```
<?php
$amount=50000;
$withdrawal=6000;

if($withdrawal>$amount){
        echo "Insufficiant Funds";
}else{
        echo "Available Amount is : ".($amount-
$withdrawal);
}?>
```

11)
```
<?php
$account1="Savings";
$account2="Current";
echo "Types of Account <br/>";
echo "$account1 <br/>$account2";
?><body>
<form method="POST" action="">
<input type="text" name="user_choice"
placeholder="Enter your choice"><br/><br/>
<input type="submit" name="submit" value="Submit">
</form></body>
<?php   if(isset($_POST['submit'])){
if($_POST['user_choice']=='Savings'){
echo " Savings Account is Created";}else
if($_POST['user_choice']=='Current'){
echo "Current Account is Created";
}else{echo "Please Enter one of the Types of Account";
}}?>
```

12)
```
<body>
<form method="post" action="">
<label>Enter the Number</label><br/><br/>
<input type="number" name="number"><br/><br/>
<input type="submit" name="submit" value="Submit">
</form></body>
<?php
if(isset($_POST['submit'])){
$number=$_POST['number'];
echo "You have Entered $number.<br/>";
if($number<10)
echo "It is Single Digit";
else if($number>10 and $number<100)
echo "It is Double Digit";
else if($number>100 and $number<1000)
echo "It is Triple Digit";
else if($number>1000 and $number<10000)
echo "It is Four Digit";
else if($number>10000)
echo "It is More than Four Digits";
}?>
```

13)
```
<body>
<form method="post" action="">
<label>User Name </label>
<input type="text" name="username"
placeholder="Enter User Name"
required="required"><br/><br/>
<label>Password</label>
<input type="password" name="password"
placeholder="Enter Password"><br/><br/>
<input type="submit" name="submit" value="Login">
</form></body><?php
if(isset($_POST['submit'])){
```

```
$username=$_POST['username'];
$password=$_POST['password'];
if($username==='Rashmi' and $password==="1234"){
echo "Login Successful";
}else{
echo "Please Check Username and Password";    }} ?>
```

14)
```
<?php
$today=date("D");
/*
d- date
m-month (num)
y-year(2 digit)
Y-year(4 digit)
M-month(3 char)
F-month(Full Month)
*/
echo "$today";
?>
```

15)
```
<?php
$today=date("D");

if($today==='Mon'){
        echo "Today is Monday";
}else if($today==='Tue'){
        echo "Today is Tue";
}?>
```

16)
```
<body>
        <form method="post" action="">
        <label>Enter the Number from 1 to 3</label>
        <input type="number"
name="number"><br/><br/>
        <input type="submit" name="submit"
value="Submit"></form>
</body>
<?php
if(isset($_POST['submit'])){
        $number=$_POST['number'];
        if($number==='1'){
                echo "List of Animals <br/>";
                echo "1. Dog";
        }else if($number==='2'){
                echo "List of Flowers <br/>";
                echo "1. Lotus";
        }else if($number==='3'){
                echo "List of Colors <br/>";
                echo "1. Red";
        }else{echo "Invalid Number";
        }}?>
```

17)
```
<body>
        <form method="post" action="">
        <label>Enter the Number from 1 to 3</label>
        <input type="number"
name="number"><br/><br/>
        <input type="submit" name="submit"
value="Submit"></form>
</body> <?php
if(isset($_POST['submit'])){
        $number=$_POST['number'];
        switch ($number) {
                case '1':
                        echo "List of Animals <br/>";
                        echo "1. Dog";
                        break;
                case '2':
                        echo "List of Flowers <br/>";
                        echo "1. Lotus";
                        break;
                case '3':
                        echo "List of Colors <br/>";
                        echo "1. Red";
                        break;
                default:
                        echo "Invalid Number";
                        break;   }          }   ?>
```

18)
```
<?php
$num=2;
$counter=1;
for($i=1;$i<10;$i++){
        echo "$i:$num<br/>";
        $counter++; }
echo "<br/>count:$counter";
?>
```

19)
```
<?php
$num =25;
for($i=1;$i<=10;$i++){
echo "$num * $i = ".($num*$i)."<br/>";} ?>
```

20)
```
<?php
$num=16;
$i=1;
while($i<=10){
        echo "$num * $i = ".($num*$i)."<br/>";
        $i++;}  ?>
```

21)
```
<?php
for($i=1;$i<=5;$i++){
        for($j=1;$j<=$i;$j++){
                echo " * ";
        }       echo "<br/>";}   ?>
```

22)
```
<?php
$names=array('Rashmi','Nikhil','Sunil','Anil');
echo $names[2]." <br/>";
foreach ($names as  $value) {
echo "$value <br/>";}
/*Array is index is of type integer is called Numeric
Array, Array is index is of type String is called
Associative Array*/
?>
```

23)
```
<?php
$number=array(12,34,56,78,90);
echo $number[1]."<br/>";
echo $number[3]."<br/>";
echo $number[4]."<br/>";
?>
```

24)
```
<?php
$emp=array("Rashmi"=>5000,"Neha"=>10000,"Sunil"=
>2000);
echo $emp["Neha"]."<br/>";
echo $emp["Sunil"]."<br/>";
echo $emp["Rashmi"]."<br/>";
?>
```

25)
```
<?php
$emp=array("Rashmi"=>20000,"Naveen"=>5000,"Neha
"=>2000);
foreach ($emp as $key => $value) {
echo $key." Salary is ".$emp[$key]."<br/>";
}  ?>
```

26)
```
<?php
$result=array(   "Rashmi"=>array(
                 "maths"=>35,
                 "science"=>45,
```

```
                "kannada"=>56 ),
        "Nikhil"=>array(
                "maths"=>55,
                "science"=>65,
                "kannada"=>54 ) );
echo $result["Rashmi"]["science"]."<br/>";
echo $result["Nikhil"]["kannada"]."<br/>";
?>
```

27)
```
<?php
//implode
$arr = array('Hello','ToDay','is','Thursday');
echo implode(" ",$arr);
echo "<br/>";
//explode
$string1 = "$1000 | $2000 | $500| $3000";
print_r(explode("|",$string1));
?>
```

28)
```
<?php
/* Creating Cookies */setcookie("name", "Naveen",
time()+3600, "/","", 0);   setcookie("age", "36",
time()+3600, "/", "",  0);
?><html>     <head>
    <title>Setting Cookies with PHP</title>
  </head>   <body>
    <?php echo "Set Cookies"?>
  </body> </html>
```

29)
```
<html>
  <head>
    <title>Accessing Cookies with PHP</title>
  </head>     <body>    <?php
  echo $_COOKIE["name"]. "<br />";
   /* is equivalent to */
 //  echo $HTTP_COOKIE_VARS["name"]. "<br />";
echo $_COOKIE["age"] . "<br />";
              /* is equivalent to */
    //    echo $HTTP_COOKIE_VARS["age"] . "<br />";
     ?>        </body> </html>
```

30)
```
<?php
session_start();
$_SESSION['username']="Rashmi";
$_SESSION['email']="RNV@ampwork.com";
echo "Session is Created";
?>
```

31)
```
<?php
session_start();
echo $_SESSION['username'];
echo "<br/>";
echo $_SESSION['email'];
/*Un set the Session Data */
unset($_SESSION['email']);
session_destroy();
?>
```

32)
```
<?php ?>
<!DOCTYPE html>
<html> <head>
<title>GET Request</title>
</head> <body>
<form method="GET" action="process.php">
<input type="text" name="username"
placeholder="Enter Username" required="required">
<br/><br/>
<input type="password" name="password"
placeholder="Enter Password" required="required">
<br/><br/>
<input type="submit" name="submit" value="submit">
</form> </body> </html>
```

33)
```
<?php ?><!DOCTYPE html>
<html><head>  <title>POST Request</title>
</head><body><form method="POST"
action="process1.php"> <input type="text"
name="username" placeholder="Enter Username"
required="required">     <br/><br/>        <input
type="password" name="password" placeholder="Enter
Password" required="required"><br/><br/>
          <input type="submit" name="submit"
value="submit"></form></body></html>
```

34)
```
<!DOCTYPE html>
<html><head>  <title>File Upload</title>
</head><body>
<form method="post" action=""
enctype="multipart/form-data" >
          <label>Choose File </label>
          <input type="file" name="image" required="">
          <input type="submit" name="submit"
value="Upload"></form></body></html><?php
          if(isset($_FILES['image'])){
                $error=array();
                $file_name=$_FILES['image']['name'];
                $file_size=$_FILES['image']['size'];
          $file_tmp=$_FILES['image']['tmp_name'];
                $file_type=$_FILES['image']['type'];
                $tmp=explode('.',$file_name);
                $file_ext=strtolower(end($tmp));
          $extensions=array("jpg","jpeg","gif","png");
          if(in_array($file_ext,$extensions)===false){
          $error[]= "Invalid File Format";
          }if($file_size>204800){
          $error[]="File Size must be Less than 2 MB";}
if(empty($error)==true){
move_uploaded_file($file_tmp, 'img/'.$file_name);
echo "Success"; }else{                print_r($error);
}} ?>
```

