UNIT – 1

- Introduction to web technology
- HTML
- types of HTML tags
- basic Structure of HTML
- Web design principles
- HTML attributes
- styles
- Hypertext
- Formatting text
- Forms & formulating instructions & formulation elements
- Commenting code
- Back grounds
- Images Hyperlinks
- Lists
- Tables
- Frames

INTRODUCTION

INTERNET : The internet is often used to indicate the community of users and computers collectively . The internet is a network of networks consists of many computers called servers or clients of clients. The internet is also a very large distribution system and enables users to make use of services such as the world wide web.

WEB TERMINOLOGIES CONCEPTS :

1.**PAGE OR WEB PAGE :** A file that can be read over the world wide web.

2.**WEB PAGES :** The global collection of domain clients associated with and accessible wide the World Wide Web(WWW).

3.**HYPERLINK :** A string of clickable text or clickable graphic that points to another web page or document .When the hyperlink is selected, another webpage is requested retrieved and rendered by the browser.

4.HYPERTEXT: Web pages that have hyperlink to other pages and links to other text.

5.BROWSER: A software tool used to view web pages, read e-mails and etc. Browsers are also called as Web Clients. The available Web browsers in the market are as follows.

a. Internet explorer
b. Netscape Navigator
c. Google chrome
d. Mozilla fire fox
e. Opera

6)MULTIMEDIA: Information in the form of graphics, audio, video or movies. A multi media document contains a media . Element other than just plain text.

7.WEB SITE : An entity on the internet that publishes Web pages. A web site typically has a computer serving Web pages. For example : WWW.Gmail.com.

8.WEB SERVER: A computer that having all the information of the Web pages of the Website.

9.WEB MASTER: A person who maintains and creates Web pages for a business organization or university.

10)ISP (Internet Service Provider): An ISP or Internet service is a company that provides you with a point of access to the internet . When you connect to ISP, computer becomes an extension of the internet. There are many internet service providers available in India.

Example : BSNL, Reliance, Airtel etc.

11.URL (Uniform Resource locater): The address of the Web page being displayed is shown under the tool bar in the location area of the browser window . If the browser is successful in finding the page, the browser will display it.

Example : WWW.Avanthi.com

HTML(HYPER TEXT MARK UP LANGUAGE)

Q1) What is HTML ?

A)HTML stands for "Hyper Text Mark up language" immensely used in developing Web pages.

This is also called as publishing Language used by the World Wide Web.

HISTORY OF HTML :

1.HTML was originally developed by **Tim Berners –Lee** at CERN and popularized by the mosaic browser developed at NCSA.

2.HTML 2.0 was developed under the aegis of the Internet Engineering Task Force (IETF) in November 1994.

3.In 1995, HTML 2.0 was codified common practice and proposed for richer versions to HTML 3.0.

4. Next version is HTML 3.2 in January 1997 by the efforts of World Wide Web HTML working group.

5.HTML has been developed with the vision that all manner of devices should be able to used to retrieve the information of the Web.

6.HTML 4.0 extends HTML with mechanisms for style shades, scripting, frames etc.

7. HTML 4.01 is a version which corrects errors and makes some changes in the web pages.

8. **HTML 5.0** is a new version after HTML 4.01 with all the options of entertainment and advanced scripting languages.

HTML are programs used to :

1. Publish and retrieve online information via hypertext line at the click of a button.
2. Publish online documents with headings text tables, list photos etc.
3. Include spreadsheets, video clips , audio clips and other applications directly in their documents.
4. Publish a Webpage on Website.
5. Design forms for conducting transactions with remote services, for use in searching of information, making reservations, ordering products etc.

Q2. Explain types of tags in HTML ?

A) HTML is composed of tags. HTML tags always enclosed in angular braces(< >) and are case insensitive. That is it does not matter whether you type them in upper or lower case.

Tags typically occur in Begin – End pairs. These pairs are in the form

Syntax:- <tag name> .....</tag name>

Here opening tag <tag> indicates the beginning of a tag pair and where as closing tag </tag> indicates the end of the tag.

Document tags : Documents tag means the tags which divide a Webpage into its basic sections, such as the header information and another part of the page which contains the displayed text and graphics which is called as body section.

Tags are of two types

1.Paired tags.

2.Unpaired tags (or) singular tags :

1. Paired Tags : A tag is said to be a paired tag , if it has a closing tag. The effect of the text will be from the beginning of the tag to the end of the tag. The opening tag activates the effect and closing tag turns off the effect.

Syntax: <tag name> _ _ _ _ _ </tag name>

Example : <HTML>Welcome to avanthi </HTML>

2) Unpaired tags (or) Singular tags : It is also called as stand alone tag does not require any closing tag.

Syntax: <tag name>

Example :
. It will insert a line bunch in the Web page. This tag does not require any closing tag.

Q3. Explain HTML program structure with example?

A. Structure of an HTML Program : Every HTML program has a rigid structure. The entire page is enclosed with <HTML> and </HTML> tags within these two tags two distinct sections are created using the <head> & </head> tags and the <body> & </body> tags.

HTML document is divided into two sections .

(i) Document head

(ii) Document body

i) DOCUMENT HEAD OR HEAD SECTION:-

The information placed in this section is essential to the inner workings of the document and has nothing to do with the content of the document. A tag <title> _ _ _ </title> is placed in this section. Information which is placed in the title tag is displayed on the browser.

Syntax : <Head> _ _ _ </HEAD>

Example : < Head >

<title> Hello </title >

</Head>

ii) DOCUMENT BODY : -

The tags used to indicate the start and end of the main body of the textual information are

Syntax : - < Body > _ _ _ <Body>

Example : <Body>

Welcome to HTML

</ Body>

The Body tag has three important attributes.

(i) **BGCOLOR :-** It changes the default background whatever color is specified with this tag. The user specify a color by a name or its equivalent Hexadecimal number.

Syntax:<Body BGColor = “Any color”>

Example : <Body BGcolor = “Blue”>

(ii) **BACKGROUND :-** It specifies the name of image file that will be used as the background of the documents.

Syntax :- < Body Background = “file name”>

Example : <Body Background = “XYZ.JPG”>

We can take images from C driver ex:=C:\abc.gif

(iii) **TEXT :** It changes the body text color from its default value to the color specified with this attribute.

Syntax: < Body text = “Any color”>

Example : < Body text = “blue”>

Write a sample program to display Welcome to HTML World with BG color pink and text color is blue : ?

<HTML>

< Head >

<title> hello </title>

</Head>

<Body BG color = “pink”, Text =”Blue” >

Welcome to HTML World

</Body>

</HTML>

Write a HTML program on the information on Internet atleast five lines with BG color blue and text color is orange.?

<HTML>

<Head>

<title>Welcome to HTML </title>

</Head>

<Body BG color = “blue”, Text= “orange” >

Internet is often used to indicate the community of users and computers collectively. The internet is a network of networks consists of many computers called servers or clients.

</Body>

</HTML>

Q4) What is a header tag in HTML?

A) HEADER TAG OR HEADINGS: - Heading are used to divide sections of text.HTML defines 6 lines of headings. The Number indicates heading line from 1 to 6 i.e., <H1> to <H6>. This heading creates a text in bold font of various sizes <H1> creates a largest text and <H6> creates the smallest one.

Syntax : <H1 >.....</H1>

<H2>.....</H2>

<H3>.....</H3>

<H4>.....</H4>

<H5>.....</H5>

<H6>.....</H6>

Example : <H1 >INTERNET</H1>

<H2>INTERNET</H2>

<H3>INTERNET</H3>

<H4>INTERNET</H4>

<H5>INTERNET</H5>

<H6>INTERNET</H6>

This tag contains a attribute called "ALIGN" which set said to left/ center / right / justify.

Write a program using heading tags for displaying friends, name at all 6 levels.?

<HTML>

<Head>

<title> friends </title>

</Head>

<Body BG color = "pink", text = "Blue" >

<H1 align = center > FRIENDS </H1>

<H2 align = center >MEGHANA </H2>

<H3 align = center >DIVYA </H3>

<H4 align = center >VYSHNAVI </H4>

<H5 align = center >SRAVANTHI </H5>

<H6 align = center >CHANDANA </H6>

</Body>

</HTML>

PROCEDURE TO WRITE HTML PROGRAM :

Step I :- To write a html program open a notepad . This can be done by clicking the start >All programs > Accessories > Notepad or typing the Note pad on command prompt.

Step II :- Type the HTML code in notepad . Save the file as file name .HTML.

Step III:- The output of the created html program can be visualized by typing the page along with the .html extension on the address bar of any web browser(or) Double click on the created file than it automatically opens in the browser.

FORMATING TEXT WITH HTML :

Q5. Explain formatting a text with HTML with examples.

A. A tag that you apply to an individual character or text is referred to as a character tag or formatting tags. These tags are used for changing the appearance of the text. There are two types :-

A. **Physical tags**. Physical Tags are used in HTML to provide actual physical formatting to the text. Following are the Physical tags commonly used in HTML.

B**. Logical tags**. Logical Tags are used in HTML to display the text according to the logical styles

A. Physical tags :-

1. Header tag : - Heading is used to divide sections of text. HTML defines 6 levels of headings, the number indicates heading line from 1 to 6 i.e., <H1> creates a largest text size and <H6> creates a smallest text size. This contains attribute called “ALIGN”, which set to left/center/right/justify.

Syntax :- <H1>- - - - -</H1>

<H2>- - - - -</H2>

<H3>- - - - -</H3>

<H4>- - - - -</H4>

<H5>- - - - -</H5>

<H6>- - - - -</H6>

Example :- <H1>Web programming</H1>

<H2>E commerce</H2>

<H3>Corporate accounting</H3>

<H4>cost accounting</H4>

<H5>management accounting</H5>

<H6>B- law</H6>

Example :- <H1 align="center">- - - </H1>

2. <B> (Bold) :- This tag sets the text style to bold.

Syntax :- <B> ..... </B>

Example :- <B> Hello </B>

Output :- **Hello**

This tag contains closing tag also. So this is called as paired tag.

3. (Italic) :- This tag sets the text style to Italic.

Syntax :- .....

Example :- Hello

O/P :- *Hello*

This tag contains closing tag also. So this is called as paired tag.

4. <u> (underline) :- This tag is used for giving underline to the text.

Syntax : - <u>.....</u>

Example :- <u> Hello </u>

O/P = Hello

This tag contains closing tag also. So this is called as paired tag.

5. (center) :- This tag aligns the text to the center of your webpage.

Syntax:- - - - - -

Example :- hello

6.<S> or <Strike> :- This tag strikes the text in the webpage.

Syntax : - <S> .....</S> or <strike>- - - - - </strike>

Example : <S> ABC </S> or < strike>ABC </strike>

O/P :- ~~ABC~~

This tag contains closing tag also. So this is called as paired tag.

7.<BIG> :- It displays the text in bigger font than the current default size.

Syntax :- <BIG> - - - - - </BIG>

Example :- <BIG>WELCOME</BIG>

O/P :- WELCOME

This tag contains closing tag also. So this is called as paired tag.

8.<SMALL> :- It displays the text in a smaller font than the current default size.

Syntax :- <SMALL> ……..</SMALL>

Example :- <SMALL> WELCOME </SMALL>

O/P:- WELCOME

This tag contains closing tag also. So this is called as paired tag.

9.<SUP> :- This tag styles a text as a superscript.

Syntax :- ^{……}

Example : a²

O/p :- a^2

This tag contains closing tag also. So this is called as paired tag.

10.<SUB> :- This tag styles a text as a subscript.

Syntax :- _{…..}

Example : - H ₂ O

O/P = H_2O

This tag contains closing tag also. So this is called as paired tag.

11.<EM> :- This tag emphasizes a text visually rendered in italics. This tag mainly works on Net scape navigator browser.

Syntax :- <EM> …. </EM>

Example :- <EM> Welcome </EM>

O/P :- *Welcome*

This tag contains closing tag also. So this is called as paired tag.

12. <Strong> :- This tag emphasizes text strongly , usually rendered in bold. This tag mainly works on Net scape navigator browser.

Syntax :- <Strong > ..... </Strong>

Example :- < Strong> Welcome</Strong>

O/P :- **Welcome**

This tag contains closing tag also. So this is called as paired tag.

13. <Blink> :- This tag displays enclosed text as blinking ON and OFF approximately on a second. This tag supports on Net scape navigator only.

Syntax :- <Blink> .....</Blink>

Example :- <Blink> Welcome </Blink> .

This tag contains closing tag. So, this is also called paired tag.

**14.
 (break) :-** This tag inserts a line in a Webpage. It requires only opening tag . So it is also called as unpaired tag.

Syntax :- .....

Example :- Welcome
 to
 HTML

O/P:- Welcome

To

HTML

15.<Wbr> (word break) :- This tag indicates whenever word breaks are allowed. This tag is also having only opening tag. So this is called as unpaired tag.

Syntax :- <wbr>- - - - - -

Example :-<Wbr> HELLO

O/P :- H

E

L

L

O

16. <Hr> (Horizontal rule) :- This tag is used to draw a horizontal line to separate a text. No closing tag is required for this tag. This is also called as unpaired tag. The attributes in this tag are

(i) Align :- The values are left, Right , center

(ii) Color :- It sets the color of the rule or line in the Web page.

(iii) Size :- It sets the vertical size of the horizontal rule.

(iv) Width :- It sets the width of the line.

Syntax :- <HR Align = “ ” color =” ” size = “ ” width =” “>

Example :- <HR Align = “center” color = “pink” size = “20%” width = “3”>

17. <PRE>(preformatted) :- This tag tells the browser that the enclosed text is in preformatted and should not be reformatted.

Syntax :- <PRE>- - - - - - -</PRE>

Example : <PRE>

S/W

1.‘C’ S/W

2.ORACLE

3.HTML

</PRE>

O/P: S/W

1.‘C’ S/W

2.ORACLE

3.HTML

This tag contains closing tag. So this is also called as paired tag.

18. <Font> :- This tag lets you select text, set color, style(face), size. The attributes of these tag are

i) color

ii) Size

iii) Face.

1) Color : This attribute changes the color of the selected text.

2) Size : - This attribute changes the size of the type possible values range from 1 to 7.

3) Face : - This attribute changes the style of the text.

Syntax : - < font color = “ ” size = “ ” face = “ ” >- - - - -

Example :- < font color = “pink” size = “3” face = “Arid”>Welcome to HTML

Create web page to display INTERNET should be middle.

INTRODUCTION :

Internet is a network of networks www is a complex international ISP means Internet service provider.

<HTML>

<HEAD>

<TITLE> INTERNET </TITLE>

</HEAD>

<BODY>

<BODY BGCOLOR = “blue” >

<H1> <u> INTERNET </u> </H1>

<u> INTRODUCTION </u>

<p> INTERNET : - INTERNET

is a network of networks
 <u>www </u> is a complex International network
  ISP means internet service provider.

</body>

</HTML>

19.<p> (paragraph) :- This tag is used to format text into a paragraph and add a space before the paragraph. For this tag only opening tag is required. So, this is called as unpaired tag.

Syntax :- <p>.....

Example :- <p> Welcome to HTML World

O/p :- Welcome to HTML World

This tag contains a attribute called align.

ALIGN :- it sets the alignment of text in the paragraph and the values are left, right and center.

Syntax :- <p Align = “Left/Right/Centre”>

Example :- <p Align = Right > HTML means type markup language.

20. <MARQUE> :- This tag displays a text in a marquee style . This tag gives a animation effect to the text.

Syntax : <Marquee > - - - - - - - </Marquee>

Example: <Marquee> Welcome </Marquee>

This tag contains closing tag also.

The attribute of this tag are :-

i) Align

ii) BG Color

iii) Direction

iv) Loop

i) Align :- This attribute sets the alignment of the text relative to the marquee. The values are top , middle, bottom.

ii) BG color :- It sets the background color for the marquee box.

iii) Direction : - It sets the direction to the text the values are left, right, down and up.

iv) Loop :- It sets how many times you want the marquee's cycle. The value should be positive integer.

Example : <marquee Align = "Top" BG color = "Pink" Direction = "Right" Loop = "5"> Welcome </Marquee>

Programs for font tag and Marquee tag create a web page with sports names atleast five in five different colors, different sizes, and different faces and give the animation affect the heading sports.

```
<HTML>
<HEAD>
<TITLE> SPORTS </TITLE>
</HEAD>
<BODY BG COLOR = "blue".>
<Marquee BG color = "Pink" Align ="bottom" direction =" Right" Loop = "5">
<H1> SPORTS </H1> </Marquee>
<font size = "5" color =" Red" face = "Times Roman" >Cricket </Font> <br>
 <font size = "4" color = "blue" face = "Ariel" >Tennis </font> <br>
 <font size = "3" color = "yellow" face = "courier" > foot ball  </font> <br>
 <font size ="2" color = "black" face = "courier" >volley ball </font> <br>
 <font size =" 7" Color = "Green" face = "bold" >badminton</font>
</body>
</HTML>
```

IMAGES

Q7) How we can create images in HTML?

A) The <img> tag defines an image in an HTML page.

The <img> tag has required following attributes:

1) src: Specifies the URL of an image

2) alt: Specifies an alternate text for an image

3) height: Specifies the height of an image

4) width: Specifies the width of an image

5) align : Specifies the alignment of an image. values are top / bottom / middle /left /right

6) Border: you can specify border thickness in terms of pixels using border attribute. A thickness of 0 means, no border around the picture.

Syntax : <img src=" " alt=" " height=" " width=" " align=" " border=" ">

Example : <img src="smiley.gif " alt="Smiley face" height="42" width="42" align="middle" border="5">

Sample program :

```
<html>
<head>
<title> welcome </title>
</head>
<body>
<h1 align = "center" > sample program for image</h1>
<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

</body>
</html>
```

Anchor tag

Q8) What is anchor tag? Explain with example.

Or

What is hyperlink? Explain how we can create hyperlink in HTML

A) HTML links are hyperlinks. To create hyperlinks in HTML use tag called <a> (anchor tag). A link does not have to be text. It can be an image or any other HTML element.

Syntax : *link text*

Where **href** attribute specifies the destination address of the link.

Clicking on the link text will send you to the specified address.