35)
```
<?php
$dirname = "img/";$images =
glob($dirname."*.{jpg,jpeg,png}",GLOB_BRACE);
foreach($images as $image) {
   ?>    <img src="<?=$image?>" /><br />
   <a href="<?=$image?>" download> Download
</a><br/>    <?php        } ?>
```

36)
```
<?php
if(isset($_GET['submit'])){
          $username=$_GET['username'];
          $password=$_GET['password'];
          if($username==='Rashmi' and
$password==='1234'){
//header("location:http://facebook.com");
          echo "Login Successful";}else{  echo "Invalid
Login Information";       }}?>
```

37)
```
<?php
if(isset($_POST['submit'])){
          $username=$_POST['username'];
          $password=$_POST['password'];
          if($username==='Rashmi' and
$password==='1234'){
          //header("location:http://facebook.com");
echo "Login Successful";}else{  echo "Invalid Login
Information";    } } ?>
```

ORACLE-DBMS-RDBMS-MYSQL/PLSQL

1

❖ **What is SQL?**

SQL is Structured Query Language, which is a computer language for storing, manipulating
and retrieving data stored in relational database.
SQL is the standard language for Relation Database System. All relational database
management systems like MySQL, MS Access, Oracle, Sybase, Informix, postgres and SQL
Server use SQL as standard database language.

❖ **Why SQL?**

- Allows users to access data in relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre
compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures, and views

❖ **History:**

- 1970 -- Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational
databases. He described a relational model for databases.
- 1974 -- Structured Query Language appeared.
- 1978 -- IBM worked to develop Codd's ideas and released a product named System/R.
- 1986 -- IBM developed the first prototype of relational database and standardized by
ANSI. The first relational database was released by Relational Software and its later
becoming Oracle.

❖ **SQL Process:**

When you are executing an SQL command for any RDBMS, the system determines the best
way to carry out your request and SQL engine figures out how to interpret the task.
There are various components included in the process. These components are Query
Dispatcher, Optimization Engines, Classic Query Engine and SQL Query Engine, etc. Classic
query engine handles all non-SQL queries but SQL query engine won't handle logical files.
Following is a simple diagram showing SQL Architecture:

2

❖ **SQL Commands:**

The standard SQL commands to interact with relational databases are CREATE, SELECT,
INSERT, UPDATE, DELETE and DROP. These commands can be classified into groups
based on their nature:

❖ **DDL - Data Definition Language:**

❖ **DML - Data Manipulation Language:**

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

Command	Description
SELECT	Retrieves certain records from one or more tables
INSERT	Creates a record
UPDATE	Modifies records
DELETE	Deletes records

3

❖ **DCL - Data Control Language:**

❖ **What is RDBMS?**

RDBMS stands for Relational Database Management System.
RDBMS is the basis for SQL,
and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and
Microsoft Access.
A Relational database management system (RDBMS) is a database management system
(DBMS) that is based on the relational model as introduced by E. F. Codd.

❖ **What is table?**

The data in RDBMS is stored in database objects called tables. The table is a collection of
related data entries and it consists of columns and rows.
Remember, a table is the most common and simplest form of data storage in a relational
database. Following is the example of a CUSTOMERS table:

```
+----+----------+-----+-----------+----------+
| ID | NAME | AGE | ADDRESS | SALARY |
+----+----------+-----+-----------+----------+
| 1 | sunil | 32 | Ahmedabad | 2000.00 |
| 2 | Khilan | 25 | Delhi | 1500.00 |
| 3 | kaushik | 23 | Kota | 2000.00 |
+----+----------+-----+-----------+----------+
```

❖ **What is field?**

Every table is broken up into smaller entities called fields. The fields in the CUSTOMERS
table consist of ID, NAME, AGE, ADDRESS and SALARY.
A field is a column in a table that is designed to maintain specific information about every
record in the table.

❖ **What is record or row?**

A record, also called a row of data, is each individual entry that exists in a table.

Command	Description
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

4

For example there are 7 records in the above CUSTOMERS table. Following is a single row
of data or record in the CUSTOMERS table:

```
+----+----------+-----+-----------+----------+
| 1 | Anil| 32 | Ahmedabad | 2000.00 |
+----+----------+-----+-----------+----------+
```

A record is a horizontal entity in a table.

❖ **What is column?**

A column is a vertical entity in a table that contains all information associated with a specific
field in a table.
For example, a column in the CUSTOMERS table is ADDRESS, which represents location
description and would consist of the following:

```
+-----------+
| ADDRESS |
+-----------+
| Ahmedabad |
| Delhi |
| Kota |
| Mumbai |
+----+------+
```

❖ **What is NULL value?**

A NULL value in a table is a value in a field that appears to be blank, which means a field
with a NULL value is a field with no value.
It is very important to understand that a NULL value is different than a zero value or a field
that contains spaces. A field with a NULL value is one that has been left blank during record
creation.

❖ **SQL Constraints:**
Constraints are the rules enforced on data columns on table. These are used to limit the type
of data that can go into a table. This ensures the accuracy and reliability of the data in the
database.
Constraints could be column level or table level. Column level constraints are applied only to
one column where as table level constraints are applied to the whole table.

5
Following are commonly used constraints available in SQL:
• NOT NULL Constraint: Ensures that a column cannot have NULL value.
• DEFAULT Constraint: Provides a default value for a column when none is specified.
• UNIQUE Constraint: Ensures that all values in a column are different.
• PRIMARY Key: Uniquely identified each rows/records in a database table.
• FOREIGN Key: Uniquely identified a rows/records in any another database table.
• CHECK Constraint: The CHECK constraint ensures that all values in a column satisfy
certain conditions.
• INDEX: Use to create and retrieve data from the database very quickly.
❖ **ORACLE**
It is a very large and multi-user database management system. Oracle is a relational database
management system developed by 'Oracle Corporation'.
Oracle works to efficiently manage its resource, a database of information, among the
multiple clients requesting and sending data in the network.
It is an excellent database server choice for client/server computing. Oracle supports all major
operating systems for both clients and servers, including MSDOS, NetWare, UnixWare, OS/2
and most UNIX flavors.
❖ **History:**
Oracle began in 1977 and celebrating its 32 wonderful years in the industry (from 1977 to
2009).
• 1977 - Larry Ellison, Bob Miner and Ed Oates founded Software Development
Laboratories to undertake development work.
• 1979 - Version 2.0 of Oracle was released and it became first commercial relational
database and first SQL database. The company changed its name to Relational
Software Inc. (RSI).
• 1981 - RSI started developing tools for Oracle.
• 1982 - RSI was renamed to Oracle Corporation.
• 1983 - Oracle released version 3.0, rewritten in C language and ran on multiple
platforms.
• 1984 - Oracle version 4.0 was released. It contained features like concurrency control
- multi-version read consistency, etc.
• 1985 - Oracle version 4.0 was released. It contained features like concurrency control
- multi-version read consistency, etc.

6
• 2007 - Oracle has released Oracle11g. The new version focused on better partitioning,
easy migration etc.
❖ **Features:**
• Concurrency
• Read Consistency
• Locking Mechanisms
• Quiesce Database
• Portability
• Self-managing database
• SQL*Plus
• ASM
• Scheduler
• Resource Manager
• Data Warehousing
• Materialized views
• Bitmap indexes
• Table compression
• Parallel Execution
• Analytic SQL
• Data mining
• Partitioning
❖ **The CL SCR Command:**
The CL SCR Command is used to clear the screen.
Ex:
SQL> CL SCR
❖ **Displaying all the tables:**
To display all the tables use the below statement
SQL> SELECT * FROM TAB;
❖ **Displaying all the tables, views, index:**
SQL> SELECT * FROM CAT;

7
❖ **The SQL CREATE TABLE Statement**
The CREATE TABLE statement is used to create a table in a database.
Tables are organized into rows and columns; and each table must have a name.
Syntax
CREATE TABLE *table_name*
(
column_name1 data_type(*size*),
column_name2 data_type(*size*),
column_name3 data_type(*size*),
....
);
The column_name parameters specify the names of the columns of the table.
The data_type parameter specifies what type of data the column can hold (e.g. varchar,
integer, decimal, date, etc.).
The size parameter specifies the maximum length of the column of the table.
❖ **SQL CREATE TABLE Example**
SQL> CREATE TABLE STUDENT(ROLLNO NUMBER(6),
2 NAME VARCHAR2(10),
3 DOA DATE,
4 FEES NUMBER(6));
❖ **Displaying structure of the table:**
SQL> DESC STUDENT;
Or
SQL> DESCRIBE STUDENT;
❖ **The SQL INSERT INTO Statement**
The INSERT INTO statement is used to insert new records in a table.
❖ **SQL INSERT INTO Syntax**
INSERT INTO *table_name*
VALUES (*value1*,*value2*,*value3*,...);
It is possible to write the INSERT INTO statement in four forms.

8
❖ **Insert all fields one record method:**
SQL> INSERT INTO STUDENT VALUES(1,'AMIT','01-JUN-14',6500);
❖ **Insert few fields one record:**
SQL> INSERT INTO STUDENT(ROLLNO,NAME)VALUES(2,'ANIL');
❖ **Displaying records of student table:**
SQL> SELECT * FROM STUDENT;
❖ **Insert all fields many records:**
SQL> INSERT INTO STUDENT
VALUES(&ROLLNO,'&NAME','&DOA',&FEES);
Enter value for rollno: 3
Enter value for name: MAHESH
Enter value for doa: 01-DEC-14
Enter value for fees: 98500
old 1: INSERT INTO STUDENT
VALUES(&ROLLNO,'&NAME','&DOA',&FEES)
new 1: INSERT INTO STUDENT VALUES(3,'MAHESH','01-DEC-14',98500)
1 row created.

SQL> /
Enter value for rollno: 4
Enter value for name:
RASHMI
Enter value for doa: 01-JUN-13
Enter value for fees: 9700
old 1: INSERT INTO STUDENT
VALUES(&ROLLNO,'&NAME','&DOA',&FEES)
new 1: INSERT INTO STUDENT VALUES(4,'RASHMI','01-JUN-13',9700)
1 row created.

❖ **Insert few fields many records:**

SQL> INSERT INTO
STUDENT(ROLLNO,NAME)VALUES(&ROLLNO,'&NAME');
Enter value for rollno: 6
Enter value for name: JAY
old 1: INSERT INTO
STUDENT(ROLLNO,NAME)VALUES(&ROLLNO,'&NAME')
new 1: INSERT INTO STUDENT(ROLLNO,NAME)VALUES(6,'JAY')

9

❖ **Displaying all records:**

SQL> SELECT * FROM STUDENT;

ROLLNO	NAME	DOA	FEES
1	AMIT	01-JUN-14	6500
2	ANIL		
3	MAHESH	01-DEC-14	98500
4	RASHMI	01-JUN-13	9700
5	AJAY	05-JUL-14	8500
6	JAY		
7	AKASH		
8	BASU		

❖ **The COMMIT Command:**

The COMMIT command is the transactional command used to save changes invoked by a
transaction to the database.
The COMMIT command saves all transactions to the database since the last COMMIT or

❖ **The syntax for COMMIT command is as follows:**

COMMIT;
SQL> COMMIT;
Commit complete.

❖ **The Update Command:**

SQL - UPDATE Query
The SQL UPDATE Query is used to modify the existing records in a table.
You can use WHERE clause with UPDATE query to update selected rows otherwise all the
rows would be affected.

Syntax:

The basic syntax of UPDATE query with WHERE clause is as follows:
UPDATE table_name
SET column1 = value1, column2 = value2... , columnN = valueN
WHERE [condition];

10

Ex: Updating One Record:

SQL> UPDATE STUDENT SET DOA='01-JUN-14',FEES=11000 WHERE ROLLNO=8;
1 row updated.

Ex: Updating few Records:

SQL> UPDATE STUDENT SET DOA='01-JUN-14',FEES=12000 WHERE ROLLNO>=4
AND ROLLNO<=7;
4 rows updated.

❖ **Updating all records:**

SQL> UPDATE STUDENT SET DOA='01-JUN-13',FEES=15000;
8 rows updated.

SQL - ALTER TABLE Command

The SQL ALTER TABLE command is used to add, delete or modify columns in an existing
table.
You would also use ALTER TABLE command to add and drop various constraints on a an
existing table.

Syntax:

The basic syntax of ALTER TABLE to add a new column in an existing table is as follows:
ALTER TABLE table_name ADD column_name datatype;

SQL> ALTER TABLE STUDENT ADD(CITY VARCHAR2(10),PHNO VARCHAR2(10));
Table altered.
SQL> DESC STUDENT;

Name	Null?	Type
ROLLNO		NUMBER(6)
NAME		VARCHAR2(10)
DOA		DATE
FEES		NUMBER(6)
CITY		VARCHAR2(10)
PHNO		VARCHAR2(10)

11

The basic syntax of ALTER TABLE to DROP COLUMN in an existing table is as follows:
ALTER TABLE table_name DROP COLUMN column_name;

SQL> ALTER TABLE STUDENT DROP COLUMN CITY;
Table altered.
SQL> DESC STUDENT;

Name	Null?	Type
ROLLNO		NUMBER(6)
NAME		VARCHAR2(10)
DOA		DATE
FEES		NUMBER(6)
PHNO		VARCHAR2(10)

The basic syntax of ALTER TABLE to change the DATA TYPE of a column in a table is as
follows:

ALTER TABLE table_name MODIFY COLUMN column_name datatype;

SQL> ALTER TABLE STUDENT MODIFY(NAME VARCHAR2(12));
Table altered.
SQL> DESC STUDENT;

Name	Null?	Type
ROLLNO		NUMBER(6)
NAME		VARCHAR2(12)
DOA		DATE
FEES		NUMBER(6)
CITY		VARCHAR2(10)
PHNO		VARCHAR2(10)

12

❖ **The Delete command:**

SQL - DELETE Query

The SQL DELETE Query is used to delete the existing records from a table.
You can use WHERE clause with DELETE query to delete selected rows, otherwise all the
records would be deleted.

Syntax:

The basic syntax of DELETE query with WHERE clause is as follows:
DELETE FROM table_name
WHERE [condition];

❖ **Deleting one record:**

SQL> DELETE FROM STUDENT WHERE ROLLNO=8;
1 row deleted.

❖ **Deleting few records:**

SQL> DELETE FROM STUDENT WHERE ROLLNO>=3 AND ROLLNO<=7;
5 rows deleted.

❖ **Deleting all records:**

SQL> DELETE FROM STUDENT;
SQL> SELECT * FROM STUDENT;
no rows selected

❖ **Retrieving records / restoring records**

SQL> ROLLBACK;
Rollback complete.
SQL> SELECT * FROM STUDENT;

ROLLNO	NAME	DOA	FEES	CITY	PHNO
1	AMIT	01-JUN-13	15000	HUBLI	9845457585
2	ANIL	01-JUN-13	15000	HUBLI	

3 MAHESH 01-JUN-13 15000 HUBLI
7 AKASH 01-JUN-13 15000 HUBLI
8 BASU 01-JUN-13 15000 HUBLI 9844444444

13

❖ **The SAVEPOINT Command:**

A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain
point without rolling back the entire transaction.
The syntax for SAVEPOINT command is as follows:
SAVEPOINT SAVEPOINT_NAME;
This command serves only in the creation of a SAVEPOINT among transactional statements.

❖ **Rollback**

The ROLLBACK command is used to undo a group of transactions.
The syntax for rolling back to a SAVEPOINT is as follows:
ROLLBACK TO SAVEPOINT_NAME;

```
SQL> SAVEPOINT SP1;
Savepoint created.
SQL> DELETE FROM STUDENT WHERE ROLLNO=8;
1 row deleted.
SQL> SELECT * FROM STUDENT;
ROLLNO NAME DOA FEES
---------- ---------- --------- ----------
1 AMIT 01-JUN-13 15000
2 ANIL 01-JUN-13 15000
3 MAHESH 01-JUN-13 15000
4 RASHMI 01-JUN-13 15000
5 AJAY 01-JUN-13 15000
6 JAY 01-JUN-13 15000
7 AKASH 01-JUN-13 15000
7 rows selected.
SQL> SAVEPOINT SP2;
Savepoint created.
SQL> DELETE FROM STUDENT WHERE ROLLNO>=3 AND
ROLLNO<=7;
5 rows deleted.
```

14

```
SQL> SAVEPOINT SP3;
Savepoint created.
SQL> DELETE FROM STUDENT;
2 rows deleted.
SQL> ROLLBACK TO SP2;
Rollback complete.
```

❖ **The RELEASE SAVEPOINT Command:**

The RELEASE SAVEPOINT command is used to remove a SAVEPOINT that you have
created.

The syntax for RELEASE SAVEPOINT is as follows:

RELEASE SAVEPOINT SAVEPOINT_NAME;
Once a SAVEPOINT has been released, you can no longer use the ROLLBACK command to
undo transactions performed since the SAVEPOINT.

```
SQL>RELEASE SAVEPOINT SP1;
```

❖ **Deleting records permenently:**

SQL - TRUNCATE TABLE Command

The SQL TRUNCATE TABLE command is used to delete complete data from an existing
table.
You can also use DROP TABLE command to delete complete table but it would remove
complete table structure form the database and you would need to re-create this table once
again if you wish you store some data.
Syntax:
The basic syntax of TRUNCATE TABLE is as follows:
TRUNCATE TABLE table_name;

```
SQL> TRUNCATE TABLE STUDENT;
Table truncated.
```

15

```
SQL> SELECT * FROM STUDENT;
no rows selected
```

❖ **SQL - DROP or DELETE Table**

The SQL DROP TABLE statement is used to remove a table definition and all data, indexes,
triggers, constraints, and permission specifications for that table.
NOTE: You have to be careful while using this command because once a table is deleted then
all the information available in the table would also be lost forever.