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

HTML Links - Image as Link

It is common to use images as links:

Example

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" >
</a>
```

Sample program for anchor<a> tag:

```
<html>
<body>
<a href="sample . html">HTML Images</a>
<a  href= "sample1.html" >
  < img    src="image.gif"  alt="HTML tutorial" >
</a>
</body>
</html>
```

LISTS

Q9) Explain types of lists in html?

A) Lists are used to display information in a compact, and a serial order or unordered manner.

There are five different types of lists.

1. Unordered lists.
2. Ordered lists.
3. Menu lists.
4. Directory lists.
5. Definition lists.

1)**Unordered lists** :- This unordered list is also called as bullet lists. i.e., unordered lists is a list of items, each one preceded by a bullet (a distinctive character , typically a small black circle).

The lists begins and ends with a tags <UL>and </UL>. <LI>, which stands for lists item.<LI>has a closing tags,</LI> but not compulsory.

Syntax :- <UL>

<LI>items1

<LI>items2

.

.

</UL>

Example :-<UL>

<L1> orange

<L1>grapes

<L1>mango

</UL>

O/P:- .orange

.grapes

.mango

In this unordered lists type, we have attribute called type. We can customize this lists type by this attribute. We have three different values, that we can use i.e., square, circle and disk.

Syntax:- <UL type= “value”>

<LI>items1

<LI>items2

</UL>

Example:-<UL type =”square”>

<LI>orange

<LI>grapes

<LI>mango

</UL>

O/P:-

- orange
- grapes
- mango

2) **Ordered lists**:- The ordered list look like unordered but the only difference is that instead of <UL> and </UL>,and ordered lists is contained within the tags called <OL> and</OL>, when an ordered lists is displayed in the web browser . If uses an automatically generated sequence of item markers i.e., items are numbered by default it is numbered as 1,2,3,---------- etc

Syntax:- <OL>

<LI>items1

<LI>items2

</OL>

Example:-<OL>

<LI>orange

<LI>grapes

<LI>mango

</OL>

O/P:- 1.orange
2.grapes
3.mango

The ordered lists are customized using an attribute called type as in unordered lists. The set of values can be assigned to type are :-

(i) 1,2,3,- - - -default numbering scheme
(ii) A- upper case letters (A B C D- - - - -)
(iii) a - lower case letters (a b c d - - - - - -)
(iv) I - large roman numerals (I,II,III,IV- - - - -)
(v) i - small roman numerals(i, ii ,iii , iv- - - -)

Syntax:- <OL type="I">
<LI>item1
<LI>item2
</OL>

Example:-<OL type ="i" >
<LI>orange
<LI>grapes
<LI>mango
</OL>

O/P:- i. orange
ii. grapes
iii. mango

3) **Menu list <Menu>:-** It create a menu list usually display , simply as an unordered list.

Syntax:- <menu>
<LI>items1
<LI>items2
</menu>

Example:-<menu>

<LI> pink

<LI>red

<LI>black

</menu>

O/P:- pink

red

black

4) **Directory list:-** It creates a directory list usually displayed as an unordered list . This is used to design uses interface pages, list of film names or links to other pages.

Syntax:- <Dir>

<LI>items1

<LI>items2

</Dir>

Example:-<Dir>

<LI> pink

<LI>red

<LI>black

</Dir>

O/P:- pink

red

black

5) **Definition list:-** Definition list begins and end with the tags <DL> </DL> unlike. The unordered an ordered list, definition list and are not based on list items. They are instead based on term definition pairs. A good way to think of it is that

<DL> - Definition list.

<DT> - Definition list term.

<DD> - Definition term definition.

Syntax:- <DL>

<DT>term1

<DD>term1 definition

<DT>term2

<DD>term2 definition

</DL>

Example:-<DL>

<DT> web

<DD>collection of networks

<DT>html

<DD>hyper text markup language

<DT>internet

<DD>collection of webs

</DL>

O/P:- web

collection of networks

html

hyper text markup language

internet

collection of webs

- **Create a web page using ordered list for sports atleast 10.**

```
<html>
<head>
 <title>sports</title>
 </head>
<body text = “blue”>
<h1> <center> <u> <i> sports </i> </u></center> </h1>
<OL type = ”A”>
<LI>cricket
<LI>foot ball
<LI>volley ball
<LI>basket ball
```

<LI>golf
<LI>tennis
<LI>badminton
<LI>rugby
<LI>hockey
<LI>kabaddi
</body>
</html>

- **Create a web page using unordered list for sports atleast 10.**

<html>
<head>
<title>sports</title>
</head>
<body text = “lavender”>
<h1> <u> sports </u> </h1>
<UL type =”square”>
<LI>cricket
<LI>foot ball
<LI>volley ball
<LI>basket ball
<LI>golf
<LI>tennis
<LI>badminton
<LI>rugby
<LI>hockey
<LI>kabaddi
</body>
</html>

TABLES

Q10.) Write about tables in html or Explain about elements of table in html.

A) A representing data in the form of tables one of those traditional approaches of data management. Here data is usually organized in terms of rows and columns.

A table can be set to have the following

(i) A title

(ii) Heading of each column that explain what the cells, rows or columns in the table contain

(iii)Rows

(iv)Cells

Table elements and tags are

(i) the tag <table> and </table> defines a table in html. Enter code corresponding to given table should be inscribed within these tags.

(ii) <capition> </capition> defines a caption or the title of the table the default position of the title is center at the top the table

(iii) <TH> </TH> defines a table header cells usually or on the first row and contain information regarding the rest of the table. By default the text in this cells is bold and centred.

(iv) <TR>--</TR> opecifies a table rows within a table. Cells within each rows are defined with in these tags.

(v) <TD> ---</TD> it defines a table data cell by default the text in this cell is aligned left and centered vertically.

Some simple rules. For formatting tables.

1. The <table> </table> must surrounds the entire table definition.

2. The first item inside the table is the <caption>which is optional.

3. This can be any no of rows defined by the <TR> and </TR>

4. Within a row you can have any number of cells defines by the <TD> or <TH> and </TH>

Syntax :-

```
<table>
    <tr>
        <caption> xyz </caption>
    </tr>

    <TR>
        <TH> heading</TH>
        <TH> heading</TH>
        ……
    </TR>
    <TR>
        <TD> data1</TD>
        <TD>data2</TD>
    </TR>
    <TR>
        <TD> data1</TD>
        <TD>data2</TD>
    </TR>
</Table>
```

Student Information	
studno	**studname**
1	Sita
2	geeta

Example :

```
<Table>
<tr>
```

```
<caption> Student Information</caption>
</tr>
<TR>
    <TH> studno</TH>
    <TH> studtname</TH>
</TR>
<TR>
    <TD> 1</TD>
    <TD>Sita</TD>
</TR>
<TR>
    <TD> 2</TD>
    <TD>gita</TD>
</TR>
</Table>
```

TABLE ATTRIBUTES

(i) **Align:-** using this attributes direct a place a given table in the Browsers, by assigns values left right or center

(ii) **Background:** - to set a image or other graphics as a background to the table.

(iii) **BG color:-** using this attributes we usually set the color of the window where the table is currently displayed. This attribute uses in <TR>,<TH>,<TD>

(iv) **Colors:** - we can specify the number of columns in a given table.

(v) **Height:-** it specifies height of a given table

(vi) **Width:-** it specifies width of a given table

(vii) **Border= n:** - It is used to put a border around the table.

(viii) **V Align**(top,middle,bottom):- vertical alignment of a cell appearing inside a <TR> <TH> and <TD>

(ix) **Cell spacing = n**: - This attribute is used for giving a distance between the cells in specified pixels.

(x) Rowspan: The rowspan attribute in HTML specifies the number of rows a cell should span. It can be used with <td> and <th> element in an HTML Table. It provides the same functionality as "merge cell" in the spreadsheet program like Excel.

<td rowspan = "value">table content...</td>

<th rowspan = "value">table content...</th>

xi) Colspan: The colspan attribute in HTML specifies the number of columns a cell should span. It provides the same functionality as "merge cell" in the spreadsheet program like Excel. It can be used with <td> and <th> element while creating an HTML Table.

<td colspan = "value">table content...</td>

<th colspan = "value">table content...</th>

Write a program with the table

STUDENT DETAILS			
Stno	**Stname**	**class**	**Percentage**
1	A	BCOM	56
2	B	BCOM	60
3	C	BCOM	70
4	D	BCOM	80
5	E	BCOM	90

```
<html>
<head>
<title>INFORMATION</title>
</head>
<body>
<table>
<caption>STUDENT DETAILS</caption>
<TR>
<TH> Stno</TH>
<TH>Stname</TH>
<TH>class</TH>
<TH>Percentage</TH>
```

```
</TR>
<TR>
<TD> 1</TD>
<TD>A</TD>
<TD>B.com</TD>
<TD>56%</TD>
</TR>
<TR>
<TD> 2</TD>
<TD>B</TD>
<TD>B.com</TD>
<TD>60%</TD>
</TR>
<TR>
<TD> 3</TD>
<TD>C</TD>
<TD>B.com</TD>
<TD>70%</TD>
</TR>
<TR>
<TD> 4</TD>
<TD>D</TD>
<TD>B.com</TD>
<TD>80%</TD>
</TR>
<TR>
<TD> 5</TD>
<TD>E</TD>
<TD>B.com</TD>
```

```
<TD>99%</TD>
</TR>
</Table>
</body>
</html>
```

WEB DESIGING PRINCIPLE

Q11) EXPLAIN WEB DESIGNING PRINCIPLES.

A) Designing a web presentation like designing a book outline a building plan or a painting can sometimes be a complex and involved process A simple rule of thumb is to have each topic represented by single page. The primary forms of navigation between pages are links for forward, up, down, back or home all fall under this category of primary navigation. In addition navigation to the simple navigation links some web presentations contain extra information i.e. parallel to the main web content

Desigining and building a website is both an art and a science. Designs should enable users to reach their goals with speed , effectiveness .

Top web design principles that will make your website visually pleasing , easy to use, engaging and effective.

1. Appearance
2. Content
3. Purpose of each element
4. Design consistency
5. Effective Navigation
6. Usability
7. Functionality
8. Grid-based layout

1. **Appearance:** A website reflects company's product and service. Hence , it must be visually appealing ,polished, and professional. All the visual elements of any website- LOGO, fonts, template, layout and colors should meet company's needs and features. An attractive site generates a positive impression which helps to attract users attention. A good use of color readable text, meaningful graphics, quality photography and simplicity impress site visitors.

2. **Content**: Along with appearance,website must have good content. The key to a good website is providing relevant content that readers want. The relevant website content closely matches the information people need. The informative website content closely matches the information people. The informative website increase visitor confidence in company's knowledge and competence.

3. **Purpose of each element:** In a good design,each element should be weight and measured before being placed in the layout. Website's page is limited by a screen size and scrolling. So you can't afford to waste the precious space for unnecessary and useless decorations.

Each website page should also have a clear purpose and help visitors to find what they need. Each button, navigation menu,contact form etc. should lead your visitors exactly to their goals without confusing them.

4. **Design Consistency**: Design elements should match throughout each page. It means headings, sub headings, fonts, button styles, and sizes- everything should be the same look and feel every time they are used. Think out each detail in advance. Choose the right colors and fonts for texts and buttons.

5. **Effective Navigation:** Good navigation is the basis for effective web design. A main navigational toolbar or menu should come up with a meaningful and clear way to organize, arrange, and display content to users. The visitor looking for content on website should not be time consuming. The navigation design must have consistency, user expectations and contextual clues.

6. **Usability:** the success of website can be measure by its degree of usability. Website must be easy to read, navigate and understand. The various elements that can improve the usability of any website includes browser compatibility , user friendly, descriptive text linking, effective navigation, fast loading pages, proper scroll, screen resolution tec.

7. **Functionality:** website has to be functional and work correctly. The various elements that can improve the functionality of any website include proper hyperlinks, contact forms, site search options, event registration, multimedia elements, sitemap, and image positions.

8. **Grid – Based layout:** the randomly placed images, pieces of text , buttons and forms all over the page leave users frustrated. Proper alignment of the website content of the page it in a helps to avoid the mess and organizing it in a more user-friendly way. It helps to get the information without efforts.

FORMS

Q12) Explain Form elements?

A) An html form is a selection of a document containing normal content , markup, special elements, called controls (checkboxes, radio buttons , menus, etc) and labels on those controls. User interacts with forms through named controls .A control's "control name" is given by its name attribute.

Forms provide a means for the user to add graphical user interface (GUI) components to the web pages. Forms are generally used to accepts the data and pass it on to programs for processing.

forms should be included in between the <body> and </body> tags with in

<FORM>.....</FORM>

tags like most html tags .Then , the user can start adding form elements like the radio buttons, checkboxes ,or text input fields.

THE HTML FOR A BASIC FORM.

<HTML>

<HEAD>

<TITLE>.....</TITLE>

</HEAD>

<BODY>

<FORM>

FORM ELEMENTS

</FORM>

</BODY>

</HTML>.

Main attribute for the form is METHOD.

Values for method are GET, POST.

Method= GET

When we use GET , all information from a form is appended onto end of the URL being requested .

Method= POST

This method transmits all form input information immediately after the requested URL.

HTML Form Controls/ Elements

There are different types of form controls that you can use to collect data using HTML form –

- Text Input Controls
  - Text
  - Password
  - Text area
- Buttons
  - Checkboxes Controls
  - Radio Box Controls
  - Select Box Controls
  - Submit
  - Reset

The elements within the form are indicated by the <INPUT> tag .this tag is having two important attributes Type and name

Syntax : <INPUT NAME = ” text ” type = “ text ”>

Name: indicates the name of the element .it defines the name of the data for the field.

Type: defines the various types of input fields recognized in the current html specification. 1) Text 2) radio 3) checkboxes 4) password, 5) submit.

TEXT INPUT ELEMENTS

1)TEXT FIELD: A text input box is used to enter and edit a single piece of text by the user.

Attributes are:

Size: specifies the length of the text box.

Value: specifies the value in the text box.

Max length: specifies the maximum length of text that can be entered into the text box.

Example : <INPUT TYPE =”TEXT” NAME = ”FILENAME” SIZE=10 VALUE =”XYZ” MAXLENGTH = 50>.

2) PASSWORD TEXT FIELD: this works the same way as the text field except the characters typed are echoed back to the browser in a masked format .(asterisks).

Syntax:<INPUT TYPE =”PASSWORD” NAME=”PASSWORD”>

ATTRIBUTES ARE : size , value , max length .

3) TEXT AREA FIELD: text area fields can contain many lines of text giving the use opportunity to type in larger inputs unlike the regular text input fields.

Syntax : <text area name=”xyz” rows=”n” cols=”n”> …. </Text area>

Attributes are name, rows, cols.

Name - name of the text area.

Rows - no of lines in the text area. (height).

Cols-no of characters in one row.(width).

BUTTONS

Check boxes: check boxes can be used to choose multiple items in a list . each checkbox can be either be on or off i.e it can be selected or not selected. When the user has to choose more options at a time checkbox is good element to use unlike radio buttons , where only one item from each group can be selected .

Ex:<INPUT TYPE =” CHECKBOX” NAME = “SUNDAY”> SUNDAY

<INPUT TYPE = “CHECKBOX” NAME= “MONDAY”> MONDAY

<INPUT TYPE =”CHECKBOX ” NAME=”TUESDAY”> TUESDAY

<INPUT TYPE =”CHECKBOX” NAME =”WEDNESDAY”>WEDNESDAY

Attributes are type ,name ,checked.

Checked: the boxes can be checked by the default by using the “CHECKED” attribute.

Radio buttons: these are used for a list of items of which only one can be choosen . when the user chooses one option i.e if one radio button is selected any previously selected option is unselected radio buttons are grouped together by giving the same name field for each one.

Ex:< INPUT TYPE =”RADIO” NAME =”WEEKDAY”>SUNDAY

<INPUT TYPE =”RADIO” NAME =”WEEKDAY”> MONDAY

SUBMIT BUTTON : after the user fills out the form , the “submit” button is clicked to submit the data to the CGI script.

Ex: <INPUT TYPE = “submit” value = “submit”>

This code creates a submit button with a lable “submit”.

When the user clicks the submit button , the browser collects the values of each of the input fields and sends them to the web server .

RESET BUTTON : The reset button is similar to the submit button and can be used along with it at the end of the form . It clears all the selection made by the user in the form and resets them to the default values.

Ex : <INPUT TYPE = “RESET” VALUE= “RESET TO DEFAULT”>.

SELECTION LISTS

SELECTION LISTS : selection list is another useful element that enables the user to selected from a list of items. Each item can be selected or not selected. This is similar in a scope to the checkboxes with the differences that this occupies lesser space.

Selection lists can be of three types :

1. Drop –down list boxes.

2. Simple list boxes.

3. Scrollable list boxes.

The type of selected list displayed is determined by the SIZE keyword in the SELECT tag .

Size refers to the no of rows visible to the user at a time.

If the SIZE = 1 , a drop –down list box is created.

If the size > 1 , and the no of options is less than or equal to the size , a simple list box is displayed.

If the size > 1, but the no of options available to the user is greater than the size, a scroll bar will appear to the right of the list box to allow the user to scroll through the options .

DROP-DOWN LIST BOX

<SELECT NAME = ”WEEKDAY”>
<OPTION> SUNDAY
<OPTION>MONDAY
<OPTION>TUESDAY </SELECT>
SIMPLE LIST BOX
<SELECT NAME=”WEEKDAY” SIZE=3>
<OPTION> SUNDAY
<OPTION> MONDAY
<OPTION> TUESDAY
</SELECT>
SCROLLABLE LIST BOX
<SELECT NAME=”WEEKDAY” SIZE=3>

<OPTION> SUNDAY
<OPTION> TUESDAY
<OPTION> WEDNESDAY
<OPTION> THURSDAY
<OPTION> FRIDAY
<OPTION> SATURDAY
</SELECT>

FRAMES

13) EXPLAIN FRAMES IN HTML.

OR

1) EXPLAIN ABOUT ELEMENTS OF FRAMES IN HTML.

A) Diving a single browser window into multiple regions for separate document viewing is called frame.

Frames allow the user to divide a web page into independent sections that act as sub-windows. As each frame acts as a separate web page.

Life an independent web page a frame can have a background attribute , scroll bars and name.

Frames also allow more control over the size, look and feel of a page layout

Hyperlinks in one frame can be used to update the contents of other frames making it possible to build user friendly and visually pleasing web pages.

<Frame set>

Syntax:-

Frames are defined with in <frameset> tags.

These tags are defined in a frameset document.<body> tags are replaced with <frameset>tags.

Frameset tags tell the browser how many row frames or column frames it should divide the browser window into.

Syntax:-

1. <frameset cols =”size1%,size2%,.....size n”> for column frame declarations.

2. <frameset rows =”size1% ,size%,.....size n”> for row frame declarations.

Example : -

9. <frameset cols = “50%,50%”>

10. <frameset rows = “50%,50%”>

Where n is the no of rows or columns and should be greater than one.

<FRAME>:- After declaring a frame with the <FRAMESET>tag, It is also necessary to define it using the **<FRAME> tag** .In <frame> tag attribute used is “SRC” which specifies the URL. URL is the address of the document to be displayed.

When a user loads a frameset document , this browser window will first divided in to frames , and then the documents specified in the <frame> tags will be loaded and displayed inside the frames.

Example:-

Divide the browser into two halves, ie into two columns 1st column contains a .html file and 2nd column contain b.html.

<FRAMESET ROWS =”50%,50%”>

<FRAMESET SRC=”a.html”>

<FRAME SRC=”b.html”>

</FRAMESET>

Example :

Divide the browser into two halves, ie into two rows 1st row contains **a .html** file and 2nd row contains **b.html**.

ATTRIBUTES OF FRAMES ARE:-

1. NAME: a frame can be given a name which is generally alphanumeric.
2. SCROLLING: scrolling attribute can take 3 values:-
   1. scrolling=”yes”
   2. scrolling=”no”
   3. scrolling=”auto”

This the browser decide it scrollable are required which may provide increased frame area for data.

NO FRAMES TAG:-

Not all browsers are “frame-capable” and netscape has provided a <noframes> tag set to follow the <FRAMESET> command . the code for the “normal”page is bounded by the <NOFRAMES> and </NOFRAMES> Is bounded by the <NOFRAMES> and </NOFRAMES> tag set and includes the <body> and</body> tag set and there in

```
<FRAMESET ROWS=”50%,50%”>

<FRAME SRC=”TOP.HTML”>

<FRAME SRC=”BOTTOM.HTML”>

</FRAMESET>

<NOFRAMES>

<BODY>

.
.
.</BODY>
</NOFRAMES>
```

NESTED FRAMES :-

Ussually a web document is given a better look when it is divided into more than two frames. A standard layout is a title frame and footer frame with the body divided into two frames. To crate this nested frame declarations are necessary. A frame is set to be a nested frame if it was formed by dividing another frame, called the parent frame, into multiple frames.