Syntax:

Basic syntax of DROP TABLE statement is as follows:
DROP TABLE table_name;

```
SQL> DROP TABLE STUDENT;
Table dropped.
```

❖ **Example for linking table or setting relationship for student and result tables**

```
SQL> CREATE TABLE STUD(ROLLNO NUMBER(3)PRIMARY KEY,
2 NAME VARCHAR2(10),
3 ADDRESS VARCHAR2(10));
Table created.
SQL> DESC STUD;
Name Null? Type
----------------------------------------- -------- ----------------------------
ROLLNO NOT NULL NUMBER(3)
NAME VARCHAR2(10)
ADDRESS VARCHAR2(10)
SQL> INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&ADDRESS');
Enter value for rollno: 1
Enter value for name: AJAY
Enter value for address: HOSUR
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&ADDRESS')
new 1: INSERT INTO STUD VALUES(1,'AJAY','HOSUR')
1 row created.
```

16

```
SQL> /
Enter value for rollno: 2
Enter value for name: CHETAN
Enter value for address: MADURA
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&ADDRESS')
new 1: INSERT INTO STUD VALUES(2,'CHETAN','MADURA')
1 row created.
SQL> /
Enter value for rollno: 3
Enter value for name: DEEPAK
Enter value for address: GADAG
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&ADDRESS')
new 1: INSERT INTO STUD VALUES(3,'DEEPAK','GADAG')
1 row created.
SQL> /
Enter value for rollno: 4
Enter value for name: GANESH
Enter value for address: JSS
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&ADDRESS')
new 1: INSERT INTO STUD VALUES(4,'GANESH','JSS')
1 row created.
SQL> /
Enter value for rollno: 5
Enter value for name: JAY
Enter value for address: ARTS COLL
old 1: INSERT INTO STUD VALUES(&ROLLNO,'&NAME','&ADDRESS')
new 1: INSERT INTO STUD VALUES(5,'JAY','ARTS COLL')
1 row created.
SQL> SELECT * FROM STUD;
```

17

```
ROLLNO NAME ADDRESS
---------- ---------- ----------
1 AJAY HOSUR
2 CHETAN MADURA
3 DEEPAK GADAG
4 GANESH JSS
5 JAY ARTS COLL
SQL> CREATE TABLE RES(ROLLNO NUMBER(3)PRIMARY KEY,
2 S1 NUMBER(3),
3 S2 NUMBER(3),
4 TOT NUMBER(3),
5 PER NUMBER(3));
Table created.
SQL> DESC RES;
Name Null? Type
----------------------------------------- -------- ----------------------------
ROLLNO NUMBER(3)
```

```
S1 NUMBER(3)
S2 NUMBER(3)
TOT NUMBER(3)
PER NUMBER(3)
SQL> INSERT INTO
RES(ROLLNO,S1,S2)VALUES(&ROLLNO,&S1,&S2);
Enter value for rollno: 1
Enter value for s1: 50
Enter value for s2: 60
old 1: INSERT INTO
RES(ROLLNO,S1,S2)VALUES(&ROLLNO,&S1,&S2)
new 1: INSERT INTO RES(ROLLNO,S1,S2)VALUES(1,50,60)
1 row created.
SQL> /
Enter value for rollno: 2
Enter value for s1: 70
```

18

```
Enter value for s2: 80
old 1: INSERT INTO
RES(ROLLNO,S1,S2)VALUES(&ROLLNO,&S1,&S2)
new 1: INSERT INTO RES(ROLLNO,S1,S2)VALUES(2,70,80)
1 row created.
SQL> /
Enter value for rollno: 3
Enter value for s1: 60
Enter value for s2: 80
old 1: INSERT INTO
RES(ROLLNO,S1,S2)VALUES(&ROLLNO,&S1,&S2)
new 1: INSERT INTO RES(ROLLNO,S1,S2)VALUES(3,60,80)
1 row created.
SQL> /
Enter value for rollno: 4
Enter value for s1: 80
Enter value for s2: 70
old 1: INSERT INTO
RES(ROLLNO,S1,S2)VALUES(&ROLLNO,&S1,&S2)
new 1: INSERT INTO RES(ROLLNO,S1,S2)VALUES(4,80,70)
1 row created.
SQL> /
Enter value for rollno: 5
Enter value for s1: 90
Enter value for s2: 70
old 1: INSERT INTO
RES(ROLLNO,S1,S2)VALUES(&ROLLNO,&S1,&S2)
new 1: INSERT INTO RES(ROLLNO,S1,S2)VALUES(5,90,70)
1 row created.
SQL> SELECT * FROM RES;
ROLLNO S1 S2 TOT PER
---------- ---------- ---------- ---------- ----------
1 50 60
2
```

19

```
4 80 70
5 90 70
SQL> UPDATE RES SET TOT=S1+S2;
5 rows updated.
SQL> SELECT * FROM RES;
ROLLNO S1 S2 TOT PER
---------- ---------- ---------- ---------- ----------
1 50 60 110
2 70 80 150
3 60 80 140
4 80 70 150
5 90 70 160
SQL> UPDATE RES SET PER=TOT/2;
5 rows updated.
SQL> SELECT * FROM RES;
ROLLNO S1 S2 TOT PER
---------- ---------- ---------- ---------- ----------
1 50 60 110 55
2 70 80 150 75
3 60 80 140 70
4 80 70 150 75
5 90 70 160 80
```

```
SQL> COMMIT;
Commit complete.
SQL> SELECT STUD.ROLLNO,NAME,S1,S2,TOT,PER FROM STUD,RES
2 WHERE STUD.ROLLNO=RES.ROLLNO;
```

20

```
2 CHETAN 70 80 150 75
3 DEEPAK 60 80 140 70
4 GANESH 80 70 150 75
5 JAY 90 70 160 80
SQL> COMMIT;
Commit complete.
```

❖ **Example for linking table or setting relationship for staff and salary tables**

```
SQL>CREATE TABLE STAFF10(STNO NUMBER(3)PRIMARY KEY,
2 STNAME VARCHAR2(10)NOT NULL,
3 DESIG VARCHAR2(10));
Table created.
SQL> DESC STAFF10;
Name Null? Type
----------------------------------------- -------- ----------------------------
STNO NOT NULL NUMBER(3)
STNAME NOT NULL VARCHAR2(10)
DESIG VARCHAR2(10)
SQL> CREATE TABLE SAL(STNO NUMBER(3)PRIMARY KEY,
2 BASIC NUMBER(5),
3 HRA NUMBER(5),
4 DA NUMBER(5),
5 GROSS NUMBER(5),
6 PF NUMBER(5),
7 NETSAL NUMBER(5),
8 FOREIGN KEY (STNO)REFERENCES STAFF10);
Table created.
SQL> DESC SAL;
```

```
ROLLNO NAME S1 S2 TOT PER
---------- ---------- ---------- ---------- ---------- ----------
1 AJAY 50 60 110 55
```

21

```
Name Null? Type
----------------------------------------- -------- ----------------------------
STNO NOT NULL NUMBER(3)
BASIC NUMBER(5)
HRA  NUMBER(5)
DA NUMBER(5)
GROSS NUMBER(5)
PF NUMBER(5)
NETSAL NUMBER(5)
SQL> DESC STAFF10;
Name Null? Type
----------------------------------------- -------- ----------------------------
STNO NOT NULL NUMBER(3)
STNAME NOT NULL VARCHAR2(10)
DESIG VARCHAR2(10)
SQL> INSERT INTO STAFF10 VALUES(1,'AMIT','ACCT');
1 row created.
SQL> INSERT INTO STAFF10 VALUES(2,'BABU','CLERK');
1 row created.
SQL> INSERT INTO STAFF10 VALUES(3,'CHETAN','MANAGER');
1 row created.
SQL> SELECT * FROM STAFF10;
STNO STNAME DESIG
---------- ---------- ----------
1 AMIT ACCT
2 BABU CLERK
3 CHETAN MANAGER
SQL> COMMIT;
Commit complete.
```

22

```
SQL> INSERT INTO SAL(STNO,BASIC)VALUES(1,10000);
1 row created.
SQL> INSERT INTO SAL(STNO,BASIC)VALUES(2,12000);
1 row created.
SQL> INSERT INTO SAL(STNO,BASIC)VALUES(3,15000);
1 row created.
SQL> SELECT * FROM SAL;
```

```
STNO BASIC HRA DA GROSS PF NETSAL
---------- ---------- ---------- ---------- ---------- ---------- ----------
1 10000
2 12000
3 15000
SQL> UPDATE SAL SET HRA=BASIC*20/100;
3 rows updated.
SQL> SELECT * FROM SAL;
STNO BASIC HRA DA GROSS PF NETSAL
…… …… …… …… …… …… ……
1 10000 2000
2 12000 2400
3 15000 3000
SQL> UPDATE SAL SET DA=BASIC*20/100;
3 rows updated.
SQL> SELECT * FROM SAL;
STNO BASIC HRA DA GROSS PF NETSAL
---------- ---------- ---------- ---------- ---------- ---------- ----------
1 10000 2000 2000
2 12000 2400 2400
```

23

```
3 15000 3000 3000
SQL> UPDATE SAL SET GROSS=BASIC+HRA+DA;
3 rows updated.
SQL> SELECT * FROM SAL;
STNO BASIC HRA DA GROSS PF NETSAL
…… …… …… …… …… …… ……
1 10000 2000 2000 14000
2 12000 2400 2400 16800
3 15000 3000 3000 21000
SQL> UPDATE SAL SET PF=BASIC*12/100;
3 rows updated.
SQL> SELECT * FROM SAL;
STNO BASIC HRA DA GROSS PF NETSAL
---------- ---------- ---------- ---------- ---------- ---------- ----------
1 10000 2000 2000 14000 1200
2 12000 2400 2400 16800 1440
3 15000 3000 3000 21000 1800
SQL> UPDATE SAL SET NETSAL=GROSS-PF;
3 rows updated.
SQL> SELECT * FROM SAL;
STNO BASIC HRA DA GROSS PF NETSAL
…… …… …… …… …… …… ……
1 10000 2000 2000 14000 1200 12800
```

❖ **Create table emp with the following fields:**

```
SQL>CREATE TABLE EMP(
EMPNO NUMBER(4)PRIMARY KEY,
ENAME VARCHAR2(10),
JOB VARCHAR2(9),
MGR NUMBER(4),
```

24

```
HIREDATE    DATE,
SAL NUMBER(7,2),
COMM NUMBER(7,2),
DEPTNO NUMBER(2));
SQL> DESC EMP;
Name Null? Type
…… …… ……
EMPNO NOT NULL NUMBER(4)
ENAME VARCHAR2(10)
JOB VARCHAR2(9)
MGR NUMBER(4)
HIREDATE DATE
SAL NUMBER(7,2)
COMM NUMBER(7,2)
DEPTNO NUMBER(2)
```

Add 14 records to employee table (data is shown below)

```
SQL> SELECT * FROM EMP;
EMPNO ENAME JOB MGR HIREDATE SAL
COMM DEPTNO7369 SMITH CLERK 7902
17-DEC-80 900 20
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 2975 20
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
7876 ADAMS CLERK 7788 23-MAY-87 1100 20
7900 JAMES CLERK 7698 03-DEC-81 950 30
7902 FORD ANALYST 7566 03-DEC-81 3000 20
7934 MILLER CLERK 7782 23-JAN-82 1300 10
101 MATHEW CLERK 7777 01-JUN-10 6000 500 10
15 rows selected.
SQL> SET PAGESIZE 100;
```

25

```
SQL> SET LINESIZE 100;
SQL> /
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---
-------7369 SMITH CLERK 7902 17-DEC-80 900 20
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 2975 20
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
SQL> SELECT EMPNO,ENAME,SAL FROM EMP;
EMPNO ENAME SAL
---------- ---------- ----------
7369 SMITH 900
7499 ALLEN 1600
7521 WARD 1250
7566 JONES 2975
7654 MARTIN 1250
```

26

```
101 MATHEW 6000
15 rows selected.
SQL> SELECT EMPNO,ENAME,JOB FROM EMP;
EMPNO ENAME JOB
---------- ---------- ---------
7369 SMITH CLERK
7499 ALLEN SALESMAN
7521 WARD SALESMAN
7566 JONES MANAGER
7654 MARTIN SALESMAN
SQL> SELECT * FROM EMP WHERE SAL>=1000 AND SAL<=3000;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---
-------7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300
30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 2975 20
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
```

```
SQL> SELECT * FROM EMP WHERE SAL>=3000;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7788 SCOTT ANALYST 7566 19-APR-87 3000 20
7839 KING PRESIDENT 17-NOV-81 5000 10
7902 FORD ANALYST 7566 03-DEC-81 3000 20
101 MATHEW CLERK 7777 01-JUN-10 6000 500 10
SQL> SELECT * FROM EMP WHERE SAL<3000;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7369 SMITH CLERK 7902 17-DEC-80 900 20
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 2975 20
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
7876 ADAMS CLERK 7788 23-MAY-87 1100 20
7934 MILLER CLERK 7782 23-JAN-82 1300 10
11 rows selected.
SQL> SELECT * FROM EMP WHERE DEPTNO=10;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
7839 KING PRESIDENT 17-NOV-81 5000 10
7934 MILLER CLERK 7782 23-JAN-82 1300 10
101 MATHEW CLERK 7777 01-JUN-10 6000 500 10
SQL> SELECT * FROM EMP WHERE DEPTNO=30;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
7900 JAMES CLERK 7698 03-DEC-81 950 30
6 rows selected.
```

```
28
SQL> SELECT * FROM EMP WHERE JOB='CLERK';
7876 ADAMS CLERK 7788 23-MAY-87 1100 20
7900 JAMES CLERK 7698 03-DEC-81 950 30
7934 MILLER CLERK 7782 23-JAN-82 1300 10
101 MATHEW CLERK 7777 01-JUN-10 6000 500 10
SQL> SELECT * FROM EMP WHERE JOB='SALESMAN';
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
SQL> SELECT * FROM EMP WHERE JOB='SALESMAN' AND
SAL>=1500;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
SQL> SELECT * FROM EMP WHERE JOB='CLERK' AND DEPTNO=10;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7934 MILLER CLERK 7782 23-JAN-82 1300 10
101 MATHEW CLERK 7777 01-JUN-10 6000 500 10
SQL> SELECT * FROM EMP WHERE DEPTNO=10 OR DEPTNO=30;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
10 rows selected.
```

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

---------- ---------- --------- ---------- --------- ----------

```
29
SQL> SELECT * FROM EMP WHERE SAL BETWEEN 2000 AND 5000;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- 7566
JONES MANAGER 7839 02-APR-81 2975 20
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
7788 SCOTT ANALYST 7566 19-APR-87 3000 20
7839 KING PRESIDENT 17-NOV-81 5000 10
7902 FORD ANALYST 7566 03-DEC-81 3000 20
6 rows selected.
SQL> SELECT * FROM EMP WHERE COMM IS NULL;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7369 SMITH CLERK 7902 17-DEC-80 900 20
7566 JONES MANAGER 7839 02-APR-81 2975 20
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
7788 SCOTT ANALYST 7566 19-APR-87 3000 20
7839 KING PRESIDENT 17-NOV-81 5000 10
```

---------- ----------7369 SMITH CLE

```
7876 ADAMS CLERK 7788 23-MAY-87 1100 20
10 rows selected.
SQL> SELECT * FROM EMP WHERE NOT JOB='CLERK';
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---
-------7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300
30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 2975 20
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
SQL> SELECT * FROM EMP WHERE JOB!='CLERK';
```

```
30
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---
-------7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300
30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 2975 20
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
7788 SCOTT ANALYST 7566 19-APR-87 3000 20
SQL> SELECT * FROM EMP WHERE EMPNO
IN(7499,7654,7902);
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---
-------7902 FORD ANALYST 7566 03-DEC-81 3000 20
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
SQL> SELECT * FROM EMP WHERE ENAME
IN('ALLEN','FORD');
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---
-------7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300
30
7902 FORD ANALYST 7566 03-DEC-81 3000 20
SQL> SELECT * FROM EMP WHERE JOB
IN('CLERK','SALESMAN');
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---
-------7369 SMITH CLERK 7902 17-DEC-80 900 20
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
9 rows selected.
SQL> SELECT * FROM EMP WHERE ENAME LIKE 'S%';
```

```
31
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

7369 SMITH CLERK 7902 17-DEC-80 900 20
7788 SCOTT ANALYST 7566 19-APR-87 3000 20
SQL> SELECT * FROM EMP WHERE ENAME LIKE '%S';
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ---
-------7566 JONES MANAGER 7839 02-APR-81 2975 20
7876 ADAMS CLERK 7788 23-MAY-87 1100 20
7900 JAMES CLERK 7698 03-DEC-81 950 30
SQL> SELECT * FROM EMP WHERE ENAME LIKE'%A%';
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
7876 ADAMS CLERK 7788 23-MAY-87 1100 20
7900 JAMES CLERK 7698 03-DEC-81 950 30
101 MATHEW CLERK 7777 01-JUN-10 6000 500 10
8 rows selected.
SQL> SELECT * FROM EMP WHERE ENAME LIKE 'K%' AND SAL LIKE
'5%';
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7839 KING PRESIDENT 17-NOV-81 5000 10
SQL> SELECT * FROM EMP WHERE ENAME='ALLEN';
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
SQL> SELECT EMPNO,ENAME,JOB,SAL,COMM FROM EMP WHERE
ENAME='JAMES';
```

```
SQL> SELECT EMPNO,ENAME,SAL,COMM FROM EMP;
EMPNO ENAME SAL COMM
---------- ---------- ---------- ----------
7369 SMITH 900
7499 ALLEN 1600 300
7521 WARD 1250 500
7566 JONES 2975
7654 MARTIN 1250 1400
7698 BLAKE 2850
7782 CLARK 2450
7788 SCOTT 3000
7839 KING 5000
7844 TURNER 1500 0
7876 ADAMS 1100
15 rows selected.
SQL> SELECT EMPNO,ENAME,SAL,SAL*12 "ANNUAL SALARY"
FROM EMP;
EMPNO ENAME SAL ANNUAL SALARY
---------- ---------- ---------- -------------
7369 SMITH 900 10800
7499 ALLEN 1600 19200
7521 WARD 1250 15000
7566 JONES 2975 35700
7654 MARTIN 1250 15000
15 rows selected.
SQL> SELECT
EMPNO,ENAME,SAL,COMM,SAL+NVL(COMM,0)"MONTHLY SAL"
FROM EMP;
```