```
<frameset rows = “15%,70%”>
<frame src = “top.html”>
<frameset cols = “50%,50%” >
<frame src= “left.html”>
<frame src = “right.html”>
</frameset>
</frameset>
```

<Frameset> attributes:-

- **Border :** it sets width of the frames border in pixels.

- **Frame border :** it sets border to the frame.

EXAMPLE:-
trail.html

```
<HTML>
<HEAD>
<TITLE> FRAME </TITLE>
</HEAD>
<FRAMESET COLS=”20%,  80 %”>
<FRAME SRC=”MENU.HTML” NAME=”MENU” SCROLLING=”YES”>
<FRAME SRC =”DETAILS.HTML” NAME=DETAILS SCROLLING=”YES”>
</FRAMESET>
</HTML>
```

MENU.HTML

```
<HTML>
<HEAD>
<TITLE> FRAME </TITLE>
<BODY> WELCOME TO FRAME1</BODY>
</HTML>
```

DETAILS.HTML

```
<HTML>
<HEAD><TITLE>FRAME</TITLE>
</HEAD>
<BODY>
WELCOME TO FRAME 2
</BODY>
</HTML>
```

1)Example of an HTML Form :

```
<html>
<body>

<form>
 Username:<br>
 <input type="text" name="username">
 <br>
 Email id:<br>
 <input type="text" name="email_id">
 <br><br>
 <input type="submit" value="Submit">
</form>
</body>
</html>
```

2) Example for menus

```
<html>
<h3>Example of a Select Box</h3>
<body>
   <form>
      <label for="country">Country:</label>
      <select name="country" id="country">
         <option value="India">India</option>
         <option value="Sri Lanka">Sri Lanka</option>
         <option value="Australia">Australia</option>
      </select>
   </form>
</body>
</html>
```

DYNAMIC HTML

Q1) What is DHTML? Explain its Features?

Dynamic HTML is a combination of technologies to make web-pages dynamic & used to create interactive and animated web sites by using a combination of HTML 4.0,a client side scripting language,and a presentation definition language such as CSS,and DOM (Document object model).

DHTML makes the web experience faster and more interactive for end users.

Components of DHTML:-

DHTML requires following components:

1.**HTML** – It stands for Hyper Text Markup Language. It is used for creating different page elements like text, tables ,forms.

2.**CSS(Cascading Style Sheet)-**cascading style sheets allows the authors and users to attach style to structured documents by saperating content of documents and the presentation style of documents.css simplifies site maintanance and web authoring.

3**. VBSCRIPT/JAVASCRIPT**: A scripting language is a programming language designed for integrating and communicating with other programming languages. Some of the most widely used scripting languages are JavaScript, VBScript, PHP, Perl, Python, Ruby, ASP and Tcl. Since a scripting language is normally used in conjunction with another programming language, they are often found alongside HTML, Java or C++.

4**. DOM** : With the object model, JavaScript gets all the power it needs to create dynamic HTML: The DOM is a W3C (World Wide Web Consortium) standard.The DOM defines a standard for accessing documents:

Features of DHTML:-

DHTML allows the used to create dynamic web pages. A dynamic web page is a web page where the structure, style or content of the page can be changed after the page is loaded in the browser.

(1) Document Object Model (DOM):-

DHTML provides a comprehensive object model for HTML. This model exposes all page elements as objects. These objects can easily be manipulated by changing their attributes or applying methods to them at any time. DHTML also provides full supports for keyboard & mouse events, on all page elements.

(2) Dynamic content:-

Text on graphics can be added, deleted modified on the fly. For ex: A web page can display an updated headline, without refreshing the page. The text surrounding the headline will reflow automatically.

(3) Dynamic styles & cascading style sheets (CSS):-

Any CSS attribute, including color & font can be updated without a server round-trip. For example text can change color or size when a mouse pointer passes over it. Multimedia filters & transition effects can be applied to HTML elements simply by adding the filter CSS attribute.

(4) Absolute positioning:-

With DHTML, CSS positioning coordinates for existing page content can be updated at any time to create animated effects, without reloading the page.

(5) Data Binding:-

DHTML provides data- driven application front ends which can be built to present, manipulate (eg. Sort, filter) and update data- on the client without numerous round-trips to the server.

(6) Script-lets:-

A script-let is a web page, authored with Dynamical HTML, which content providers can use as a component in their web applications. With Scrip let's, content providers can author content once, then easily re use the content in other web pages or applications.

Q 2) What is cascading style sheet(CSS)?

- CSS is used to control the style of a web document in a simple and easy way.
- CSS is the acronym for "Cascading Style Sheet".
- CSS was invented by **Håkon Wium Lie** on October 10, 1994 and maintained through a group of
- people within the W3C called the CSS Working Group. The CSS Working Group creates documents called **specifications**.
- CSS is an excellent addition to plain HTML.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

With plain HTML you define the colors and sizes of text and tables throughout your Pages. If you want to change a certain element you will therefore have to work your way through the document and change it.

With CSS you define the colors and sizes in "styles". Then as you write your documents you refer to the styles. Therefore: if you change a certain style it will change the look of your entire site.

Another big advantage is that CSS offers much more detailed attributes than plain HTML for defining the look and feel of your site.

Advantages of CSS

- **CSS saves time** – You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

- **Multiple Device Compatibility** – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
- **Offline Browsing** – CSS can store web applications locally with the help of an offline catche.Using of this, we can view offline websites.The cache also ensures faster loading and better overall performance of the website.
- **Platform Independence** – The Script offer consistent platform independence and can support latest browsers as well.

CSS stands for Cascading Style Sheets. It is a way to divide the content from the layout on web pages.

A style is a definition of fonts, colors, etc. Each style has a unique name: **a selector.**

The selectors and their styles are defined in one place. In your HTML contents you simply refer to the selectors whenever you want to activate a certain style.

For example:

Instead of defining fonts and colors each time you start a new table cell, you can define a style and then, simply refer to that style in your table cells.

Compare the following examples of a simple table:

Classic HTML

```
<table>

<tr><td bgcolor="#FFCC00" align="left"><font face="arial" size="2" color="red"><b>this is line 1</b></font></td></tr>

<tr><td bgcolor="#FFCC00" align="left"><font face="arial" size="2" color="red"><b>this is line 2</b></font></td></tr>

<tr><td bgcolor="#FFCC00" align="left"><font face="arial" size="2" color="red"><b>this is line 3</b></font></td></tr>

</table>
```

With CSS (assuming that a selector called subtext is defined)

```
<table>
<tr><td class="subtext">this is line 1</td></tr>
<tr><td class="subtext">this is line 2</td></tr>
<tr><td class="subtext">this is line 3</td></tr>
</table>
```

While CSS lets you separate the layout from the content, it also lets you define the layout much more powerfully than you could with classic HTML.

2)Explain how we can include CSS in HTML.

A) <html>

```
<head>
<title>HTML CSS</title>
</head>

<body>
 <p style = "color:green; font-size:24px;" >Hello, World!</p>
</body>

</html>
```

Q3)Define a STYLE RULE OR STYLE? OR What is a selector?

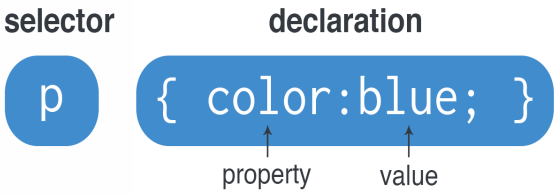
A style sheet is made up of style rules that deal a browser how to present documents. There are various way of linking these style rules to your HTML documents, but the simplest method for starting out is to use HTML'S style element. It is placed in Head tag.

Style Rule:-

A style rule is combination of property and value.

Syntax: selector {property: value}

A CSS rule-set consists of a selector and a declaration block:



Selector: it is an HTML element such as body, p, H1 etc.

Property: it is assigned to a selector in order to manipulate its style.

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:

Example:

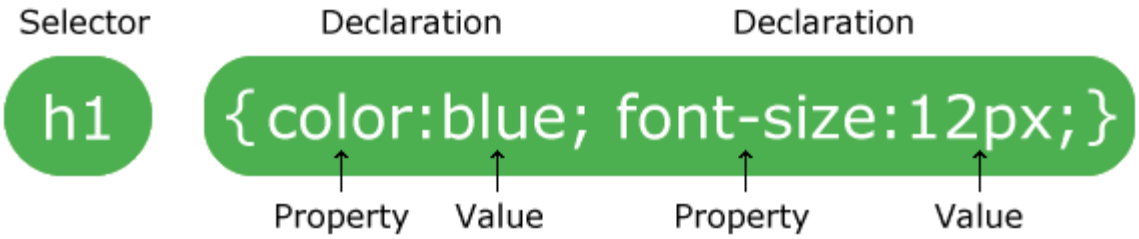
```
p {
   color: red;
   text-align: center;
    }
```

For example properties include color, margin & font etc.

Values: it is an assignment that a property receives for example the property color could receive the value red

- **Multiple style declarations for a single selector may be separated by semicolon.**

Syntax: selector {property1: value1; property2: value2}



Ex:

```
 <head>
<title> css example </title>
<style type= “text/css”>
H1 {font-size: x-large; color: red}
H2 {color: blue}
</style> </head>
```

Sample program for CSS

```
<html>
<head>
<style type = “text/css”>
h1 {color:red;}
p {color:blue;}
</style>
</head>
<body>
<h1>A heading</h1>
<p>A paragraph.</p>
</body>
</html>
```

Q 4) What are different types of selectors?(IMP)

A) **Selector: it is an HTML element such as body, p, H1 etc.**

Different types of selectors are

1) HTML selector
2) CLASS selector
3) ID selector
4) Group selector

I) HTML selector:-

Any HTML element is a possible CSS1 selector. The selector is simply the element that is linked to a particular style.

The general **syntax** for an HTML selector is:

HTMLSelector {Property1:Value;property2:value;property3:value....}

Ex: B {font-family:arial; font-size:14px; color:red}

where B is bold tag in html and is called selector in CSS.

Program using HTML selector.

```
HTML>
<HEAD>
<style type="text/css">
B {font-family:arial; font-size:14px; color:red}
</style>

</HEAD>

<BODY>
<b>This is a customized headline style bold</b>
</BODY>

</HTML>
```

II) Class selector:- A simple selector can have different classes thus allowing the same element to have different styles. Style sheets support classes or set of style changes for a document. & class can be defined to change the style in a specific way for any element it is applied to. The styles can be applied direct to each HTML element or applied to part of a document with <SPAN> </SPAN> tags.

The general syntax for a Class selector is:

.Class Selector {Property: Value ;}

A class selector is a name preceded by a full stop (**.**).

Class selectors are used when you want to define a style that does not redefine an HTML tag entirely. When referring to a Class selector you simply add the **class** attribute to an HTML tag .

Ex: .intro {color: red}

 computer //HTML

Programs using class selectors.

```
<HTML>
<HEAD>
<style type="text/css">
.headline {font-family: Arial; font-size:14px; color: red}
</style>
</HEAD>

<BODY>
<b class="headline">This is a bold tag carrying the headline class</b>
<br>
<i class="headline">This is an italics tag carrying the headline class</i>
</BODY>

</HTML>
```

Program 2:

```
<html>

  <head>
    <title>HTML Internal CSS</title>

    <style type = "text/css">
      .red
     {
        color: red;
     }
     .thick
     {
        font-size:20px;
     }
     .green
     {
        color:green;
     }
    </style>
  </head>

  <body>
    <p class = "red">This is red</p>
    <p class = "thick">This is thick</p>
    <p class = "green">This is green</p>
    <p class = "thick green">This is thick and green</p>
  </body>

</html>
```

When referring to a Class selector you simply add the class to an HTML tag like in the above example (class="headline").

III) ID selectors:-

ID selectors are individually assigned for the purpose of defining on a per- element basis. In ID selector is assigned by using the indicator "#" to precede a name. These are individually identified (named) selector to which a specific style is declared.

The general syntax for an ID selector is:

#IDSelector {Property:Value;}

ID selectors are used when you want to define a style relating to an object with a unique ID.
Ex: #svp940 {text-indent: 3em}

This would be referenced in HTML by the ID attribute as

 text indented 3em

program using ID Selector & div

```
<HTML>
<HEAD>
<style type="text/css">
#layer1 {position:absolute; left:100;top:100; z-Index:0}
#layer2 {position:absolute; left:140;top:140; z-Index:1}
</style>
</HEAD>

<BODY>
<div ID="layer1">
<table border="1" bgcolor="#FFCC00"><tr><td>THIS IS LAYER 1<br>POSITIONED AT
100,100</td></tr></table>
</div>

<div ID="layer2">
<table border="1" bgcolor="#00CCFF"><tr><td>THIS IS LAYER 2<br>POSITIONED AT
140,140</td></tr></table>
</div>
</BODY>
</HTML>
```

(V) Group selector:
we can apply a style to many selectors at a time. just seperate the selectors with a comma.

Syntax : **tag1,tag2,tag3{property :value}**

ex:- h1,h2,h3,h4,h5,h6{color ; blue}

Q 5) What is a style sheet? What are the types of CSS style sheets?

a) A Style Sheet is a collection of style rules that that tells a browser how the various styles are to be applied to the HTML tags to present the document. Rules can be applied to all the basic HTML elements, for example the tag, or you can define you own variation and apply them where you wish to. Cascading style sheets allows separating the web sites design from its content. All the design element (colors, fonts, backgrounds, page layout, etc) are stored in a cascading style sheet file with the extension of '.CSS'. All web pages link to this file to get their design information. If you need to

change the design you only need to edit one file to change the design elements on your entire website. All style rules are enclosed in <style> ……..</style>

```
Ex: <style type="text/css">
        h1 {color:blue;}
        h2 {color:red;}
        p {color:green;}
    </style>
```

Types of style sheets:-

There are different types of style sheets. Some they are as follows.\

i. Inline style sheet

ii. Embedded style sheet

iii. External style sheet

i. Inline style sheets:-

Inline style sheets are designed by using the **style attribute** to insert a style rule directly into an HTML tag.

<h1 style =”color: red”> </h1>

Inline styles can be very useful for overriding a style for only one specific instance.

For example, suppose you have a linked style sheet that defines all your headings as green, but you went a red heading in only one specific case. You can use an inline style to override the green color for only that one instance.

Multiple inline styles can be added to the same tag by enclosing the entire style rule in quotes & separating each property value pair with a semi color.

<h1 style = “color: blue; text-align: center”></h1>

Program for inline style sheets

```
<html>

  <head>
    <title>HTML Inline CSS</title>
  </head>

  <body>
    <p style = "color:red;">This is red</p>
    <p style = "font-size:20px;">This is thick</p>
    <p style = "color:green;">This is green</p>
    <p style = "color:green;font-size:20px;">This is thick and green</p>
  </body>

</html>
```

ii) Embedded style sheets:-

Embedded style sheets are placed directly with in the head portion of an HTML file. The embedded styles apply only to the specific HTML file where they are placed. Embedded style sheets are suited for documents with unique design requirements. If the styles need to be applied across muliple documents, you should link to an external style sheet instead of using an embedded style sheet.

Syntax : <style type = "text/css"> ..... </style>

<Head>

<Title> embedded style sheet </title>

<style type = "text/css">

h1 {font-family: Helvetica, sans, serif }

p{color: navy}

</style> </head>

Place the style sheet rules within comment tags, this will prevent older browsers from displaying your style sheet rules on the page.

The embedded style sheets begins with

<style type ="text/css"> and ends with </style>. The embedded style will override the linked style sheet.

Ex: <head>
<title> embed</title>
<style>
P{color: black}
</style>
</head>

Program for internal or embedded style sheet

```
<html>
<head>
  <title>Internal CSS</title>
  <style>
    body{
    background-color:#9F6;
    }
    h1{
    color:#C39;
    text-align:left;
    text-transform:capitalize;
    text-decoration:underline;
    }
    P{
    font-size:20px;
    font-family:Verdana, Geneva, sans-serif;
    background-color:#FFF;
    color:#963;
    }
    h2{
    color:#F03;
    margin-left:10px;
    }
```

```
  </style>
</head>
<body>
  <h1>Example for Internal CSS</h1>
  <p>Cascading Style Sheet is a style language that defines layout of HTML documents.CSS properties such as background, border, font, float, display, margin, opacity, padding, text-align, vertical-align, position, color etc.</p>
  <h2>Image Affected with styles</h2>

</body>
</html>
```

iii. External style sheets:-

Using external style sheets concept by creating a single **.css** file, you can link multiple HTML documents to it with the help of **LINK** tag. If you can develop the large websites with hundreds of HTML files. With an external style sheets you can create a template that provides consistent formatting for your entire company website. You can go back & change the styles of multiple pages simply by modifying the external style sheet or **.css** file.

Ex: <LINK REL = "stylesheet" type ="text /css" HREF = "style.css">

- LINK tag must be placed within head tag
- Type attribute is used to specify the type of style sheet used in document.
- HREF declares the URL or file name of the external style sheet we are linking.

.CSS file:- It contains text & page formatting rules that provide style for each HTML elements.

Ex:
```
h1 {color: red}
P {margin-left: 2em}
```

1.Type css code into a plain text file, and save with .css extension(i.e external style sheet)
2.Add the **link** tag with the attributes **rel, href, type** between the <head> and </head> tags of all HTML documents that we want to reference the external style sheet .

Ex: <link rel=stylesheet href=.....css type = "text/css>

Program for external style sheet:

```
<html>
  <head>
    <title>HTML External CSS</title>
    <link rel = "stylesheet" type = "text/css" href = "/html/style.css">
  </head>

  <body>
    <p class = "red">This is red</p>
    <p class = "thick">This is thick</p>
    <p class = "green">This is green</p>
    <p class = "thick green">This is thick and green</p>
  </body>

</html>
```

Q6)Difference between HTML & DHTML?

HTML	DHTML
I. It stands for Hypertext markup language. II. HTML is used to design the static web pages. III. HTML is not a procedural programming language. IV. Context cannot be placed dynamically; rather it has to be placed at a time of web page creation. V. No dynamically positioning. VI. Errors in HTML code are usually not fatal. VII. HTML is not an invention but it is an improved version of SGML (standard generalized markup language).	I. It stands for Dynamic hypertext markup language. II. DHTML is used to design dynamic web pages. III. DHTML used the DOM i.e., document object mode. IV. Contents on the web-page can be added, deleted or modified dynamically. V. It is possible Dynamic positioning mean to change in the position of an element dynamically. VI. If DHTML contains an VB script then we have possibility for trapping errors, single VB script provides error object. VII. It is a combination of technologies including HTML, java script, VB script Dom & Css.

Q7) Explain CSS positioning.

CSS POSITIONING

The CSS positioning properties allow you to position an element. Can also place an element behind another, and specify what should happen when an elements content is too big.

Elements can be positioned using the **top, bottom , left, right** properties. However, these properties will set with the method called **"position".**

Positioning method are of four types

1. static positioning
2. fixed positioning
3. relative positioning
4. Absolute positioning

1) **Static positioning** :- Html elements are positioned static by default. A static positioned element is always positiioned according to the normal flow of the page. Static positioned elements are not affected by the top,bottom,left,right properties.

2) **Fixed Positioned** :- An element with fixed positiioned relative to the browser window.

It will not move even if the window is scrolled:

Example:

```
<html>
<head>
```

```
<style type=”text/css”>
p.posfix
{ position : fixed;
top: 30px;
right:5px;
}
</style>
</head>
<body>
<p class=”pos-fixed”>Some more text</p>
<p><b>Note:</b>IE7 and IE8 supports the fixed value only if a !DOCTYPE is specified.
</body>
</html>
```

Relative Positioning

A relative positioned element is positioned relative to its normal position .

Example:

```
<html>

<head>
<style type=”text/css”>
h2.pos_left
{ position : relative;
left:-20px;
}
h2.pos_right
{
position:relative;
left:20px;
}
</style>
</head>
<body>
<h2> This is a heading with no position</h2>
<h2 class=”pos_left”>This heading is moved left according to its normal position </h2>
<h2 class=”pos_right”> This heading is moved right according to its normal positioned
</h2>
<p>Relative positioning moves an element RELATIVE to its original position.</p>
<p>The style “left:-20px” subtracts 20 pixels from the element's original left position.
</p>
<p>The style “left:20px” adds 20 pixels to the element's original left position.</p>
</body>
</html>
```

Absolute Positioning

An absolute position element is positioned of the first parent element that has a position other

than static .If no such element is found, the containing block is <html>:

Example:

```
<html>
<head>
<style type=”text/css”>
h2
```

```
{
position:absolute;
left:100px;
top:150px;
}
</style>
</head>
<body>
h2>This is a heading with an absolute position </h2>
<p>With absolute positioning, an element can be placed anywhere on a page. The  heading
below is placed 100px from the left of the page and 150px from the top of the  page.
</p>
</body>
</html>
```

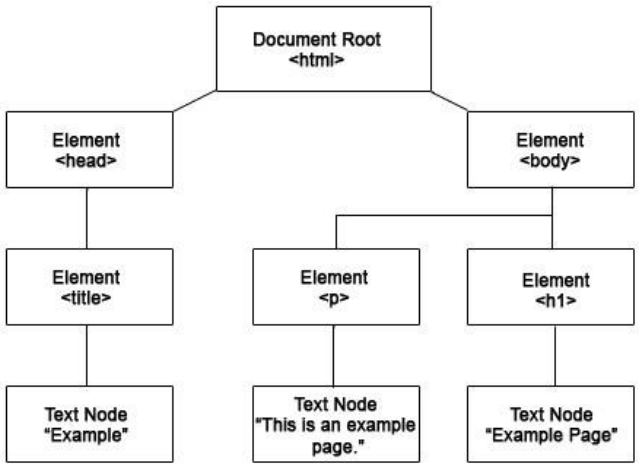
Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the obsolutely positioned element does not exist. Absolutely posoitoned elements can other elements.

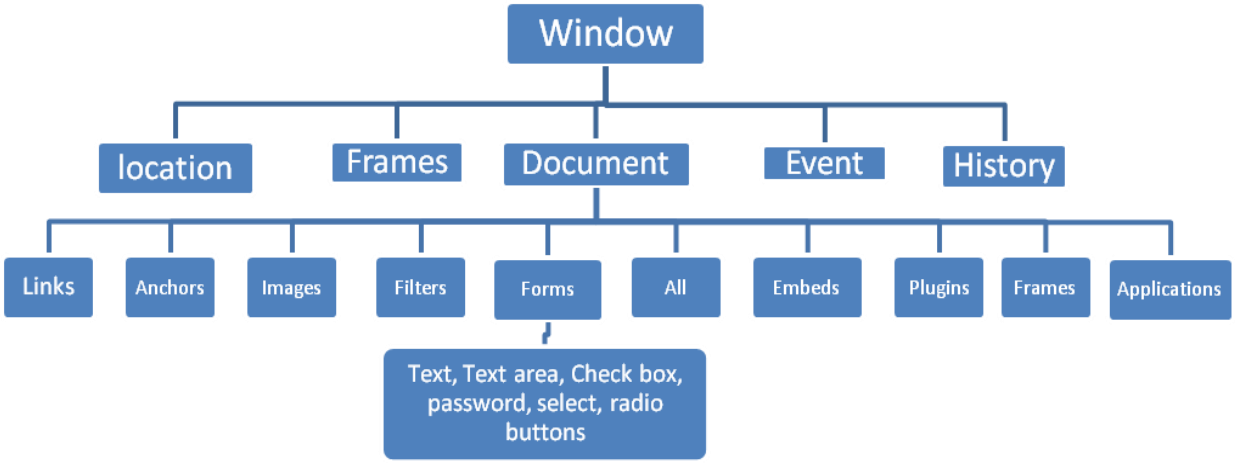
Q 8) Write about DOM?

The Document Object Model (DOM) is an Application Programming Interface (API) used in web browsers or other devices that give programmers and developers a way of accessing the elements within HTML and XML documents. The structure of the DOM for any document will resemble the actual structure of the markup of the document. For example, consider the below simple HTML document.

```
<html>
<head>
<title>Example</title>
</head>
<body>
<h1>Example Page</h1>
<p>This is an example page.</p>
</body>
</html>
```

The DOM for this document will include all of the elements and any text nodes within those elements. This code will create an object hierarchy as shown below.



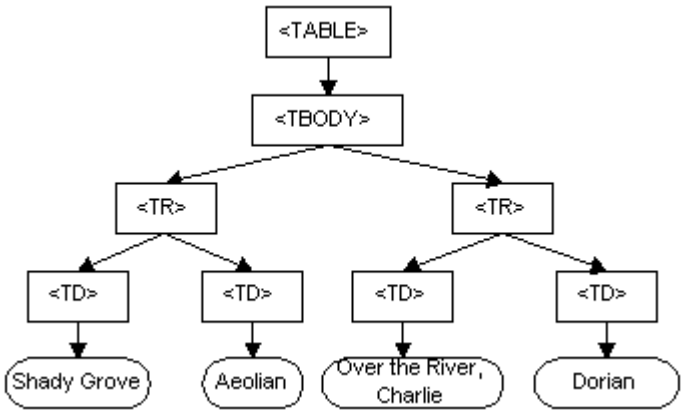


For each element under the document root (<html>), there is an element node, and these element nodes have text nodes containing the text that is within the element. If there were an element with attributes, an attribute node would be created for that element, and any text for the attribute would create a text node under that attribute node.

For example:

```
<TABLE>
    <TABLE>
    <TR>
    <TD>Shady Grove</TD>
    <TD>Aeolian</TD>
    </TR>
    <TR>
    <TD>Over the River, Charlie</TD>
    <TD>Dorian</TD>
    </TR>
    </TBODY>
    </TABLE>
```

A graphical representation of the DOM of the example table is:



the DOM identifies:

- the interfaces and objects used to represent and manipulate a document
- the semantics of these interfaces and objects - including both behavior and attributes
- the relationships and collaborations among these interfaces and objects

The most common programming language used in accessing the DOM is JavaScript, which is used very often in web sites. This allows for dynamic changes to be made to the DOM, which can include showing/hiding elements (such as text, tables, images, and entire divisions), moving elements, animating elements, and more.

In the past, the DOM had fundamental differences between browsers, but today has become much more standardized in modern browsers. This allows for easier cross-browser scripting to be performed by developers.

The DOM stands for documents object model. It gives access to every element in a document. The DHTML DOM allows authors direct, programmable access to the individual components of their web documents, from individual elements to containers. This access combined with the event model, allows the browser to react to user input, execute scripts the fly, & display the new context without downloading additional documents from a server.

Dom creates an object for each element on the page. A hierarchy of objects as displayed in the following figure.

The window object is the top-most object in the hierarchy. The document object, which is a child object of window object, represents, the page loaded in the browser window. A part from objects, the object model also contains collections. A collection is an array holding one or more objects. For example all collection on the document on the object represents all the elements in the document hierarchy.

Q9)Write about SPAN & DIV tags?

SPAN: - It allows authors to give style that could not be attached to a structural HTML element. SPAN may be used a selector in a style sheet & it also accepts the style, class & ID attributes. It is an inline element so it may be used just as elements such as EM and STRONG in HTML the difference between EM & strong is these carry structural meaning SPAN has no meaning.

```
Ex: <html>
        <title> example of SPAN</title>
        <style type=”text/css”>
        first words {font-variant: small-caps}
        </style>
        <body>
        <p> <SPAN class = first words> the first few words </SPAN> paragraph could be in small- caps.
<SPAN style = “font – family: Arial”>Arial </SPAN> </p>
        </body>
        </html>
```

DIV:-

This is similar to SPAN element in function. DIV stands for Division. It is a block level element. It may contain paragraphs, headings tables & even other divisions. This makes ideal for marking different classes of containers, such as a chapter, abstract, or note.

```
Ex: <Div class =note>
<h1> division </h1>
<p> the div element is defined in H 3.2, but only the align attribute permitted. </p>
<p> since Div may contain other block- containers, it is useful for marking less sections </p>
<p> the closing tag is required </p>
</DIV>
```

Example for div tag

```
<html>
    <head>
        <title>
        DIV and Span
        </title>
    </head>
        <style type="text/css">
    h1      {
        border: 1px solid black;
        background-color: 006478;
        }
        .nav

        {
        font: normal bold 12px/1.5 arial;

        background-color: efefef;
        text-align: center;
        color: white;
        padding: 10px;
        margin: 5px;
        border: 1px solid black;
        width: 200px;
        }
        </style>
    <body>

            <div class="nav">
            <h1>
                Home
            </h1>
            <h1>
                Products
            </h1>
            <h1>
                Services
            </h1>
            <h1>
                About
            </h1>
        </div>
    </body>
</html>
```

Example for span tag

```
<html>
    <head>
        <title>
            Span
        </title>

        <style type="text/css">
    span.emphasis

    {
      font: italic bold 12px/1 arial;
      color: red;
      background-color: yellow;
     }
        </style>
    </head>
    <body>
    The White Sox are clearly the <span
class="emphasis">superior</span> Chicago
baseball team, because they have
    <span class="emphasis">Ken Griffey,
Jr.</span>      </body>
</html>
```

Q 10) What are Filters and Transitions?

(Or)

Explain multimedia effecting filters and Transitions.

A) Filters and transitions enable us to create the Multimedia effects applying **filters** to text and images causes chances that are persistent. **Transition** allows us to transfer from our stage to another with a pleasant visual effect such as random dissolve.

you can apply various multimedia-style visual effects to your Web page. You can implement these effects in Web pages using Cascading Style Sheets (CSS) properties. By combining filters and transitions with basic scripting, you have a powerful tool for creating visually engaging and interactive documents.

FILTERS:

Visual filters are extensions to CSS properties that change the display of an object's contents.

Filters are applied to HTML controls through the filter property. The **filter** property is a string of filter descriptions that uses a function-like notation, but you do not need to know script to implement this property. The following syntax shows an Internet Explorer 5.5 **filter** declaration in a STYLE attribute.

Syntax:-

<ELEMENT STYLE="filter: filtername(Properties)" >

WHERE ELEMENT IS HTML TAG.

Ex:-<TD STYLE = "FILTER :FLIPH>WELCOME</TD>

Above example gives the effect using filters to flip text. The name of the filter property is the fliph, which flips the affected object horizontally.

We can apply more than one filter for an object.

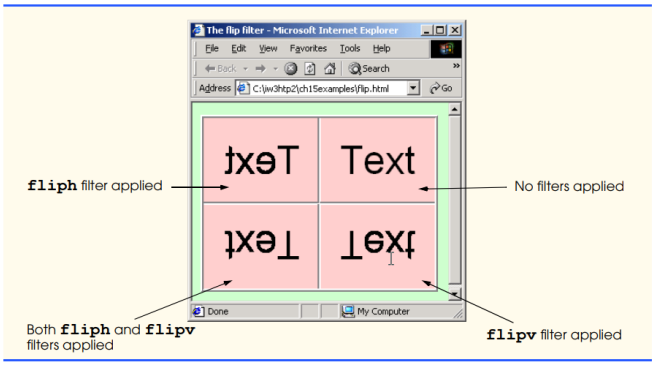
Ex: <TD STYLE = "FILTER: FLIPV SHADOW>STUDENTS</TD> IN ABOVE EXAMPLE TWO FILTERS NAMES ARE FLIPV AND SHADOW.

PROGRAM USING FILTERS: fliph and flipv

```
<html>
<head>
<title>The flip filter</title>
<style >
body { background-color: #CCFFCC }
table { font-size: 3em;
font-family: Arial, sans-serif;
      background-color: #FFCCCC;
      border-style: ridge ;
      border-collapse: collapse }
      td   { border-style: groove;
      border style:
      padding: 1ex }
      </style>
      </head>
      <body>
Outline
     <table>
```

```
<tr>
  <td style = "filter: fliph">Text</td>
  <td> Text </td>
</tr>
<tr>
<td style = "filter: flipv fliph"> Text </td>
<td style = "filter: flipv"> Text </td>
</tr>
</table>
</body>
</html>
```

Output



Specific elements that filters can be applied are: BODY, IMG, INPUT, MARQUEE, TABLE, TD, TEXTAREA, TH, TR, THEAD.

Specific elements that cannot have filters and transitions applied to them include:

- The embed and applet elements
- The select and option form elements
- The tr, tHead, tBody, and tFoot table elements

Some filter effects

Property	Argument	Description	Example
Blur	Add,direction,stremght	Makes the element blur	Filter:blur(add=true,direction=90, strength = 6
Fliph	None	Flips the element horizontally	Filter : fliph
Flipv	None	Flips the element vertically	Filter : flipv
invert	None	Renders the element in its reverse color and brightness values	Filter : invert;
Gray	None	Renders the element in black and white	Filter : gray
Mask	Color	Renders the element with the specified background color and transparent foreground color	Filter : mask(color= # ff0000)
Xray	None	Renders the element in black & white with revenge color and brightness value.	Filter: xray

Image filters are of 3 types

1) invert -it applies a negative image effect-dark areas become light and light areas become dark.

2) gray scale- The gray filter applies a grayscale image effect,in which all color is stripped from the image and all that remains is brightness data.

3) xray - the xray filter applies an xray effect,which is basically just an inversion of the grayscale effect.

program using image filters

```
<html>

<head>

<TITLE>Misc. Image Filters </TITLE>

<style>
.cap { front-weight:bold;
background-color : #DDDDAA;
test-align:center }
</style>
</head>
<body>
<table>
<tr class = "cap">
<td> normal </td>
<td> grayscale</td>
<td>xray</td>
<td> invert </td>
</tr>
<tr>
<td><img src = "abc.JPG" height=100 width = 100>
<td><img src = "abc.JPG" height=100 width = 100 STYLE="filter : gray"></td>
<td><img src = "abc.JPG" height=100 width = 100 STYLE="filter : xray"></td>
<td><img src = "abc.JPG" height=100 width = 100 STYLE="filter : invert"></td>
</tr>
</table>
</body>
</html>
```

Transitions :- Transitions are one of the most useful features of DHTML. With transitions, web designers can implement fast and easy visual effects on their web pages with simple HTML and very minimal script code.Transitions are particularly useful in slide show presentations to move from one slide to next called interpage transitions.

Transition are time-varying visual filters. Their use is to visually transition an object from one state to the next.

Transition filters are having two types of methods to initiate the visual effect.

1.apply() :- this method freezes the visual appearance of the control by applying the transition.

2.play():- It invokes the transition

3.Stop():- finish or stop the transition by the calling the STOP() method.

There are two types of Transition:-

1.Blend transition: The blend filter allows a simple fade-in and fade-out of an object with a specified duration. keyword is **blendTrans**

syntax:- {filter:blendTrans(duration=0.000)}

parameters:

duration: length of time the transition should take to complete

2.Reveal Transition : The Reveal transition filter is a set of predefined transition effects such as box-in,circle-out,horizontal and vertical swipes and soon that can be applied to any page object. keyword is **revealTrans**

predefined types are 0 to 23.

syntax:- {filter:revealTrans(duration:0.000,transition = 0 to 23)}

parameters :

duration: length of time the transition should take to complete .

transition: type of transition (which are predefined)

The following steps outline how to apply a transition on an image:

1. Define the image, specifying the desired transition.
2. Write a function to start the transition
3. finally ,call the function

Q11) What is Event Handling? Explain types of even Handling.

A) An event is a notification that occurs in response to an action, such as a change in state, or as a rolesult of the user clicking the mouse or pressing a key while viewing the document.
An event handles is a code, typically a function or routine written in a scripting language that receives control when the corresponding event occurs. For ex: with event handles user can do something with an element when an event occurs. When the user clicks an element, when the page loads, when a form is submitted etc..

```
<h1 onclick =style. Color = "red"> click on </h1>
```

The example above defines a header that can also add a script in the head section of the page & then call the function from the event handles.

```
<Html>
<Head>
<script type =”text/JavaScript”>
Function change color ( )
{
Document. GetelementByID (‘header’). Style. Color =”red”
}
</script> </head>
<Body>
<h1 id =”header” onclick =”change color ( )”>
Click on this </h1>
</body> </html>
```

Types of event Handlers

1. Key board Events
2. Mouse Events
3. Window Events

1. **Key board Events: -** Those events occur when the user presses or release a keyboard key. It has different events **like onkeydown & onkeyup** events fired when key changes states as it pressed

or released, respectively. These events fire for all keys on the key board, including shift state keys such as shift, ctrl & alt.

On key press: - This event fires when the user's keyboard input is translated to a character.

Ex: when user presses letter or number keys or a combination of shift keys, letters & numbers.

When a keyboard event occurs, the key code property of the event object contains the Unicode key code of the corresponding key. You can change which key is associated with the event by either changing the value of the key code property. You can cancel the event by returning zero or false.

The on help event is a special keyboard event that occurs when the users presses the help key (F_1).

Attribute	Value	Description
onkeydown	*script*	Fires when a user is pressing a key
onkeypress	*script*	Fires when a user presses a key
onkeyup	*script*	Fires when a user releases a key

2. **Mouse Events:-** These events occur when the user moves the mouse on clicks the left button. The on mouse move event fires when the user moves the mouse . On mouse over & on mouse out fires when the mouse moves in & out of an element. On mouse down & onmouse up events fire when the left mouse button changes state as it is pressed or released respectively.
   On click & ondblclick events fire when the user single clicks & double- clicks the button.
   When a mouse event occurs, the button property of the event object identifies which mouse button, If any pressed. The X & Y properties specify the location of the mouse point at the time of the event. The To element & from element properties specify the elements the mouse is moving to & from you can cancel a mouse click by returning false or setting the return value property to false.

3. **Window Events:-** These events occur on window object. The different window events are :
   The Onload events fires after the document is loaded and all the elements on the page are completely downloaded.
   The unload event fires immediately prior to the document being unloaded as when navigation to another document. Specifying code for the onload & onunload events can be done. On the body tag. These events actually occur on the window object.

Event	Occurs when
Onabort	A user aborts page loading
Onblur	A user leaves an object
Onchange	User changes the value of an object

Onclick	A user clicks on an object
Ondblclick	User double clicks on object
On focus	A User makes an object active
Onkey down	Key board is on its way down
Onkeypress	A key board key is pressed
Onkeyup	A key board key is released
Onload	A page is finished loading
Onmousedown	A user presses a mouse-button
Onmousemove	A cursor moves on an object
Onmoveover	A cursor moves over an object
Onmouseout	A cursor moves off an object
Onmouseup	A user releases a mouse- button
Onreset	A user resets a form
Onsubmit	A user submits a form
Onunload	A user closes page

Program using onload event:

```
<html>
<head>
<script>
Function mymessage()
{
Alert (“this message was triggered from the onload event”)
}
</script>
</head>
<body onload =”mymessage()”>
</body>
</html>
```

****

IMPORTANT QUESTIONS:-

1) What is style? Explain with example.
2) What is a selector? Explain types of selectors?
3) Explain DOM.
4) Explain Event handlers in dhtml?
5) Explain filters in dhtml
6) Explain types of stylesheets in css.
7) Difference between HTML and DHTML.
8) what is CSS?
9)what is DHTML.

UNIT-III: JAVA SCRIPT

- **Introduction**
- **Client side Java script**
- **Server side Java script**
- **Core features**
- **Data types and variables**
- **Operators**
- **Expressions and statements**
- **Functions**
- **Objects**
- **Array**
- **Date and math related objects**
- **Document object model**
- **Event handling**

Q1) What is scripting language?

A) A scripting language is a programming language designed for integrating and communicating with other programming languages. Some of the most widely used scripting languages are **JavaScript**, VBScript, PHP, Perl, Python, Ruby, ASP and Tcl. Since a scripting language is normally used in conjunction with another programming language, they are often found alongside HTML, Java or C++.

Introduction to Javascript:

JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web.

First appeared: December 4, 1995
Stable release: ECMAScript 2017 / June 2017
Typing discipline: Dynamic, duck
Developer: Netscape Communications Corporation, Mozilla Foundation, Ecma International
Designed by: Brendan Eich
Paradigm: Multi-paradigm: object-oriented (prototype-based), imperative, functional, event-driven

Q2) What is Java Script?

A) JavaScript is a very powerful **client-side scripting language**. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

- *JavaScript* was initially created to *"make webpages alive"*.

The programs in this language are called *scripts*. They can be written right in the HTML and execute automatically as the page loads.

Scripts are provided and executed as a plain text.

- When JavaScript was created, it initially had another name**: "LiveScript".** But Java language was very popular at that time, so it was decided that positioning a new language as a "younger brother" of Java would help.
- But as it evolved, JavaScript became a fully independent language, with its own specification called **ECMAScript.**
- At present, JavaScript can execute not only in the browser, but also on the server, or actually on any device where there exists a special program called the JavaScript engine.

- The browser has an embedded engine, sometimes it's also called a **"JavaScript virtual machine".**

There are at least *three* great things about JavaScript:

- Full integration with HTML/CSS.
- Simple things done simply.
- Supported by all major browsers and enabled by default.

Combined, these three things exist only in JavaScript and no other browser technology.

- Being a scripting language, **JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code**.
- When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it.
- The main advantage of JavaScript is that **all modern web browsers support** JavaScript. Internet Explorer, Google Chrome, Firefox or any other browser.
- JavaScript will be supported. Also, JavaScript **runs on any operating system** including Windows, Linux or Mac. Thus, JavaScript overcomes the main disadvantages of VBScript (Now deprecated) which is limited to just IE and Windows.
- **Tools required :** To start with, you need a text editor to write your code and a browser to display the web pages you develop. You can use a text editor of your choice including Notepad++, Visual Studio Code, Sublime Text, Atom or any other text editor you are comfortable with. You can use any web browser including Google Chrome, Firefox, Microsoft Edge, Internet Explorer etc.
- JavaScript has stormed the web technology and nowadays small software ventures to fortune 500, all are using **node js** for web apps. Recently wordpress.com has rewritten its dashboard in javascript, paypal also chose to rewrite some of its components in java script. Be it google/twitter/facebook, javascript is important for everyone. It is used in applications like single page applications, Geolocation APIs, net advertisements etc.

Q3) What is JavaScript? Write about client-side java script, serve-side JavaScript, Core JavaScript? (IMP)

A) JavaScript is Netscape's cross-platform, object-oriented scripting language.

JavaScript language are 1) core JavaScript
2) client-side JavaScript
3) server-side JavaScript

1)Core JavaScript contains a core set of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core

JavaScript can be extended for a variety of purposes by supplementing it with additional objects; Core JavaScript

Client-side and server-side JavaScript have the following elements in common:

- Keywords
- Statement syntax and grammar
- Rules for expressions, variables, and literals
- Underlying object model (although client-side and server-side JavaScript have different sets of predefined objects)
- Predefined objects and functions, such as such as Array, Date, and Mat

2) Client-side JavaScript Client Side JavaScript (CSJS) is an extended version of JavaScript that enables the enhancement and manipulation of web pages and client browsers. In a browser environment , your code will have access to things provided only by the browser, like the document object for the current page, the window, functions like alert that pop up a message, etc. The main tasks of Client side JavaScript are validating input, animation, manipulating UI elements, applying styles, some calculations are done when you don't want the page to refresh so often. In web developing it's the browser, in the user's machine, that runs this code, and is mainly done in javascript . Also, this code must run in a variety of browsers.

Advantages of client-side javaScript:

- **Speed**. Client-side JavaScript is very fast because it can be run immediately within the client-side browser. Unless outside resources are required, JavaScript is unhindered by network calls to a backend server. It also has no need to be compiled on the client side which gives it certain speed advantages (granted, adding some risk dependent on that quality of the code developed).
- **Simplicity**. JavaScript is relatively simple to learn and implement.
- **Popularity**. JavaScript is used everywhere in the web. The resources to learn JavaScript are .numerous. StackOverflow and GitHub have many projects that are using Javascript and the language as a whole has gained a lot of traction in the industry in recent years especially.
- **Interoperability**. JavaScript plays nicely with other languages and can be used in a huge variety of applications. Unlike PHP or SSI scripts, JavaScript can be inserted into any web page regardless of the file extension. JavaScript can also be used inside scripts written in other languages such as Perl and PHP.
- **Server Load**. Being client-side reduces the demand on the website server.
- **Extended Functionality**. Third party add-ons like Greasemonkey enable JavaScript developers to write snippets of JavaScript which can execute on desired web pages to extend its functionality.
- **Versatility**. Nowadays, there are many ways to use JavaScript through Node.js servers. If you were to bootstrap node.js with Express, use a document database like mongodb, and use JavaScript on the front-end for clients, it is possible to develop an entire JavaScript app from front to back using only JavaScript.

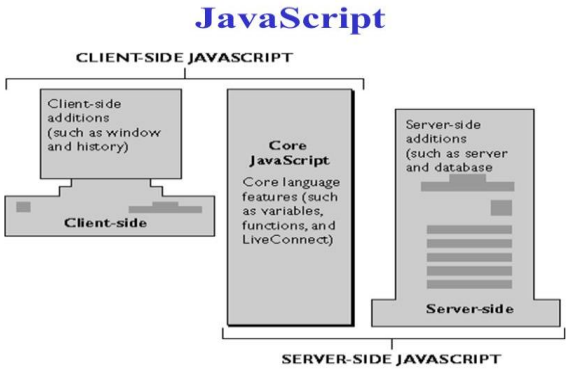
- **Updates**. Since the advent of EcmaScript 5 (the scripting specification that Javascript relies on), Ecma International has dedicated to updating JavaScript annually. So far, we have received browser support for ES6 in 2017 and look forward to ES7 being supported in future months.

3) Server-side JavaScript Server Side JavaScript (SSJS) is an extended version of JavaScript that enables back-end access to databases, file systems, and servers. Server side javascript, is javascript code running over a server local resources , it's just like C# or Java, but the syntax is based on JavaScript. A good example of this is Node.JS , with Node.JS you write javascript to program on the server side, and that code can be seen as normal C#, C, or any other server side language code. Moreover, with server-side code , you can still send javascript to the client-side, but there is a great difference between both, because the client side code is restricted to the clients machine resources, in terms of computing power and permissions. For example client-side javascript can't access the clients hard disk , while with server side you can access your server hard disk without any problem. The primary advantage to server-side scripting is the ability to highly customize the response based on the user's requirements, access rights, or queries into data stores.

Creating JavaScript applications is a two-stage process.

Step 1: In the first stage, you create HTML pages (which can contain both client-side and server-side JavaScript statements) and JavaScript files. You then compile all of those files into a single executable.

Step 2: In the second stage, a page in the application is requested by a client browser. The runtime engine uses the application executable to look up the source page and dynamically generate the HTML page to return. It runs any server-side JavaScript statements found on the page. The result of those statements might add new HTML or client-side JavaScript statements to the HTML page. The run-time engine then sends the resulting page over the network to the Navigator client, which runs any client-side JavaScript and displays the results.



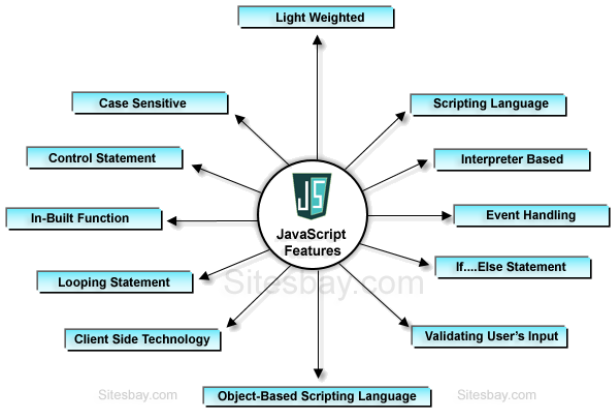
Client side programming includes any coding or computation or effects or animation or any sort of interaction your website performs with the user via browser . But server side programming is that which performs all the task in the server only . So the user is unaware of that. Few years ago JavaScript compilers were available only on the client machine

(browsers). So java script was called as a client side scripting language. On the client side JavaScript is run by v8 engine (Google chrome). But now in the server side also JavaScript is used. The v8 engine (with some modifications to provide the server functionality) is also used in the servers to run js codes. So, in both cases the language is the same, only the environment is different.

Q4). List out features of core javascript , client-side javascript, server-side javascript.

The features of **core javascript** are:

- JavaScript is a object-based scripting language.
- Giving the user more control over the browser.
- It Handling dates and time.
- It Detecting the user's browser and OS,
- It is light weighted.
- JavaScript is a scripting language and it is not java.
- JavaScript is interpreter based scripting language.
- JavaScript is case sensitive.
- JavaScript is object based language as it provides predefined objects.
- Every statement in javascript must be terminated with semicolon (;).
- Most of the javascript control statements syntax is same as syntax of control statements in C language.
- An important part of JavaScript is the ability to create new functions within scripts. Declare a function in JavaScript using **function** keyword.



Features of **client-side javaScript**:

1. Control Document Appearance and Content

2. Control the Browser

3. Interact with HTML Forms

4. Interact with the User

5. Read and Write Client State with Cookies

6. JavaScript can change the image displayed by an <img> tag to produce image rollover and animation effects.

7.JavaScript can interact with Java applets and other embedded objects that appear in the browser. JavaScript code can read and write the properties of these applets and objects and can also invoke any methods they define. This feature truly allows JavaScript to script Java.

8.As mentioned at the start of this section, JavaScript can perform arbitrary computation. JavaScript has a floating-point data type, arithmetic operators that work with it, and a full complement of standard floating-point mathematical functions.

9.The JavaScript Date object simplifies the process of computing and working with dates and times.

10.The Document object supports a property that specifies the last-modified date for the current document. You can use it to automatically display a timestamp on any document.

Q5) Explain JavaScript program.

A) **Computer program** is a list of "instructions" to be "executed" by a computer. In a programming language, these programming instructions are called **statements**. A **JavaScript program** is a list of programming **statements**. In HTML, JavaScript programs are executed by the web browser.

JavaScript Statements

JavaScript statements are composed of: Values, Operators, Expressions, Keywords, and Comments. Most JavaScript programs contain many JavaScript statements. The statements are executed, one by one, in the same order as they are written.JavaScript programs (and JavaScript statements) are often called JavaScript code.

Rules to follow to write javascript code

- **Semicolons ;**

Semicolons separate JavaScript statements. Add a semicolon at the end of each executable statement:

```
var a, b, c;   // Declare 3 variables
a = 5;         // Assign the value 5 to a
```

When separated by semicolons, multiple statements on one line are allowed:

```
a = 5; b = 6; c = a + b;
```

On the web, you might see examples without semicolons.
Ending statements with semicolon is not required, but highly recommended.

- **JavaScript White Space**

JavaScript ignores multiple spaces. You can add white space to your script to make it more readable.
The following lines are equivalent:

```
var person = "Hege";
var person="Hege";
```

A good practice is to put spaces around operators (= + - * /):

```
var x = y + z;
```

- **JavaScript Line Length and Line Breaks**

For best readability, programmers often like to avoid code lines longer than 80 characters. If a JavaScript statement does not fit on one line, the best place to break it after an operator:
Example

```
document.getElementById("demo").innerHTML =
"Hello World!";
```

- **JavaScript Code Blocks**

JavaScript statements can be grouped together in code blocks, inside curly brackets {...}.
The purpose of code blocks is to define statements to be executed together.

6Q) List out keywords of JavaScript.

JavaScript statements often start with a **keyword** to identify the JavaScript action to be performed. Here is a list of some of the keywords you will learn about in this tutorial:

Keyword	Description
break	Terminates a switch or a loop
continue	Jumps out of a loop and starts at the top
debugger	Stops the execution of JavaScript, and calls (if available) the debugging function

do ... while	Executes a block of statements, and repeats the block, while a condition is true
for	Marks a block of statements to be executed, as long as a condition is true
function	Declares a function
if ... else	Marks a block of statements to be executed, depending on a condition
return	Exits a function
switch	Marks a block of statements to be executed, depending on different cases
try ... catch	Implements error handling to a block of statements
var	Declares a variable

Note: JavaScript keywords are reserved words. Reserved words cannot be used as names for variables.

7Q) How to display output in JavaScript?

A) **JavaScript can "display" data in different ways:**

- Writing into an HTML element, using **innerHTML**.
- Writing into the HTML output using **document.write()**.
- Writing into an alert box, using **window.alert()**.
- Writing into the browser console, using **console.log()**.

1. **Using innerHTML**

To access an HTML element, JavaScript can use the **document.getElementById(id)** method.

The **id** attribute defines the HTML element. The **innerHTML** property defines the HTML content:

Example

```
<html>
<body>

<h1>My First Web Page</h1>
```

```
<p>My First Paragraph</p>

<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>
</body>
</html>
```

2. Using document.write()

It is convenient to use **document.write()**: it is same as printf() in C lang.

Example

```
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

3. Using window.alert()

You can use an alert box to display data:

Example

```
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

4. Using console.log()

For debugging purposes, you can use the **console.log()** method to display data. In browser press F12 to see console data.

Example

```
<html>
<body>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

8Q) How we will accept input in JavaScript?

A) JavaScript has three kind of popup boxes:1) Alert box, 2) Confirm box, and 3) Prompt box.

1)Alert Box

An alert box is often used if you want to make sure information comes through to the user. When an alert box pops up, the user will have to click "OK" to proceed.

Syntax
window.alert("*sometext*");
The **window.alert()** method can be written without the window prefix.
Example

```
alert("I am an alert box!");
```

2)Confirm Box

A confirm box is often used if you want the user to verify or accept something. When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed. If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

Syntax
window.confirm("*sometext*");
The **window.confirm()** method can be written without the window prefix.
Example

```
if (confirm("Press a button!"))
 {
   txt = "You pressed OK!";
}
 else
 {
   txt = "You pressed Cancel!";
}
```

3)Prompt Box
A prompt box is often used if you want the user to input a value before entering a page.When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax
window.prompt("*sometext*","*defaultText*");
The **window.prompt()** method can be written without the window prefix.

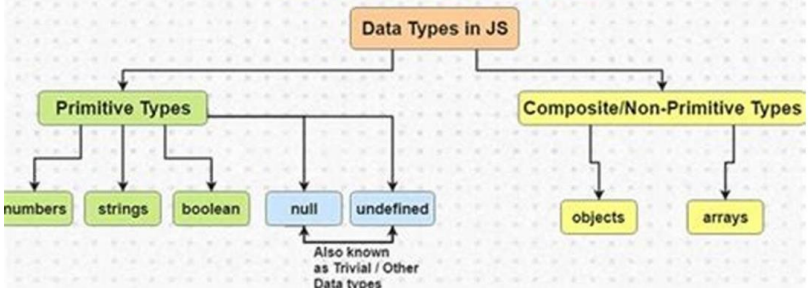
Example
```
var person = prompt("Please enter your name");
```
Addition of two numbers using JavaScript
```
<html>
<body>
<script>
var a = prompt("enter first number")
var b = prompt("enter second number")
var c = Number(a)+Number(b)
document.write("the addition of two nos is" + c);
</script>
</body>
</html>
```

- **Line Breaks**

To display line breaks inside a popup box, use a back-slash followed by the character n.
Example
```
alert("Hello\nHow are you?");
```

9Q) Write about data types of JavaScript.(IMP)

A) Two Kinds of Data in JavaScript
In JavaScript there are two different kinds of data: **primitive** and **non primitive**.
A primitive is simply a data type that is not an object, and has no methods.



1) Primitive Datatype:
In JS, there are five primitive data types:

Data Type	Description
String	represents sequence of characters e.g. "hello"
Number	represents numeric values e.g. 100
Boolean	represents boolean value either false or true
Undefined	represents undefined value
Null	represents null i.e. no value at all

1. **String**

Strings are used for storing text. Strings must be inside of either double or single quotes. In JS, Strings are immutable (they cannot be changed).

```
var str1 = 'hello, it is me';
var str2 = "hello, it's me";
```

2. **Number**

There is only one type of Number in JavaScript. Numbers can be written with or without a decimal point. A number can also be +Infinity, -Infinity, and NaN (not a number).

```
var num1 = 32;
var num2 = +Infinity;
```

3. **Boolean**

A boolean represents only one of two values: **true**, or **false**. Think of a boolean as an on/off or a yes/no switch.

```
var boo1 = true;
var boo2 = false;
```

4. **Undefined**

A variable that has no value is **undefined**.

```
var testVar;
console.log(testVar); // undefined
```

5. **Null**

Null has one value: **null.** It is explicitly nothing.

```
var nothing = null;
```

2)Non-primitive Data Type

1. **Object:** Object is a non-primitive data type in JavaScript. It is like any other variable; the only difference is that an object holds multiple values in terms of properties and methods. Properties can hold values of primitive data types and methods are functions

2**. Array:** An array is a special type of variable, which can store multiple values using special syntax. Every value is associated with numeric index starting with **0 (zero)**. .

10Q) Explain objects in JavaScript

A) Object: Object is a non-primitive data type in JavaScript. It is like any other variable, the only difference is that an object holds multiple values in terms of properties and methods. Properties can hold values of primitive data types and methods are functions.

JavaScript Objects

A JavaScript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.

JavaScript is an object-based language. Everything is an object in JavaScript.

JavaScript is template based not class based. Here, we don't create class to get the object. But, we direct create objects.

Creating Objects in JavaScript

There are 3 ways to create objects.

1. By object literal
2. By creating instance of Object directly (using new keyword)
3. By using an object constructor (using new keyword)

1) JavaScript Object by object literal

The syntax of creating object using object literal is given below:

object={property1:value1, property2:value2 .....propertyN:valueN}

As you can see, property and value is separated by : (colon).

Let's see the simple example of creating object in JavaScript.

```
<script>
emp={id:102,name:"Shyam Kumar",salary:40000}
document.write(emp.id+" "+emp.name+" "+emp.salary);
</script>
```

Output of the above example

102 Shyam Kumar 40000

2) By creating instance of Object

The syntax of creating object directly is given below:

var objectname=new Object();

Here, **new keyword** is used to create object.

Let's see the example of creating object directly.

```
<script>
var emp=new Object();

emp.id=101;
emp.name="Ravi Malik";
emp.salary=50000;
document.write(emp.id+" "+emp.name+" "+emp.salary);
</script>
```

Output of the above example

```
101 Ravi 50000
```

3) By using an Object constructor

Here, you need to create function with arguments. Each argument value can be assigned in the current object by using this keyword.