33

```
EMPNO ENAME SAL COMM MONTHLY SAL
---------- ---------- ---------- ---------- -----------
7369 SMITH 900 900
7499 ALLEN 1600 300 1900
7521 WARD 1250 500 1750
7566 JONES 2975 2975
7654 MARTIN 1250 1400 2650
7698 BLAKE 2850 2850
7782 CLARK 2450 2450
15 rows selected.
```

❖ **Example for Aggregate functions**

```
SQL> SELECT MAX(SAL)FROM EMP;
MAX(SAL)
----------
6000
SQL> SELECT MIN(SAL)FROM EMP;
MIN(SAL)
----------
900
SQL> SELECT AVG(SAL)FROM EMP;
AVG(SAL)
----------
2341.66667
SQL> SELECT SUM(SAL)FROM EMP;
SUM(SAL)
----------
35125
SQL> SELECT COUNT(*)FROM EMP;
COUNT(*)
----------
```

34

```
15
SQL> SELECT COUNT(COMM)FROM EMP;
COUNT(COMM)
-----------
5
```

❖ **Dual is a system level table**

```
SQL> DESC DUAL;
Name Null? Type
----------------------------------------------------- -------- ------------------------
------------
DUMMY  VARCHAR2(1)
SQL> SELECT * FROM DUAL;
D
-
X
```

32

```
EMPNO ENAME JOB SAL COMM
--------- --------- --------- --------- ------
----7900 JAMES CLERK 950
SQL> COMMIT;
Commit complete.
```

```
SQL> SELECT ENAME,LOWER(ENAME)FROM EMP;
ENAME LOWER(ENAM

SMITH smith
ALLEN allen
WARD ward
JONES jones
MARTIN martin
BLAKE blake
CLARK clark
SCOTT scott
KING king
TURNER turner
ADAMS adams
JAMES james
FORD ford
```

35

```
MILLER miller
MATHEW mathew
15 rows selected.
```

❖ **Example for String functions**

```
SQL> SELECT LOWER('KEONICS')FROM DUAL;
LOWER('
-------
keonics
SQL> SELECT ENAME,UPPER(ENAME)FROM EMP;
ENAME UPPER(ENAME)
SMITH SMITH
ALLEN ALLEN
WARD WARD
JONES JONES
MARTIN MARTIN
BLAKE BLAKE
CLARK CLARK
SCOTT SCOTT
KING KING
TURNER TURNER
ADAMS ADAMS
JAMES JAMES
FORD FORD
MILLER MILLER
MATHEW MATHEW
15 rows selected.
SQL> SELECT UPPER('keonics')FROM DUAL;
```

36

```
SQL> SELECT ENAME,INITCAP(ENAME)FROM EMP;
ENAME INITCAP(EN
---------- ----------
SMITH Smith
ALLEN Allen
WARD Ward
JONES Jones
MARTIN Martin
BLAKE Blake
CLARK Clark
SCOTT Scott
KING King
TURNER Turner
ADAMS Adams
JAMES James
FORD Ford
MILLER Miller
MATHEW Mathew
15 rows selected.
SQL> SELECT INITCAP('KEONICS')FROM DUAL;
INITCAP
-------
Keonics
SQL> SELECT ENAME,LENGTH(ENAME)FROM EMP;
ENAME LENGTH(ENAME)
---------- --------------
```

```
SMITH 5
ALLEN 5
WARD 4
JONES 5
MARTIN 6
BLAKE 5
CLARK 5
```

```
37
SCOTT 5
KING 4
TURNER 6
ADAMS 5
JAMES 5
FORD 4
MILLER 6
MATHEW 6
15 rows selected.
SQL> SELECT LENGTH('KEONICS')FROM DUAL;
LENGTH('KEONICS')
-----------------
7
SQL> SELECT JOB,REPLACE(JOB,'SALESMAN','MKTG')FROM EMP;
JOB REPLACE(JOB,'SALESMAN','MKTG')
..... .....
CLERK CLERK
SALESMAN MKTG
SALESMAN MKTG
MANAGER MANAGER
SALESMAN MKTG
MANAGER MANAGER
MANAGER MANAGER
ANALYST ANALYST
PRESIDENT PRESIDENT
SALESMAN MKTG
CLERK CLERK
CLERK CLERK
ANALYST ANALYST
CLERK CLERK
CLERK CLERK
15 rows selected.
```

```
38
SQL> SELECT SAL,RPAD(SAL,10,'*')FROM EMP;
SAL RPAD(SAL,1
---------- ----------
900 900*******
1600 1600******
1250 1250******
SQL> SELECT SAL,LPAD(SAL,10,'?')FROM EMP;
SAL LPAD(SAL,1
---------- ----------
???????900
??????1600
??????1250
SQL> SELECT SUBSTR('INDIA',3,5)FROM DUAL;
SUB
---
DIA
```

❖ **Example for Numerical/mathematical functions**

```
SQL> SELECT ROUND(12.5866656,2)FROM DUAL;
ROUND(12.5866656,2)
-------------------
12.59
SQL> SELECT ROUND(12.50)FROM DUAL;
ROUND(12.50)
------------
13
SQL> SELECT ROUND(12.49)FROM DUAL;
ROUND(12.49)
------------
12
```

```
39
SQL> SELECT CEIL(12.99)FROM DUAL;
CEIL(12.99)
.....
13
SQL> SELECT CEIL(12.1)FROM DUAL;
CEIL(12.1)
----------
13
SQL> SELECT FLOOR(12.99)FROM DUAL;
FLOOR(12.99)
------------
12
SQL> SELECT SQRT(16)FROM DUAL;
SQRT(16)
----------
4
SQL> SELECT POWER(2,3)FROM DUAL;
POWER(2,3)
----------
8
SQL> SELECT SIN(90)FROM DUAL;
SIN(90)
----------
.893996664
SQL> SELECT COS(60)FROM DUAL;
COS(60)
----------
-.95241298
```

```
40
SQL> SELECT TAN(45)FROM DUAL;
TAN(45)
----------
1.61977519
```

❖ **Example for Date functions**

```
SQL> SELECT SYSDATE FROM DUAL;
SYSDATE
---------
24-NOV-15
SQL> ALTER SESSION SET NLS_DATE_FORMAT='DD/MM/YYYY';
Session altered.
SQL> SELECT * FROM EMP;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- ---------- ---------- ---------- ----------
7369 SMITH CLERK 7902 17/12/1980 900 20
7499 ALLEN SALESMAN 7698 20/02/1981 1600 300 30
7521 WARD SALESMAN 7698 22/02/1981 1250 500 30
7566 JONES MANAGER 7839 02/04/1981 2975 20
7654 MARTIN SALESMAN 7698 28/09/1981 1250 1400 30
7698 BLAKE MANAGER 7839 01/05/1981 2850 30
15 rows selected.
SQL> SELECT TO_CHAR(SYSDATE,'DD/MM/YY')FROM DUAL;
TO_CHAR(
--------
24/11/15
SQL> SELECT TO_CHAR(SYSDATE,'DD-MON-YY')FROM DUAL;
TO_CHAR(S
---------
24-NOV-15
```

```
41
SQL> SELECT TO_CHAR(SYSDATE,'DAY-MONTH-YEAR')FROM
DUAL;
TO_CHAR(SYSDATE,'DAY-MONTH-YEAR')
---------------------------------------------------------
TUESDAY -NOVEMBER -TWENTY FIFTEEN
SQL> SELECT TO_CHAR(SYSDATE,'DY-MON-YY')FROM DUAL;
TO_CHAR(SY
----------
TUE-NOV-15
SQL> SELECT TO_CHAR(SYSDATE,'YEAR')FROM DUAL;
TO_CHAR(SYSDATE,'YEAR')
-----------------------------------------
TWENTY FIFTEEN
SQL> SELECT TO_CHAR(SYSDATE,'YY')FROM DUAL;
TO
--
15
SQL> SELECT TO_CHAR(SYSDATE,'MONTH')FROM DUAL;
TO_CHAR(S
.....
```

```
NOVEMBER
SQL> SELECT TO_CHAR(SYSDATE,'DAY')FROM DUAL;
TO_CHAR(S
---------
TUESDAY
SQL> SELECT TO_CHAR(SYSDATE,'DY')FROM DUAL;
TO_
---
TUE
```

```
42
SQL> SELECT TO_CHAR(SYSDATE,'HH:MM:SS')FROM DUAL;
TO_CHAR(
--------
04:11:03
SQL> COMMIT;
Commit complete.
SQL> CREATE TABLE EMP_NEW AS SELECT * FROM EMP;
Table created.
SQL> DESC EMP_NEW;
Name Null? Type
----------------------------------------------------- -------- ------------------------
------------
EMPNO NUMBER(4)
ENAME VARCHAR2(10)
JOB VARCHAR2(9)
MGR NUMBER(4)
HIREDATE DATE
SAL NUMBER(7,2)
COMM NUMBER(7,2)
DEPTNO NUMBER(2)
SQL> SELECT * FROM EMP_NEW;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
..... ..... ..... ..... ..... ..... ..... .....
7369 SMITH CLERK 7902 17/12/1980 900 20
7499 ALLEN SALESMAN 7698 20/02/1981 1600 300 30
7521 WARD SALESMAN 7698 22/02/1981 1250 500 30
7566 JONES MANAGER 7839 02/04/1981 2975 20
7654 MARTIN SALESMAN 7698 28/09/1981 1250 1400 30
7698 BLAKE MANAGER 7839 01/05/1981 2850 30
7782 CLARK MANAGER 7839 09/06/1981 2450 10
7788 SCOTT ANALYST 7566 19/04/1987 3000 20
15 rows selected.
```

```
43
SQL> CREATE TABLE EMP5 AS SELECT * FROM EMP WHERE
EMPNO=0;
Table created.
SQL> DESC EMP5
Name Null? Type
..... ..... .....
------------
EMPNO NUMBER(4)
ENAME VARCHAR2(10)
JOB VARCHAR2(9)
MGR NUMBER(4)
HIREDATE DATE
SAL NUMBER(7,2)
COMM NUMBER(7,2)
DEPTNO NUMBER(2)
SQL> SELECT * FROM EMP5;
no rows selected
❖ Creating duplicate table
SQL> CREATE TABLE EMP6 AS SELECT * FROM EMP WHERE
JOB='CLERK';
Table created.
SQL> SELECT * FROM EMP6;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
..... ..... ..... ..... ..... ..... ..... .....
7369 SMITH CLERK 7902 17/12/1980 900 20
7876 ADAMS CLERK 7788 23/05/1987 1100 20
7900 JAMES CLERK 7698 03/12/1981 950 30
7934 MILLER CLERK 7782 23/01/1982 1300 10
101 MATHEW CLERK 7777 01/06/2010 6000 500 10
❖ Creating duplicate table
SQL> CREATE TABLE EMP7 AS SELECT * FROM EMP WHERE
DEPTNO=20;
```

```
Table created.
```

```
44
SQL> DESC EMP7
Name Null? Type
----------------------------------------------------- -------- ------------------------
------------
EMPNO NUMBER(4)
ENAME VARCHAR2(10)
JOB VARCHAR2(9)
MGR NUMBER(4)
HIREDATE DATE
SAL NUMBER(7,2)
COMM NUMBER(7,2)
DEPTNO NUMBER(2)
SQL> SELECT * FROM EMP7;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- ---------- ---------- ---------- ----------
7369 SMITH CLERK 7902 17/12/1980 900 20
7566 JONES MANAGER 7839 02/04/1981 2975 20
7788 SCOTT ANALYST 7566 19/04/1987 3000 20
7876 ADAMS CLERK 7788 23/05/1987 1100 20
7902 FORD ANALYST 7566 03/12/1981 3000 20
❖ Creating duplicate table with few fields of the old table
SQL> CREATE TABLE EMP8 AS SELECT EMPNO,ENAME,JOB,SAL
FROM EMP;
Table created.
SQL> DESC EMP8
Name Null? Type
----------------------------------------------------- -------- ------------------------
------------
EMPNO NUMBER(4)
ENAME VARCHAR2(10)
JOB VARCHAR2(9)
SAL NUMBER(7,2)
SQL> SELECT * FROM EMP8;
```

```
45
EMPNO ENAME JOB SAL
---------- ---------- --------- ----------
7369 SMITH CLERK 900
7499 ALLEN SALESMAN 1600
7521 WARD SALESMAN 1250
7566 JONES MANAGER 2975
7654 MARTIN SALESMAN 1250
7698 BLAKE MANAGER 2850
15 rows selected.
❖ Creating duplicate table with few fields of the old table
SQL> CREATE TABLE EMP9 AS SELECT EMPNO,ENAME,JOB,SAL
FROM EMP
WHERE DEPTNO=30;
Table created.
SQL> DESC EMP9
Name Null? Type
----------------------------------------------------- -------- ------------------------
------------
EMPNO NUMBER(4)
ENAME VARCHAR2(10)
JOB VARCHAR2(9)
SAL NUMBER(7,2)
SQL> SELECT * FROM EMP9;
EMPNO ENAME JOB SAL
---------- ---------- --------- ----------
7499 ALLEN SALESMAN 1600
7521 WARD SALESMAN 1250
7654 MARTIN SALESMAN 1250
7698 BLAKE MANAGER 2850
7844 TURNER SALESMAN 1500
7900 JAMES CLERK 950
6 rows selected.
```

```
46
❖ Creating view
SQL> CREATE VIEW EMP10 AS SELECT * FROM EMP;
View created.
SQL> DESC EMP10
Name Null? Type
..... ..... .....
```

47

SQL> SELECT * FROM EMP11;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

7499 ALLEN SALESMAN 7698 20/02/1981 1600 300 30
7521 WARD SALESMAN 7698 22/02/1981 1250 500 30
7654 MARTIN SALESMAN 7698 28/09/1981 1250 1400 30
7844 TURNER SALESMAN 7698 08/09/1981 1500 0 30

❖ **Creating view**

SQL> CREATE VIEW EMP12 AS SELECT * FROM EMP WHERE EMPNO=0;
View created.
SQL> DESC EMP12;
Name Null? Type

EMPNO NOT NULL NUMBER(4)
ENAME VARCHAR2(10)
JOB VARCHAR2(9)
MGR NUMBER(4)
HIREDATE DATE
SAL NUMBER(7,2)
COMM NUMBER(7,2)
DEPTNO NUMBER(2)
SQL> SELECT * FROM EMP12;
no rows selected

48

❖

❖

❖ **Connecting to admin account and Creating user accounts**

SQL> CONNECT SYSTEM/MANAGER;
Connected.
SQL> CREATE USER KEONICS IDENTIFIED BY WELCOME;
User created.

SQL> GRANT CREATE ANY TABLE TO KEONICS;
Grant succeeded.
SQL> GRANT CREATE SESSION TO KEONICS;

Grant succeeded.
SQL> GRANT DROP ANY TABLE TO KEONICS;
Grant succeeded.
SQL> GRANT RESOURCE TO KEONICS;
Grant succeeded.
SQL> COMMIT;
Commit complete.

❖ **Login to user account**

SQL> CONNECT KEONICS/WELCOME;
Connected.
SQL> CREATE TABLE STUD(ROLLNO NUMBER(6),
2 NAME VARCHAR2(10));
Table created.
SQL> DESC STUD;

EMPNO NOT NULL NUMBER(4)
ENAME VARCHAR2(10)
JOB VARCHAR2(9)
MGR NUMBER(4)
HIREDATE DATE
SAL NUMBER(7,2)
COMM NUMBER(7,2)
DEPTNO NUMBER(2)
SQL> SELECT * FROM EMP10;
EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

7369 SMITH CLERK 7902 17/12/1980 900 20
7499 ALLEN SALESMAN 7698 20/02/1981 1600 300 30
7521 WARD SALESMAN 7698 22/02/1981 1250 500 30
7566 JONES MANAGER 7839 02/04/1981 2975 20
7654 MARTIN SALESMAN 7698 28/09/1981 1250 1400 30
7698 BLAKE MANAGER 7839 01/05/1981 2850 30
7782 CLARK MANAGER 7839 09/06/1981 2450 10
7788 SCOTT ANALYST 7566 19/04/1987 3000 20
7839 KING PRESIDENT 17/11/1981 5000 10
15 rows selected.

❖ **Creating view**

SQL> CREATE VIEW EMP11 AS SELECT * FROM EMP WHERE JOB='SALESMAN';
View created.