The **this keyword** refers to the current object.

The example of creating object by object constructor is given below.

```
<script>
function emp(id,name,salary){
this.id=id;
this.name=name;
this.salary=salary;
}
e=new emp(103,"Vimal Jaiswal",30000);

document.write(e.id+" "+e.name+" "+e.salary);
</script>
```

Output of the above example

```
103 Vimal Jaiswal 30000
```

Defining method in JavaScript Object

We can define method in JavaScript object. But before defining method, we need to add property in the function with same name as method.

The example of defining method in object is given below.

```
<script>
function emp(id,name,salary){
this.id=id;
this.name=name;
this.salary=salary;

this.changeSalary=changeSalary;
function changeSalary(otherSalary){
this.salary=otherSalary;
}
}
e=new emp(103,"Sonoo Jaiswal",30000);
document.write(e.id+" "+e.name+" "+e.salary);
e.changeSalary(45000);
document.write("<br>"+e.id+" "+e.name+" "+e.salary);
</script>
```

11Q) Explain Date object in JavaScript?(IMP)

Date: JavaScript provides Date object to work with date & time including days, months, years, hours, minutes, seconds and milliseconds.
The following example shows how to display current date and time using Date object in JavaScript.
Example: Current Local Date

```
Date(); //current date
```

or

```
var currentDate = new Date(); //current date
```

As you can see in the above example, we can display current date and time either by calling Date as function or creating an object with **new** keyword.

In order to work with date other than current date and time, we must create a date object by specifying different parameters in the Date constructor.

Syntax:
var dt = new Date();

var dt = new Date(milliseconds);

var dt = new Date('date string');

var dt = new Date(year, month[, date, hour, minute, second, millisecond]);
As per the above syntax, the following parameters can be specified in Date constructor.

- **No Parameter:** Date object will be set to current date & time if no parameter is specified in the constructor.
- **Milliseconds:** Milliseconds can be specified as numeric parameter. The date object will calculate date & time by adding specified numeric milliseconds from mid night of 1/1/1970
- **Date string**: String parameter will be treated as a date and will be parsed using Date.parse method.

- **year**: Numeric value to represent year of a date.
- **month**: Numeric value to represent month of a date. Starts with 0 for January till 11 for December
- **date**: Numeric value to represent day of a date (optional).
- **hour**: Numeric value to represent hour of a day (optional).
- **minute**: Numeric value to represent minute of a time segment (optional).
- **second**: Numeric value to represent second of a time segment (optional).

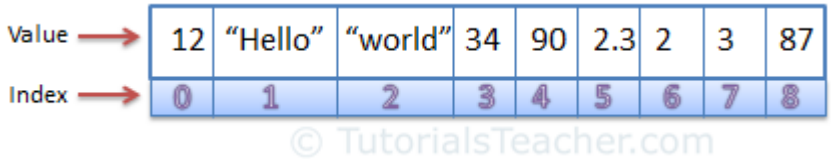
Example: Create Date by Specifying Milliseconds
```
var date1 = new Date(0);  // Thu Jan 01 1970 05:30:00

var date2 = new Date(1000); // Thu Jan 01 1970 05:30:01

var date3 = new Date(5000); // Thu Jan 01 1970 05:30:05
```

12Q) What is an Array object? Explain arrays in JavaScript. (IMP)

A) JavaScript Array

We have learned that a variable can hold only one value, for example `var i = 1`, we can assign only one literal value to i. We cannot assign multiple literal values to a variable i. To overcome this problem, JavaScript provides an array.

An array is a special type of variable, which can store multiple values using special syntax. Every value is associated with numeric index starting with 0. The following figure illustrates how an array stores values.



JavaScript Array

Array Initialization
An array in JavaScript can be defined and initialized in two ways,

1. array literal and
2. Array constructor

1.Array Literal
Array literal syntax is simple. It takes a list of values separated by a comma and enclosed in square brackets.

Syntax:

var <array-name> = [elemento, element1, element2,... elementN];

The following example shows how to define and initialize an array using array literal syntax.

Example: Declare and Initialize JS Array

```
var stringArray = ["one", "two", "three"];

var numericArray = [1, 2, 3, 4];

var decimalArray = [1.1, 1.2, 1.3];

var booleanArray = [true, false, false, true];

var mixedArray = [1, "two", "three", 4];
```

Array Constructor
You can initialize an array with Array constructor syntax using **new** keyword.

The Array constructor has following three forms.

var arrayName = new Array();

var arrayName = new Array(Number length);

var arrayName = new Array(element1, element2, element3,... elementN);

Accessing Array Elements
An array elements (values) can be accessed using index (key). Specify an index in square bracket with array name to access the element at particular index. Please note that index of an array starts from zero in JavaScript.

Example: Access Array Elements

```
var stringArray = new Array("one", "two", "three", "four");

stringArray[0]; // returns "one"
stringArray[1]; // returns "two"
stringArray[2]; // returns "three"
stringArray[3]; // returns "four"
```

JavaScript Array Methods(functions)

The Array object has many properties and methods which help developers to handle arrays easily and efficiently. You can get the value of a property by specifying arrayname.property and the output of a method by specifying arrayname.method().

1. **length property** --> If you want to know the number of elements in an array, you can use the length property.
2. **prototype property** --> If you want to add new properties and methods, you can use the prototype property.
3. **reverse method** --> You can reverse the order of items in an array using a reverse method.
4. **sort method -->** You can sort the items in an array using sort method.
5. **pop method** --> You can remove the last item of an array using a pop method.
6. **shift method** --> You can remove the first item of an array using shift method.
7. **push method** --> You can add a value as the last item of the array.

Points to Remember :

1. An array is a special type of variable that stores multiple values using a special syntax.
2. An array can be created using array literal or Array constructor syntax.
3. Array literal syntax: var stringArray = ["one", "two", "three"];
4. Array constructor syntax:var numericArray = new Array(3);
5. A single array can store values of different data types.
6. An array elements (values) can be accessed using zero based index (key). e.g. array[0].
7. An array index must be numeric.
8. Array includes length property and various methods to operate on array objects.

13Q) Explain Scope of variables in JavaScript.

A) **Variable Declaration**

Java script did not provide any data types for declaring variables and a variable in java script can store any type of value. Hence java script is loosely typed language. We can use a variable directly without declaring it.

Only var keyword are use before variable name to declare any variable.

Syntax: **var varianlename;**
Ex: **var a;**

Rules to declaring a variable

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

Example of Variable declaration in JavaScript

```
<script>
var a=10;
var b=20;
var c=a+b;
document.write(c);
</script>
```

Output
30

Variable Scope

Variable scope in JavaScript is the region of the code where a particular variable can be accessed or modified.

Types of Scopes in JavaScript:

- Block scope
- Function scope
- Local scope
- Global scope

• **Block scope**

Earlier JavaScript had only Global Scope and Function Scope. **let** and **const** are the two new important keywords ,these two keywords provide Block Scope in JavaScript. Variables that are declared inside a { } block cannot be accessed from outside the block.

Example: Below is the example of **let keyword.**

```
{
 let x = 2;
}
x cannot be used here
```

Example: Below is the example of **var keyword.**

```
{
 var x = 2;
}
x can be used here
```

Variables declared with the *var* keyword cannot have block scope and they can be declared inside a { } block and can be accessed from outside the block.

Function scope

JavaScript has function scope and each function creates a new scope. Variables defined inside a function are not accessible from outside the function and variables declared with **var**, **let** and **const** are quite similar when declared inside a function.

Example: Below is an example of **var keyword.**

```
function myFunction() {
  var firstName = "Krishna";   // Function Scope
}
```

Example: Below is an example of **let keyword.**

```
function myFunction() {
 let firstName = "Krishna";   // Function Scope
}
```

Example: Below is an example of **const keyword.**

```
function myFunction() {
  const firstName = "Krishna";   // Function Scope
}
```

Local scope

Variables declared inside a function become local to the function. Local variables are created when a function starts and deleted when the function is executed. Local variables have Function Scope which means that they can only be accessed from within the function.

Example:

```
// This part of code cannot use firstName
function myFunction() {
 let firstName = "Krishna";
 // This part of code can use firstName
}
This part of code cannot use firstName
```

Global scope

Variables declared Globally (outside of any function) have Global Scope and Global variables can be accessed from anywhere in a program. Similar to function scope variables declared with **var**, **let** and **const** are quite similar when declared outside a block.

let keyword:

```
let x = 2;       // Global scope
```

const keyword:

```
const x = 2;     // Global scope
```

var keyword:

```
var x = 2;       // Global scope
```

14Q) Explain operators in JavaScript (IMP)

A) **JavaScript Operators**

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

For example:

var sum=10+20;

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

There are following types of operators in JavaScript.

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

1. JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations on the operands. The following operators are known as JavaScript arithmetic operators.

Operator	Description	Example
+	Addition	10+20 = 30
-	Subtraction	20-10 = 10
*	Multiplication	10*20 = 200
/	Division	20/10 = 2
%	Modulus (Remainder)	20%10 = 0
++	Increment	var a=10; a++; Now a = 11
--	Decrement	var a=10; a--; Now a = 9

2. JavaScript Comparison Operators

The JavaScript comparison operator compares the two operands. The comparison operators are as follows:

Operator	Description	Example
==	Is equal to	10==20 = false
===	Identical (equal and of same type)	10==20 = false

!=	Not equal to	10!=20 = true
!==	Not Identical	20!==20 = false
>	Greater than	20>10 = true
>=	Greater than or equal to	20>=10 = true
<	Less than	20<10 = false
<=	Less than or equal to	20<=10 = false

3. JavaScript Bitwise Operators

The bitwise operators perform bitwise operations on operands. The bitwise operators are as follows:

Operator	Description	Example
&	Bitwise AND	(10==20 & 20==33) = false
|	Bitwise OR	(10==20 | 20==33) = false
^	Bitwise XOR	(10==20 ^ 20==33) = false
~	Bitwise NOT	(~10) = -10
<<	Bitwise Left Shift	(10<<2) = 40
>>	Bitwise Right Shift	(10>>2) = 2
>>>	Bitwise Right Shift with Zero	(10>>>2) = 2

4. JavaScript Logical Operators

The following operators are known as JavaScript logical operators.

Operator	Description	Example
&&	Logical AND	(10==20 && 20==33) = false
||	Logical OR	(10==20 || 20==33) = false
!	Logical Not	!(10==20) = true

5. JavaScript Assignment Operators

The following operators are known as JavaScript assignment operators.

Operator	Description	Example
=	Assign	10+10 = 20
+=	Add and assign	var a=10; a+=20; Now a = 30
-=	Subtract and assign	var a=20; a-=10; Now a = 10
*=	Multiply and assign	var a=10; a*=20; Now a = 200
/=	Divide and assign	var a=10; a/=2; Now a = 5
%=	Modulus and assign	var a=10; a%=2; Now a = 0

6. JavaScript Special Operators

The following operators are known as JavaScript special operators.

Operator	Description
(?:)	Conditional Operator returns value based on the condition. It is like if-else.
,	Comma Operator allows multiple expressions to be evaluated as single statement.
delete	Delete Operator deletes a property from the object.
in	In Operator checks if object has the given property
instanceof	checks if the object is an instance of given type
new	creates an instance (object)
typeof	checks the type of object.
void	it discards the expression's return value.
yield	checks what is returned in a generator by the generator's iterator.

15Q)Explain JavaScript program with example.

A)Javascript example is easy to code. JavaScript provides 3 places to put the JavaScript code: within body tag, within head tag and external JavaScript file.

Let's create the first JavaScript example.

```
1. <script type="text/javascript">
2. document.write("JavaScript is a simple language for javatpoint learners");
3. </script>
```

The **script** tag specifies that we are using JavaScript.

The **text/javascript** is the content type that provides information to the browser about the data.

The **document.write()** function is used to display dynamic content through JavaScript. We will learn about document object in detail later.

3 Places to put JavaScript code

1. Between the body tag of html
2. Between the head tag of html
3. In .js file (external javaScript)

1) JavaScript Example: code between the body tag

In the above example, we have displayed the dynamic content using JavaScript. Let's see the simple example of JavaScript that displays alert dialog box.

```
<script type="text/javascript">
 alert("Hello Javatpoint");
</script>
```

2) JavaScript Example : code between the head tag

Let's see the same example of displaying alert dialog box of JavaScript that is contained inside the head tag.

In this example, we are creating a function msg(). To create function in JavaScript, you need to write function with function_name as given below.

To call function, you need to work on event. Here we are using onclick event to call msg() function.

```
<html>
<head>
<script type="text/javascript">
function msg()
{
 alert("Hello Javatpoint");
}
</script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

3) External JavaScript file

We can create external JavaScript file and embed it in many html page.

It provides **code reusability** because single JavaScript file can be used in several html pages.

An external JavaScript file must be saved by .js extension. It is recommended to embed all JavaScript files into a single file. It increases the speed of the webpage.

Let's create an external JavaScript file that prints Hello Javatpoint in a alert dialog box.

message.js

```
function msg()
{
 alert("Hello Javatpoint");
}
```

Let's include the JavaScript file into html page. It calls the JavaScript function on button click.

index.html

```
<html>
<head>
```

```
<script type="text/javascript" src="message.js"></script>
</head>
<body>
<p>Welcome to JavaScript</p>
<form>
<input type="button" value="click" onclick="msg()"/>
</form>
</body>
</html>
```

16Q)Explain conditional statements in JavaScript?

A) Conditional stements in java script are:

1) If statement
2) Switch statement

1)If statements

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

- If Statement
- If else statement
- if else if statement

- **JavaScript If statement**

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

```
if(expression)
{
//content to be evaluated
}
```

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

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

- **JavaScript If...else Statement**

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

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

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

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

- **JavaScript If...else if statement**

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

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

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

```
<script>
var a=20;
```

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

2. JavaScript Switch statement

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

The syntax of JavaScript switch statement is given below.

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

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

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

```
<script>
var grade='B';
var result;
switch(grade){
case 'A':
result="A Grade";
break;
case 'B':
result="B Grade";
break;
case 'C':
```

```
result="C Grade";
break;
default:
result="No Grade";
}
document.write(result);
</script>
```

17Q) Explain looping statements in JavaScript?(IMP)

A) JavaScript Loops

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

There are four types of loops in JavaScript.

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

1) JavaScript For loop

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

```
for (initialization; condition; increment)
{
   code to be executed
}
```

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

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

2) JavaScript while loop

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

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

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

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

3) JavaScript do while loop

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

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

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

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

18Q) Write about functions in JavaScript.

A) A JavaScript function is a block of code designed to perform a particular task.
A JavaScript function is executed when "something" invokes it (calls it).
JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

Advantage of JavaScript function

There are mainly two advantages of JavaScript functions.

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

Syntax
A JavaScript function is defined with the **function** keyword, followed by a **name**, followed by parentheses **()**.
Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
The parentheses may include parameter names separated by commas:
(*parameter1, parameter2, ...*)
The code to be executed, by the function, is placed inside curly brackets: **{}**

```
function name(parameter1, parameter2, parameter3)
 {
   code to be executed
}
```

Ex:
Function **parameters** are listed inside the parentheses () in the function definition.
Function **arguments** are the **values** received by the function when it is invoked.
Inside the function, the arguments (the parameters) behave as local variables.
A Function is much the same as a Procedure or a Subroutine, in other programming languages.
Function Invocation
The code inside the function will execute when "something" **invokes** (calls) the function:
When an event occurs (when a user clicks a button)
When it is invoked (called) from JavaScript code
Automatically (self invoked)
Function Return
When JavaScript reaches a **return statement**, the function will stop executing.
If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.
Functions often compute a **return value**. The return value is "returned" back to the "caller":

Sample program for functions
Program no :1

```
<html>
<body>
        <h1>Demo: JavaScript function</h1>

        <script>
                function ShowMessage() {
```

```
                alert("Hello World!");
            }

            ShowMessage();
    </script>
</body>
</html>
```

Program : 2

```
<html>
<head>
<script>
function add()
{
var a,b,c;
a=Number(document.getElementById("first").value);
b=Number(document.getElementById("second").value);
c= a + b;
document.getElementById("answer").value= c;
}
</script>
</head>
<body>
Enter the First number : <input id="first">
Enter the Second number: <input id="second">
<button onclick="add()">Add</button>
<input id="answer">
</body>
</html>
```

19Q) Explain JavaScript DOM in detail.

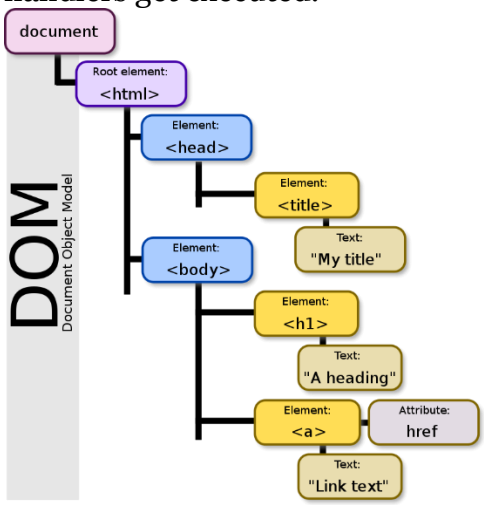
A) The **document object** represents the whole html document.

When html document is loaded in the browser, it becomes a document object. It is the **root element** that represents the html document. It has properties and methods. By the help of document object, we can add dynamic content to our web page.

```
window.document
```

The DOM represents a document with a tree Structure so that programs can read , access and change the document structure, style and content.

DOM methods allow programmatic access to the tree; with them one can change the structure, style or content of a document.
Nodes can have event handlers attached to them. Once an event is triggered, the event handlers get executed.



Every Browser uses some document object model to make web pages accessible via JavaScript

20Q) Explain math objects in JavaScript.(IMP)

A) The JavaScript **Math** object provides a common and easily accessible resource for mathematical formulas and informations.It enables users to work with advanced mathematical calculations like, logarithms, trignometry, roots , random numbers etc.

Calculations performed using Math Object get executed ***faster*** compared to the same task performed using regular JavaScript. It provides a number of properties and methods to perform such computations.

Math object is a **built-in-object** , hence you need not create an instance of Math object before using it using keyword ***new***.

JavaScript Math Object: Constants and Simple Calculations.

The Math object stores a lot of mathematical constansts, like log values, PI. It can also be used to compute power and square roots of a number.

In the below demo, constants and some simple mathematical operations are performed using Math Object .

Javascript Math Object: Rounding off Numbers

The Math object provides methods of **ceil(), floor() and round()** to round off the numbers both upwards and downwards

All of these methods truncate the number provided as an argument to return a whole number rounded up based on the method used.

floor() : Method rounds numbers to the next ***lowest*** whole number, and then removes all values after decimal point.

Syntax: math.floor(value)

Ex:math.floor(4.4)

ceil() : Method rounds numbers to the next ***largest*** whole number, and then removes all values after decimal point.

Syntax: math.ceil(value)

Ex:math.ceil(4.4)

round() : Method rounds numbers to the next ***largest*** whole number for decimal values greater than ***.5*** and to the ***lowest***next integer for values less than ***0.5***.

Syntax: math.round(value)

Ex:math.round(4.8)

Javascript Math Object: Generating Random Numbers

There is always a need to generate a random number to display some random image, message, ads etc in an unbaised way, for that very purpose we use **Math.random()** method.

Syntax : Math.random()

The method **Math.random()** generates a random fraction number between **0 to 1**, hence to have integer you should multiply it with integers and then truncate numbers after decimal point using **Math.floor()** method.

Javascript Math Object: *min()* and *max()* Methods

The **Math** object provides methods of **min()** and **max()** are used to find the smallest and largest within a group of number.

Min(): To find the smallest with in a group

Syntax: math.min ()

Ex: math.min (0, 150, 30, 20, -8, -200);

Max(): To find the largest within a group

Syntax: math.max ()

Ex: math.max (0, 150, 30, 20, -8, -200);

EVENTS AND EVENT HANDLERS: General information about Events – Event – OnAbort – OnClick - Ondbl click - Ondrag drop – Onerror - Onfocus - Onkey Press – Onkey Up – Onload - Onmouse Down – Onmouse Move - Onmouse Out – Onmouse Over - Onmove - Onrest – Onresize - Onselect - On submit - Onunload.

1) What is an Event? Explain Event Handlers in JavaScript?

A) Events are essentially the actions that occur on a web app due to user interaction, such as clicking a button. In JavaScript, when an event occurs, the app fires the event, which is kind of a signal that an event has occurred. The app then automatically responds to the user in the form of output, thanks to event handlers in JavaScript. An event handler is essentially a function with a block of code that is executed or triggered when a specific event fire.

Sometimes, when an event occurs, it triggers multiple events. This is because web elements in an app are often nested.

There are common types of events:

- **Keyboard/touch events:** Occur when a user presses or releases a key on the keyboard or performs an action with a touch-enabled smartphone, laptop or tablet.

  Event Handler Ex: OnkeyPress , OnkeyUp

- **Mouse hover events:** These events are fired when a user performs an action with the mouse, such as scrolling a page or moving the cursor.

  Event Handler Ex: OnClick ,Ondblclick, Ondragdrop, Onmouse Down, OnmouseMove, OnmouseOut , OnmouseOver, Onmove

- **Form/submit events:** Triggered when a user submits a form, modifies it, or resets it.

  Event Handler Ex: On submit , Onfocus

- **Window events:** Occurs when a user loads unloads the web page window

  Event Handler Ex: Onresize, Onunload, Onerror, OnAbort

Explain of Event Handlers

1) **OnAbort :-** The onabort event occurs when the user aborts the loading of an <img> or <input type="image"> element.
   - **Applies to:** Image. The onabort event occurs when the user aborts the loading of an <img> or <input type="image"> element.
   - **Triggered when:** The loading of the image is cancelled.Invoked when an event has been aborted. For example, the browser stops fetching media data before it is completely downloaded.
   - **Supports in** : IE browser

Syntax:

In HTML:

<*element* onabort="*myScript*">

Ex:- <img src="image_w3default.gif" Onabort="alert('Error: Loading of the image was aborted')">

In JavaScript:

***object*.onabort = function(){*myScript*};**

2) OnClick :- The onclick event occurs when the user clicks on an element.
   - **Applies to:** Button, Document, Checkbox, Link, Radio, Reset, Submit
   - **Triggered when:** The object is clicked on. This is the most frequently used event type which occurs when a user clicks the left button of his mouse. You can put your validation, warning etc., against this event type.
   - **Supports in: IE, chrome , mozilla**

Syntax:

In HTML:

<*element* onclick="*myScript*">

Ex: <input type="button" onclick="sayHello()" value="Say Hello" />

In JavaScript:

***object*.onclick = function(){*myScript*};**

Program :

```
<html>
  <head>

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

```
        function sayHello()
        {
          alert("Hello World")
        }
    </script>
      </head>

  <body>
    <p>Click the following button and see result</p>

    <form>
      <input type="button" onclick="sayHello()" value="Say Hello" />
    </form>

  </body>
</html>
```

3) Ondblclick :- The ondblclick event occurs when the user double-clicks on an element.

Applies to: Document, Link

Trigred when: The ondblclick event occurs when the user double-clicks on an element.

Supports in : IE, chrome , modzilla

In HTML:

Syntax: <*element* ondblclick="*myScript*">

Ex: <p ondblclick="myFunction()" >

In JavaScript:

***object*.ondblclick = function(){*myScript*};**

Sample program :

```
<html>
<body>
<p     ondblclick="myFunction()">Double-click this paragraph to trigger a
function.</p>
<p     id="demo"></p>
<script>
function myFunction()
 {
   document.getElementById("demo").innerHTML = "Hello World";
}
</script>
</body>
</html>
```

4)Ondrag (drag and drop): – The ondrag event occurs when an element or text selection is being dragged.
Applies to: all HTML Tags
Trigred when: The ondrag attribute fires when an element or text selection is being dragged.
Supports in : : IE(9.0), chrome(4.0) , modzilla (3.5)

Syntax: <*element* ondrag="*script*">
Ex: Drag me!

To make an element draggable, use the global TML5 **draggable** attribute.
There are many event attributes that are used, and can occur, in the different stages of a drag and drop operation:

Events fired on the draggable target (the source element)**:**

- ondragstart - fires when the user starts to drag an element
- ondrag - fires when an element is being dragged
- ondragend - fires when the user has finished dragging the element

Events fired on the drop target:

- ondragenter - fires when the dragged element enters the drop target
- ondragover - fires when the dragged element is over the drop target
- ondragleave - fires when the dragged element leaves the drop target
- ondrop - fires when the dragged element is dropped on the drop target

5) Onerror :–

Applies to: images and document
Triggred when: The onerror event is triggered if an error occurs while loading an external file (e.g. a document or an image).
Supports in : : IE, chrome, modzilla
Syntax: <*element* onerror="*myScript*"> in HTML
Ex: <img src="image.gif" onerror="myFunction()">

Syntax: *object*.onerror = function(){*myScript*}; in javascript
Ex: document.getElementById("myImg").onerror = function() {myFunction()};

Program:

```
<html>
<body>

<p>This example demonstrates how to assign an "onerror" event to an img element.</p>

<img src="image.gif" onerror="myFunction()">

<p id="demo"></p>
```

```
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "The image could not be loaded.";
}
</script>

</body>
</html>
```

6)Onfocus :– The onfocus event occurs when an element gets focus.

- **Applies to:** ALL HTML elements, EXCEPT: <base>, <bdo>,
, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style>, and <title>
- **Triggred when: focus event**
- **Supports in : all browsers**
- **Syntax:** <*element* onfocus="*myScript*">
  Ex: <input type="text" onfocus="myFunction()">
  Syntax:*object*.onfocus = function(){*myScript*};
  Ex: document.getElementById("fname").addEventListener("focus", myFunction);

Program:

```
<html>
<body>
<p>This example demonstrates how to assign an "onfocus" event to an input element.</p>
Enter your name: <input type="text" id="fname" onfocus="myFunction()">
<script>
function myFunction() {
  document.getElementById("fname").style.backgroundColor = "red";
}
</script>
</body>
</html>
```

7) Onkey Press: – The onkeypress event occurs when the user presses a key (on the keyboard).

Applies to: All HTML elements, EXCEPT: <base>, <bdo>,
, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style>, and <title>

Triggred when: The onkeypress event triggers when a key is pressed and released

Supports in : All browers

HTML Syntax: <*element* onkeyup="*myScript*">

Ex: <input type = "text" onkeyup = "sayHello()">

JAVASCRIPTSyntax: *object*.onkeyup = function()
{
myScript
}

As of Microsoft Internet Explorer 4.0, the **onkeyup** event fires and can be canceled for the following keys:

- Letters: A - Z (uppercase and lowercase)
- Numerals: 0 - 9
- Symbols: ! @ # $ % ^ & * ( ) _ - + = < [ ] { } , . / ? \ | ' ` " ~
- System: ESC, SPACEBAR, ENTER

Program:

```
<html>
  <head>
    <script>
      <!--
        function sayHello() {
          alert("A key is pressed.")
        }
      //-->
    </script>
  </head>
  <body>
    <input type = "text" onkeypress = "sayHello()">
  </body>
</html>
```

8) Onkey Up :– The onkeyup event occurs when the user releases a key (on the keyboard).

Applies to: All HTML elements, EXCEPT: <base>, <bdo>,
, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style>, and <title>

Triggred when: The onkeypress event triggers when a key is pressed and released

Supports in : All browers

Syntax: <*element* onkeyup="*myScript*">

Ex: <input type = "text" onkeypress = "sayHello()">

Syntax: *object*.onkeyup = function()
{
myScript
}

As of Microsoft Internet Explorer 4.0, the **onkeyup** event fires and can be canceled for the following keys:

- Letters: A - Z (uppercase and lowercase)
- Numerals: 0 - 9
- Symbols: ! @ # $ % ^ & * ( ) _ - + = < [ ] { } , . / ? \ | ' ` " ~
- System: ESC, SPACEBAR, ENTER

Program:

```
<html>
<body>

<p>A function is triggered when the user releases a key in the input field. The function transforms the character to upper case.</p>
Enter your name: <input type="text" id="fname" onkeyup="myFunction()">

<script>
function myFunction() {
  var x = document.getElementById("fname");
  x.value = x.value.toUpperCase();
}
</script>

</body>
</html>
```

9) Onload :– The onload event occurs when an object has been loaded.

- **Applies to:** onload is most often used within the <body> element to execute a script once a web page has completely loaded all content (including images, script files, CSS files, etc.).
- **Triggred when:**
- **Supports in :All browsers**

Syntax: <*element* onload="*myScript*">
Ex:<body onload=”myfunction()”>

The onLoad attribute is an event handler that instructs the browser to run a script when the visitor loads the page. This is commonly used to forward the visitor to another page, but can be used to trigger pop-up boxes, or execute a script based on the user's browser version.

10) Onmouse Down:- The onmousedown event occurs when a user presses a mouse button over an element.

The order of events related to the onmousedown event (for the left/middle mouse button):

- onmousedown
- onmouseup
- onclick

The order of events related to the onmousedown event (for the right mouse button):

- onmousedown
- onmouseup
- oncontextmenu

Applies to : - All HTML elements, EXCEPT: <base>, <bdo>,
, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style>, and <title>
Supports : All browers
Triggers when : when we press left/right/middle button of mouse.
Syntax: In HTML:

<*element* onmousedown="*myScript*">

In JavaScript:
***object*.onmousedown = function(){*myScript*};**
ex: Click the text!

program :

```
<html>
<body>

<p id="myP" onmousedown="mouseDown()" onmouseup="mouseUp()">
Click the text! The mouseDown() function is triggered when the mouse button is pressed down over this paragraph, and sets the color of the text to red. The mouseUp() function is triggered when the mouse button is released, and sets the color of the text to green.
</p>
<script>
function mouseDown() {
  document.getElementById("myP").style.color = "red";
}

function mouseUp() {
  document.getElementById("myP").style.color = "green";
}
</script>

</body>
</html>
```

11)OnmouseOver & Onmouse Out :– The onmouseout event occurs when the mouse pointer is moved out of an element, or out of one of its children.

This event is often used together with the onmouseover event, which occurs when the pointer is moved onto an element,

Applies to: All HTML elements, EXCEPT: <base>, <bdo>,
, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style>, and <title>
Triggred when: mouse moves
Supports in : all browsers
Syntax: <*element* onmouseout="*myScript*">

Ex: <h1 onmouseover="style.color='red'" onmouseout="style.color='black'">Mouse over this text</h1>

Program:

```
<html>
<body>

<h1 onmouseover="style.color='red'" onmouseout="style.color='black'">Mouse over this text</h1>

</body>
</html>
```

12) Onmouse Down – Onmouse up:- The **onmouseup** event occurs when a user releases a mouse button over an element. And The **onmousedown** event occurs when a user presses a mouse button over an element.

Applies to: All HTML elements, EXCEPT: <base>, <bdo>,
, <head>, <html>, <iframe>, <meta>, <param>, <script>, <style>, and <title>
Triggred when: mouse presses left and right button
Supports in : All Browers
Syntax: <*element* onmousedown="*myScript*">

Ex: Click the text!

Program :-

```
<html>
<body>

<p id="myP" onmousedown="style.color='yellow'" onmouseup="style.color='pink'">
Welcome to hyd
</p>

</body>
</html>
```

13) On submit: –The onsubmit event occurs when a form is submitted.

Applies to: <form> tag only

Triggred when:

Supports in : All browers

In HTML

Syntax:- <*element* onsubmit="*myScript*">

```
<form onsubmit="myFunction()">
Enter name: <input type="text">
<input type="submit">
</form>
```

In javaScript

Syntax:- *object*.onsubmit = function(){*myScript*};

Ex:
```
<form onsubmit="myFunction()">
        Enter name: <input type="text">
        <input type="submit">
    </form>

  <script>
function myFunction()
 {
 alert("The form was submitted");
}
</script>
```

Program :
```
<html>
<body>

<p>When you submit the form, a function is triggered which alerts some text.</p>

<form action="/action_page.php" onsubmit="myFunction()">
 Enter name: <input type="text" name="fname">
 <input type="submit" value="Submit">
</form>

<script>
function myFunction() {
   alert("The form was submitted");
}
</script>

</body>
</html>
```

14)Onunload:- The onunload event occurs once a page has unloaded (or the browser window has been closed).

Triggers when : onunload occurs when the user navigates away from the page (by clicking on a link, submitting a form, closing the browser window, etc.).
Applies to : <body>
Supports: In all browsers

In HTML:
Syntax: <*element* onunload="*myScript*">
In JavaScript:
***Syntax: object*.onunload = function(){*myScript*};**
ex: <body onunload="myFunction()">

program :

```
<html>
<body onunload="myFunction()">

<h1>Welcome to my Home Page</h1>

<p>Close this window or press F5 to reload the page.</p>
<p><strong>Note:</strong> Due to different browser settings, this event may not always work as expected.</p>

<script>
function myFunction() {
   alert("Thank you for visiting W3Schools!");
}
</script>

</body>
</html>
```

15)onselect() :- The onselect event occurs after some text has been selected in an element. The onselect event is mostly used on <input type="text"> or <textarea> elements.

Supports : All browers
Triggers when: Using the select() method of the HTML DOM Input Text Object to select some content of a text field. When this happens, the onselect event fires,
Applies to : <input type="file">, <input type="password">, <input type="text">, and <textarea>
Syntax:
Syntax
In HTML:
<*element* onselect="*myScript*">
In JavaScript:
***object*.onselect = function(){*myScript*};**
ex: <input type="text" onselect="myFunction()">

program :

```
<html>
<body>

Select some of the text: <input type="text" value="Hello world!" onselect="myFunction()">
```

```
<script>
function myFunction() {
   alert("You selected some text!");
}
</script>

</body>
</html>
```

16)Onresize() :- The onresize event occurs when the browser window has been resized. To get the size of an element, use the clientWidth, client Height, innerWidth, innerHeight, outerWidth, outerHeight, offsetWidth and/or offsetHeight properties.

Supports: All browsers
Trigger when: this event fires when click on window
Applies to: <body>

Syntax:
In HTML:
<*element* onresize="*myScript*">
Ex: <body onresize="myFunction()">

In JavaScript:
object.onresize = function(){*myScript*};

program :

```
<html>
<body onresize="myFunction()">
<p>Try to resize the browser window to display the windows height and width.</p>
<p id="demo"></p>
<p><strong>Note:</strong> this example will not work properly in IE8 and earlier.
IE8 and earlier do not support the outerWidth/outerHeight propery of the window
object.</p>
<script>
function myFunction() {
   var w = window.outerWidth;
   var h = window.outerHeight;
   var txt = "Window size: width=" + w + ", height=" + h;
   document.getElementById("demo").innerHTML = txt;
}
</script>

</body>
</html>
```

UNIT – V

XML

1) What is XML? Explain with example.

Ans: **INTRODUCTION:-**

History:- GML(General Markup Language)
↓
SGML(Standard Generalized Markup Language)
↓ ↓
HTML XML

In 1996 XML working group was formed under W3C (world web consortium.(W3C). In 1998 W3C introduced XML 1.5 XML means **Extensible Markup Language**. XML is not a programming language. It is a set of rules that allow to represent data in a structured manner

Its use can be taken from its name itself:-

→ Markup – Is a colletion of Tags.
→ XML Tags – Identify the content of data.
→ Extensible – user-defined tags.

XML is a Markup language much like HTML, but used for purpose different than what HTML is used for, An XML file or XML document contains data that is tagger XML documents are text documents.
HTML Tags are used to display the data, but XML tags are used to describe data and stored data.

Applications of XML:-

→ Redefined search results:- with XML- specific tags, search engineer can give users more redefined search results. A search engine seeks the term in the tags, rather than the entire document, giving the user more precise results.

→ EDI Transactions:- XML has made electronic data interchange (EDI) transactions accessible to a broader set of users. XML allows data to be exchanged.

→ Cell phones – XML data as sent to some cell phones, which is then formatted by the specification of the cellphones s\w designer to display text, images and even paly sounds.

→ File converters:- Many applications have been written to convert existing documents into XML stanadard.
Ex:-PDF to XML converters.

→ Voice XML:- converts XML documents into an audio format so that a user can listen to an XML document.

In 1996 XML working group was formed under W3C.
XML stands for extensible markup language. The word “Extensible” implies that a developer extend his ability to describe a document & define meaningful tags for his application. XML is a text- base format that lets developers describe, deliver & exchange structure data display & manipulation. It also facilitates the transfer of structured data between different server data marked up in XML can be targeted for computer, or a cell- phone.

XML focuses more on the structure of the data than its presentation.

Ex**:** **<Book>**

<Name> XML bible </Name>
<Author> Elliott Rusty Harold </Author>
<Publisher> ID4 Books Worldwide </Publisher>
<Price> $ 39.99 </Price>
</Book>

The above mentioned example denotes that XML extends your ability to describe a document by allowing you to define meaningful tags for your applications. You can create as many tags as per your applications requirement. In the above representation, there is no description of how to display the information. While this might appear undesirable, the separation of the structure of data & its display provides numerous.

Q 2**) What are the features of XML?**

A) Features of XML: -

XML is not only good for transmitting data from a server to a browser, but it is also ideal for passing data between applications. The following are the benefits of XML.

XML is a compliment of HTML: -

It is important to understand that XML is not a replacement for HTML. In future web development it is most likely that XML will be used to describe the data, while HTML will be used to format & display the same data.

XML can separate data from HTML: -

With XML data can be stored in separate XML files. This way you can concentrate on using HTML for data layout & display & be sure that chang3es in the underlying data will not require any changes to your HTML. i.e., with XML data is stored outside your HTML.

XML is used to Exchange Data: -

With XML data can be exchanged between incompatible systems. In the real computer systems & databases contain data incompatible formats. Converting the data of XML can greatly reduce the complexity and create data that can be read by many different types of applications.

XML and B2B (Business to Business): -

With XML financial information can be exchanged over the internet. XML is going to be the main language for exchanging financial information between business over the internet a lot of interesting B2B applications are under development.

XML can be used to share Data: -

With XML plain text files can be used to share data, since XML data is stored in plain text format, XML provides a software & hardware independent way of sharing data.

XML can be used to Store Data: -

With XML plain text files can be used to store data, XML can also be used to store data in files or in databases. Applications can be written to store & retrieve information from the store & generic applications can used to display the data.

XML can make your Data More Useful: -

With XML your data is available to more users, other clients & applications can access your XML files as data sources like they are accessing databases, your data can be made available to all kinds of " reading machines" (agents) & it is easier to make your data available for blind people or people with other disabilities.

XML can be used to create new languages: -

XML is the mother to (wireless (WAP)) Markup language, used to markup internet applications for hand held devices like mobile phones, is written in XML.

Q 3) What are the Rules of XML Syntax?