Qp adca

1) Tft stands for _____?
   a) Thin film transformer
   b) Thin film translator
   c) **Thin film transistor**
   d) All the above
2) Files stores in _____?
   a) Drives
   b) Folder
   c) **Both a & b**
   d) None of the above
3) To select multiple files hold _____ button?
   a) Shift
   b) Alt +shift
   c) **Control**
   d) Control +shift
4) Short cut key to hyperlink is _____?
   a) Shift +k
   b) **Ctrl +k**
   c) Alt +k
   d) Alt +ctrl +k
5) Internet is a _____?
   a) Software program
   b) Operating system
   c) **Network of networks**
   d) None of those
6) "ctrl + p" is _____ in ms-word?
   a) Open paragraph dialog box
   b) Open page format dialog box
   c) Open save dialog box
   d) **Open print dialog box**
7) The feature of word that automatically adjusts the amount of space between certain combinations of characters so that an entire word looks more evenly spaced what is that feature called? a) Spacing
   b) Scaling
   c) **Kerning**
   d) Positioning
8) End key _____ in ms- word
   a) **Moves the cursor end of the line**
   b) Moves the cursor end of the document
   c) Moves the cursor end of the paragraph
   d) Moves the cursor end of the screen
9) The default header for a worksheet is _____ in excel?
   a) Username
   b) Date and time
   c) Sheet tab name
   d) **None**
10) If you want to type the text in a1 to g1 range cells, option used is_____? a) **Merge cells**
   b) More cells
   c) Insert cells
   d) All cells
11) If you press _____ the cell accepts your typing as its contents in excel? a) **Enter**
   b) Ctrl + enter
   c) Esc
   d) Insert
12) What happens if you select first and second slide and then click on new slide button on toolbar?
   a) A new slide is inserted as first slide in presentation
   b) A new slide is inserted as second slide in presentation
   c) **A new slide is inserted as third slide in presentation**
   d) None of above
13) _____ controls all the main slide control tasks for your presentation?
   a) **Task pane**
   b) Task bar
   c) Control panel
   d) None of above
14) Presentations can contain bulleted and numbered items are _____?
   a) **Home**
   b) Insert
   c) Format
   d) Help
15) Usually to run and stop the nudi key is?
   a) Shift
   b) Alt
   c) **Scroll lock**
   d) Num lock
16) Nudi can be used for dynamic font embedding purposes?
   a) False
   b) **True**
   c) A&b
   d) None
17) Isp stands for _____?
   a) **Internet service provider**
   b) Internet service publisher
   c) Internet server provider
   d) All the above
18) A word processor would most?
   a) Keep an account of money spent
   b) Do a computer search in media center
   c) Maintain an inventory
   d) **Type a biography**
19) Pagemaker publication has _____ default number of pages
   a) **1 page**
   b) 2 pages
   c) 3 pages
   d) 4 pages
20) To set up a new document in pagemaker click on _____?
   a) **File → new**
   b) Edit → new
   c) Layout → new
   d) Element → new
21) _____ is the short key to view 100% in pagemaker
   a) **Ctrl +1**
   b) Shift +1
   c) Ctrl +2
   d) Alt +2
22) In coreldraw shortcut key to activate order option is alt+ o?
   a) True
   b) **False**
   c) A & b
   d) None
23) _____ shortcut key for magic wand tool in photoshop?
   a) V
   b) **W**
   c) A
   d) B
24) To make wood frame for an image which option is used in photoshop?
   a) Contact sheet
   b) **Emboss**
   c) Batch
   d) Find eggs
25) To set the background color on the work space _____ key a used in photoshop.
   a) Shift +back -space
   b) Alt +
   c) Alt –
   d) **Ctrl+ back space**
26) Pdf creator is used to create pdf files by converting word. Excel, powerpoint?
   a) **Yes**
   b) No
   c) May be
   d) Never
27) Pdf creator supports access to its functionality via an?
   a) Multiple
   b) Activex
   c) **Both a & b**
   d) None of the above
28) Presentation-graphics software is typically used to prepare a?
   a) **Series of on-screen "slides"**
   b) Web animation
   c) Movie for recording on 35 mm film
   d) Music video
29) which technology is 3-d graphics software based largely on?
   a) **Object-oriented graphics**
   b) Presentation graphics
   c) Bitmapped graphics
   d) Photographic image-editing software
30) Which kind of technology are photographic image-editing programs lafgely based on?
   a) Presentation graphics
   b) Bitmapped graphics
   c) Quantitative graphics
   d) **Cad/cam graphics**
31) For nonlinear video editing, what are video and audio clips stored on?
   a) Tape
   b) Dvd
   c) Cd
   d) **Hard disk(s)**
32) Correct html to left align the content inside a table cell is
   a) <tdleft>
   b) <td raligh= "left">
   c) **<td align = "left">**
   d) <td left align>
33) what does the .com domain represents?
   a) Education domain
   b) **Commercial domain**
   c) Network
   d) None of the above
34) text within ... tag is displayed as
   a) Bold
   b) **Italic**
   c) List
   d) Indented
35) the common element which describe the web page is?
   a) Heading
   b) Paragraph
   c) List

d) **All of these**

36) What are empty elements and is it valid?
a) No, there is no such terms as empty element
b) **Empty elements are element with no data**
c) No, it is not valid to use empty element
d) None of these

37) Which attribute is used to name an element uniquely?
a) Class
b) **Id**
c) Dot
d) All of these

38) Http stands for
a) Hypertext markup language
b) Hypertext transfer para
c) Home text transfer protocol
d) **Hypertext transfer protocol**

39) Which of the following property changes the style of top border?
a) Border-bottom-style
b) **Border-top-style**
c) Border-left-style
d) Border-right-style

40) Which of the following property is used to control the position of an image in the background? a) Background-color
b) **Background-image**
c) Background-repeat
d) Background-position

41) _____ JavaScript is also called server-side javascript.
a) Microsoft
b) **Navigator**
c) Livewire
d) Native

42) C language has been developed by
a) Martin Richards
b) Bijarne stroustrup
c) **Dennis ritche**
d) Ken Thompson

43) The formatted input function is_____in c
a) Print f()
b) **Scan f()**
c) Clrscr()
d) Getch()

44) In c language _____is the format specifies of strings?
a) %f
b) **%s**
c) %d
d) %c

45) Which of the following is false in c?
a) **Keywords can be used as variable names**
b) Variable names can contain a digit
c) Variable names do not contain a blank space
d) Capital letters can be used in variables

46) In c, structures can be used _____
a) **To hold different data types**
b) Have pointers to structures
c) To assign to one another
d) All of above

47) Do while loop is also called as_____ in c
a) Post present loop
b) Last test loop
c) Post final loop
d) **Posttest loop**

48) If the two strings are identical, then
a) 1
b) **0**
c) Both a & b
d) None of the above

49) Which of the following is not a built-in data type
a) Int
b) **Text**
c) Bool
d) Long int

50) C++ library functions are defined in _____header file.
a) Stdio.h
b) Iostream.h
c) Their specific
d) **Stdlib.h**

51) An optical input device that interprets pencil marks on paper media is _____?
a) **O.m.r**
b) Punch card reader
c) Printer
d) Cable

52) Shortcut key to rename files &folders is _____?
a) F5
b) F3
c) **F2**
d) F4

53) Windows is a _____?
a) Single user operating system
b) Multiuser operating system
c) Both a & b
d) **None of the above**

54) In c, the data type of character is?
a) Int
b) Float
c) String
d) **Char**

55) Which of the following data types are available in c?
a) Number, text, string, variant
b) Number, Date, Memo, text
c) **Int, float, char, double**
d) All the above

56) Which of the following property specifies an image for the marker rather than a bullet point or number?
a) **List-style-type**
b) List-style-position
c) List-style-image
d) List-style

57) Scripting language are?
a) High level programming language
b) Assembly level programming language
c) **Machine level programming language**
d) none of these

58) Which of the following property is used to control the scrolling of an image in the background? a) Background-attachment
b) Background
c) **Background-repeat**
d) Background-position

59) When should you use path along with file name of picture in img tag?
a) Path is optional and not necessary
b) **When the location of image file and html file are in different folder**
c) When image file and html file both are on same location
d) Path is always necessary when inserting image 60) What tag is used to list?
a) **li**
b) ol
c) ul
d) None of above

61) The common element which describe the web page, is?
a) Heading
b) Paragraph
c) List
d) All of these

62) How can you make a bulleted list?
a) <dl>
b)
c) <list>
d) **<ul>**

63) Http stands for
a) Hypertext markup language
b) Hypertext transfer para
c) Home text transfer protocol
d) **Hypertext transfer protocol** 64) What is the full form of tcp / lp?
a) **Transmission control protocol/internet protocol**
b) telephone call protocol/international protocol
c) Transport control protocol/internet protocol
d) None of the above

65) The attribute, which define the relationship between current document and hrefed url is a) Rel
b) Url
c) **Rev**
d) All of these

66) Presentation –graphics software is typically used to prepare a
a) **Series of on- screen "slides"**
b) Web animation
c) Movie for recording on 35 mm film
d) Music video

67) For nonlinear video editing, what are video and audio clips stored on?
a) Tape
b) DVD
c) CD
d) **Hard disk(s)**

68) Which kind of technology are photographic image –editing programs largely based on?
a) Presentation graphics
b) **bitmapped graphics**
c) Quantitative graphics
d) CAD/CAM graphics

69) Which technology is 3-D graphics software based largely on?
a) **object-oriented graphics**
b) Presentation graphics
c) Bitmapped graphics
b) None of these