A) The Syntax Rules of XML are:

- All XML elements must have a closing tag: -

Every element should have closing tag

Ex: - computer

- XML tags are case sensitive.

Ex: <message> this is correct </message>

- All XML elements must be properly nested.

Ex: <I> computer</I>

- All XML documents must have a root tag: -

The first element in an XML document is the root tag. All other tags are nested within the root element.

Ex: - <root>
<Child>
<Sub child> ..... </Sub child>
</Child>
</root>

- Attribute values must always be quoted.

Ex: - <? XML version = "1.0" encoding = "iso-9">

- With XML, white space is preserved.
- With XML a new line is always stored as LF (Line feed).
- Comments in XML are similar to that of HTML.

Ex: - <! ..... This is a comment----- >

Q4) Explain XML documents?

A) XML documents use a self-describing and simple syntax.

ex: <?xml version="1.0" encoding = "ISO-8859-1">
<note>
< To> Tove </To>
<from> Jani</from>
<message>don't forget </message>
</note>

The first line in the document :- the XML declaration-defines the XML version and the character encoding used in the document.
In this case the document conforms to the 1.0 specification of XML and uses the ISO-8859-1 character set.

The next line describes the root element of the document.
The next 4 lines describe 4 child elements of the root i.e to,from,message
and finally the last that defines the end of the root element.
This file is saved with extension **".xml"**

Q5) What are the differences between XML and HTML?

1.XML was designed to carry data
2.XML is not a replacement for HTML.
3.XML & HTML were designed with different goals.
4.XML was designed to describe data and to focus on what data is.
5.HTML was designed to display data & to focus on how data looks.
6.HTML is about displaying information.
7.XML is about describing information.
1) XML is designed to carry data:-

XML describes and fouses on the data while HTML only displays an focuses on how data looks.

XML is used to store data in files and for sharing data between diverbe applications , XML is the best way to exchange information.

2) XML is free and extensible:- XML tags are not predefined .User must "invent" his tags .XML allows the user to define his own tags and document structure.

HTML tags are predefined . The author of HTML document can only use tags that are defined in the HTML standard.

3)XML tags are case sensitive:-

Unlike HTML, XML tags are case sensitive,

In HTML the following will work.

<body>-------<\body>

In XML, opening and closing tags must therefor be written with the same case.

Ex: <message> This is correct <\message>

4)XML element must be property nested

Improper nesting of tags makes no sense to xml. In xml all elements must be properly nested with in each other.

Ex: <b> <i>XML<\i> <\b>

In html some elements can be improperty nexted with in each other.

Ex: <b> <i> welcome <\b> <i>

5)XML is a complement to HTML:-

XML is not a replacement for HTML . It is important to understand that XML is not a replacement for HTML. In web development it is most likely that XML will be used to describe the data. While HTML will be used to format and display the same data.

Q6) Explain Elements & Attributes in XML? Give Examples

A) XML elements are the tags which include from the Starting tags such as body, attributes etc. XML elements are extensible & they have relationships. XML elements have simple naming rules.

xml is just plain text with the addition of some XML tags enclosed in angular brackets.

1.XML elements are extensible.
2.XML elements have relationships.
3.XML have content.
4.XML has element naming.
5.XML element can have attributes.

1) XML documents can be extended to carry more information.

Ex:
```
<note>
< To> Tove </To>
<Form> Jani</from>
<Body>don't forget </body>
</note>
```

We get output as message

To: Tove
From: Jani
Don't forget

Imagine that the author of the XML document added some extra information it as

```
<Note>
<date> 1999-808-01 </date>
<to>  Tove </to>
<From> Jani </from>
```

<Body> don't forget </body>
</note>

2) XML elements have relationships such as related as parents & children.

Ex: <Book>
<Title> XML</title>
<prod id = "33"> </prod>
</book>
Book is root element & title, prod are the child elements.

3)Elements have context:

It consists of different context types like content , mixed content,simple content.
the element can also have attributes.
In above example BOOK has element content because it contains other elements.

4) Element Naming:

XML must follow the naming rules such as
Names can contain letters, number & other characters.
Names must not start with a number of punctuation character
Names must not start with the letters XML (or XML or xml ….)
Names cannot contain spaces.

5) XML elements can have attributes:

Attributes: -

XML elements can have attributes in the start tag, just like HTML. Attributes are used to provide additional information about elements. Attributes often provide information that is not a part of the data.

Ex: <img src = "computer.gif">

Attribute values must always be enclosed in quotes, but either single or double quotes can be used.

Ex: <person name ="Sam">
Or
<person name= 'Sam'>

It is necessary to use single quotes like.

Ex: <gangster name = George "shotgun" Ziegler>

Element's Vs Attributes: -

Data can be stored in child elements or in attributes.

Ex:
```
<person gender ="female">
<First name> Anna </first name>
<Last name> Smith </last name>
</person>
<Person>
<Gender> Female </Gender>
<First name> Anna </First name>
<Last name> Smith </Last name>
</person>
```

In the above example, Gender is an attribute & in the last, gender is a child element both provides the same information.

Q7) Explain the importance of XML?

XML uses as the universal data format for integrated electronic business solutions.

Self describing Data: - the key to success unlike records in traditional database systems. XML does not require relational schemata; file description tables, external data type definitions etc…

Complete integration of all Traditional databases and formats: - XML documents can contain any imaginable data type from classical data like text or numbers or multimedia.
Modification to Data presentation, No programming required: - you can change the look & feel of documents or even entire websites with XSL style sheets without manipulating the data itself.
One server view of Data: - (Distributed) XML documents can consist of data from many different databases distributed over multiple servers in other words with XML entire world wide web is transformed into single all – encompassing database.
Internationalization: - It is of utmost importance for electronic business applications. XML supports multilingual documents & Unicode standard.
Open and Extensible: - XML's one of a kind open structure allows you to add other state of the art elements when needed.
Future Oriented Technology: - XML is the endorsed industry standard of the WWW consortium (W3C) & is supported of all leading software providers. Furthermore, XML is also the standard today in an increasing number of other industries. For example health care.

DOCUMENT OBJECT MODEL(DOM)

Q9) Explain DOM in xml?

What is DOM?

The Document Object Model (DOM) is a **programming interface** for **HTML(HyperText Markup Language)** and **XML**(**Extensible Markup Language**) documents. It defines the **logical structure** of documents and the way a document is accessed and manipulated.

The **HTML DOM** defines a standard way for accessing and manipulating HTML documents. It presents an HTML document as a tree-structure.

The **XML DOM** defines a standard way for accessing and manipulating XML documents. It presents an XML document as a tree-structure.

Knowing the XML DOM is a must for anyone working with XML. The DOM is a W3C (World Wide Web Consortium) standard.

The DOM is separated into 3 different parts / levels:
Core DOM - standard model for any structured document
XML DOM - standard model for XML documents
HTML DOM - standard model for HTML documents

Why is DOM Required?

The Document Object Model (DOM) is essential in web development for several reasons:

- Dynamic Web Pages
- Interactivity:
- Content Updates:
- Cross-Browser Compatibility:

According to the DOM, everything in an XML document is a node. Every XML element is an element node . The XML DOM views an XML document as a node-tree. All the nodes in the tree have a relationship to each other.

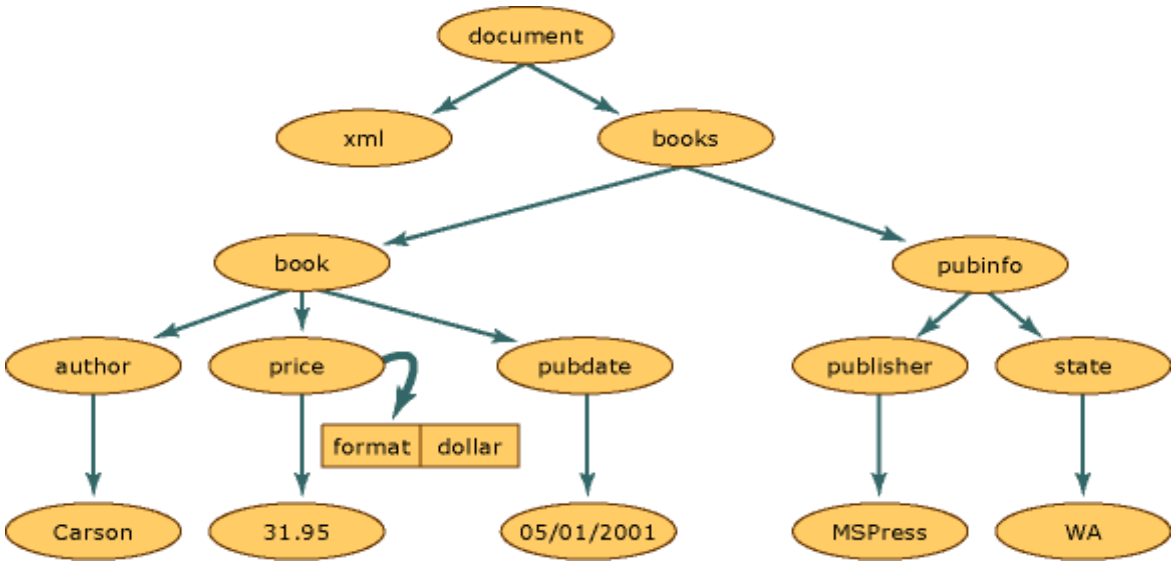
EXAMPLE

Input

XMLCopy

```
<?xml version="1.0"?>
 <books>
  <book>
    <author>Carson</author>
    <price format="dollar">31.95</price>
    <pubdate>05/01/2001</pubdate>
  </book>
  <pubinfo>
    <publisher>MSPress</publisher>
    <state>WA</state>
  </pubinfo>
 </books>
```

The following illustration shows how memory is structured when this XML data is read into the DOM structure.



XML document structure

The XML DOM views an XML document as a tree-structure. The tree structure is called a node-tree.
All nodes can be accessed through the tree. Their contents can be modified or deleted, and new elements can be created.
The node tree shows the set of nodes, and the connections between them.

The tree starts at the root node and branches out to the text nodes at the lowest level of the tree:

Explain XML Terminology.

DTD- document-type definition
XSD- XML Schema Definition

XSL- XML style sheet

What is DTD in XML ?

- DTD is a document-type definition.
- DTD contains a set of rules that control the structure and elements of XML files. When any XML file refers DTD file, it validates against those rules.
- DTD has validated elements and attributes that are defined inside the DTD document.
- They specify what elements can appear, their order, which attributes they can have, and what data types those attributes can contain.

What is XSD ?

- An XSD is similar to earlier XML schema languages, such as Document Type Definition (DTD), but it is a more powerful alternative as it provides greater control over the XML structure.
- XML Schema Definition or XSD is to describe and validate the structure and content of an XML document.
- It is primarily used to define the elements, attributes and data types the document can contain. The information in the XSD is used to verify if each element, attribute or data type in the document matches its description.

What is XSL?

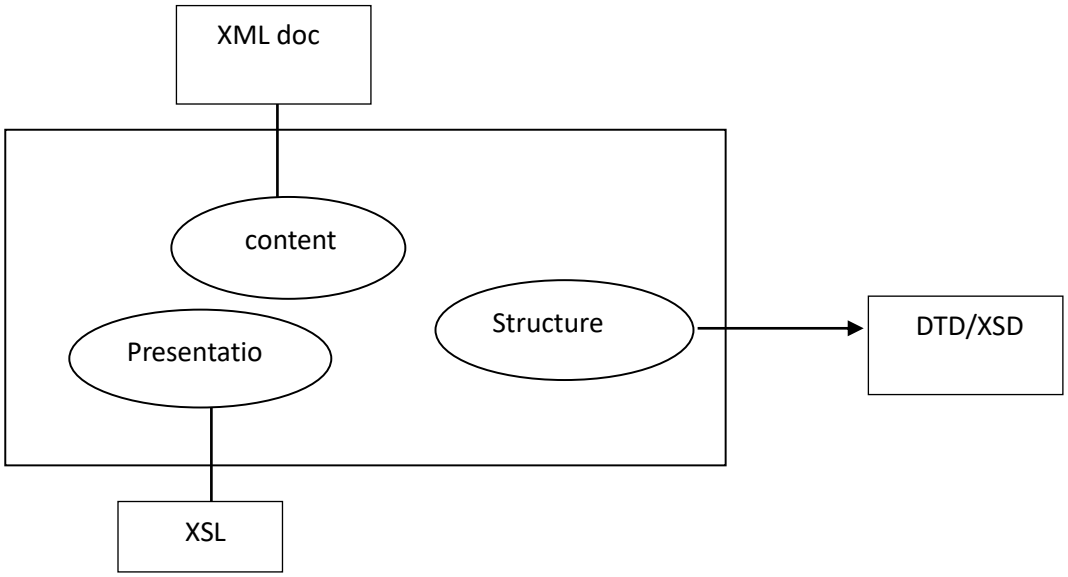
XSL is a language for expressing style sheets.

An XSL style sheet is, like with CSS, a file that describes how to display an XML document of a given type. XSL shares the functionality and is compatible with CSS2 (although it uses a different syntax). It also adds: A transformation language for XML documents: **XSLT**.

How Does It Work?

Styling requires a source XML documents, containing the information that the style sheet will display and the style sheet itself which describes how to display a document of a given type.

The following shows a sample XML file and how it can be transformed and rendered.



Q: Explain Extensible stylesheet language Transformation(XSLT)

A) Before XSLT, first we should learn about XSL. XSL stands for EXtensible Stylesheet Language. It is a styling language for XML just like CSS is a styling language for HTML.

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

What is XSL

In HTML documents, tags are predefined but in XML documents, tags are not predefined. World Wide Web Consortium (W3C) developed XSL to understand and style an XML document, which can act as XML based Stylesheet Language.

An XSL document specifies how a browser should render an XML document.

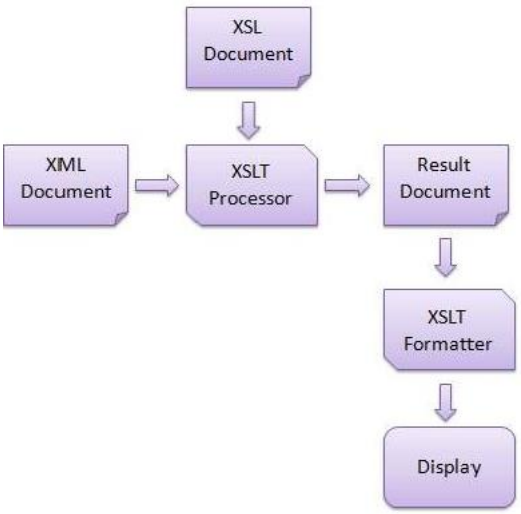
Main parts of XSL Document

- **XSLT:** It is a language for transforming XML documents into various other types of documents.
- **XPath:** It is a language for navigating in XML documents.
- **XQuery:** It is a language for querying XML documents.
- **XSL-FO:** It is a language for formatting XML documents.

How XSLT Works

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

Image representation:



XSLT Uses XPath

XSLT uses XPath to find information in an XML document. XPath uses path expressions to navigate in XML documents. XPath contains a library of standard functions. XPath is a major element in XSLT

In the transformation process, XSLT uses XPath to define parts of the source document that should match one or more predefined templates. When a match is found, XSLT will transform the matching part of the source document into the result document.

XSLT Elements

The XSLT elements from the W3C Recommendation (XSLT Version 1.0).

The links in the "Element" column point to attributes and more useful information about each specific element.

Elements supported in IE 5 may have NON-standard behavior, because IE5 was released before XSLT became an official W3C Recommendation.

element	Description
attributes	Adds an attribute
copy	Creates a copy of the current node
import	imports the contents of one style sheet into another.
include	includes the contents of one style sheet into another.
sorts	sorts the output.
stylesheet	defines the root element of a stylesheet
value-of	extracts the value of a selected node
variable	Declares a local or global

Example: a template contains rule to apply when a specified node is matched

```
<?xml version="1.0" encoding ="ISO-8859-1"?>
<xsl:stylesheet version="1.0" Xmlns:xsl="http//www.w3.org/1999/xsl/Transform">
<xsl:template match="/">
<html>
<body>
<h2>My CD Collection</h2>
<table border="1">
<tr bgcolor ="#9acd32">
<th>Title</th>
<th>Artist</th>
</tr>
<tr>
<td> </td>
<td> </td>
</tr>
</table>
```

```
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Output:

MY CD Collection

Title Artist

Since an XSL style sheet is an XML document, it always begins with the XML declaration: <?xml version = "1.0" encoding = "ISO-8859-1"?>

The next element, <xsl:stylesheet>,defines that this document is an XSLT style sheet document (along with the version number and XSLT namespace attributes).

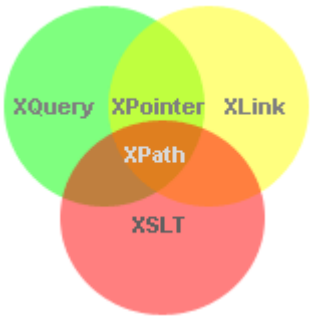
The <xsl:template> element defines a template. The match ="/" attribute associates the template with the root of the XMLsource document.

The content inside the <xsl:template>element defines some HTML to write to the output. The last two lines define the end of the template and the end of the stylesheet.

XML QUERY LANGUAGE

Q) EXPLAIN XML QUERY LANGUAGE

What is XQuery?



- XQuery is ***the*** language for querying XML data
- XQuery for XML is like SQL for databases
- XQuery is built on XPath expressions
- XQuery is supported by all major databases
- XQuery is a W3C Recommendation

XQuery is About Querying XML

XQuery is a language for finding and extracting elements and attributes from XML documents.

Here is an example of what XQuery could solve:

"Select all CD records with a price less than $10 from the CD collection stored in cd_catalog.xml"

XQuery and XPath

XQuery 1.0 and XPath 2.0 share the same data model and support the same functions and operators. If you have already studied XPath you will have no problems with understanding XQuery.

XQuery can be used to:

- Extract information to use in a Web Service
- Generate summary reports
- Transform XML data to XHTML
- Search Web documents for relevant information

XQuery Example

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

The XML Document (books.xml) is a below

We will use the following XML document in the examples below.

"books.xml":

```
</xml version="1.0"encoding="ISO-8859-1"?">
<bookstore>
<book category="COOKING">
   <title lang="an">Everyday ltalian</title>
    <author>Giada De Laurentiis</author>
     <year>2005</year>
    <price>29.99</price>
</book>

<book category="CHILDREN">
    <title lang="en">Harry potter</title>
    <author>jk.Rowling</author>
    <year>2005</year>
     <price>29.99</price>
</book>

<book category="WEB">
      <title lang="en">XQuery Kick Start</title>
      <author>Janes McGovern</author>
       <author>per Bothner</author>
       <author> kurt Cagle</author>
        <author>james Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <year>003</year>
        <price>49.99</price>
</book>
```

```
<book category=”WEB”>
      <title lang=”en”Learning XML</title>
       <author>Enk T.Ray</author>
         <year>2003</year>
         <price>39.95</price>
</book>
</bookstore>
```

IMP QUESTIONS

1) EXPLAIN XML STYLE SHEET.
2) COMPARE XML WITH OTHER WEB DESIGNING LANGUAGE.
3) EXPLAIN DOM IN XML.
4) WHAT ARE THE ELEMENTS AND ATTRIBUTES IN XML.
5) EXPLAIN THE RULES OF XML SYNTAX.
6) EXPLAIN DTDs in XML.
7) BRIEFLY EXPLAIN XML STYLESHEET.
8) BRIEFLY EXPLAIN XML QUERY LANGUAGE.

********** **END**********