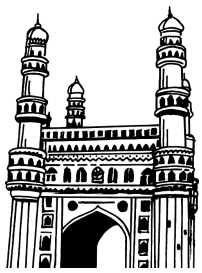
Rahul's ✔
Topper's Voice

AS PER
CBCS SYLLABUS



B.Com.

II Year IV Sem

(All Universities in Telangana)

Latest 2021 Edition

WEB TECHNOLOGIES

- ☞ FAQ's & Important Questions
- ☞ Study Manual
- ☞ Short Question & Answers
- ☞ Lab Programs
- ☞ Choose the Correct Answer
- ☞ Fill in the blanks
- ☞ One Mark Answers
- ☞ Solved Previous Question Papers
- ☞ Solved Model Papers

Price ₹. 169-00

- by -

WELL EXPERIENCED LECTURER

Rahul Publications™
Hyderabad. Ph : 66550071, 9391018098

B.Com.

II Year IV Sem

(All Universities in Telangana)

WEB TECHNOLOGIES

Inspite of many efforts taken to present this book without errors, some errors might have crept in. Therefore we do not take any legal responsibility for such errors and omissions. However, if they are brought to our notice, they will be corrected in the next edition.

Sole Distributors :

☎ : **66550071, Cell : 9391018098**

VASU BOOK CENTRE

Shop No. 3, Beside Gokul Chat, Koti, Hyderabad.

Maternity Hospital Opp. Lane, Narayan Naik Complex, Koti, Hyderabad.
Near Andhra Bank, Subway, Sultan Bazar, Koti, Hyderabad -195.

CONTENTS

WEB TECHNOLOGIES

STUDY MANUAL

SOLVED PREVIOUS QUESTION PAPERS

SOLVED MODEL PAPERS

SYLLABUS

UNIT - I

INTRODUCTION

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

UNIT - II

AN OVER VIEW OF DYNAMIC WEB PAGES & DYNAMIC WEB PAGE

An over view of dynamic web pages – technologies: Introduction to Dynamic HTML programing - Cascading style sheets (CSS) – types and advantages of CSS – CSS basic syntax and structure - Changing Text and Attributes - Dynamically changing style - Text Graphics and placements - Creating multimedia effects with filters and Transactions.

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 - Eventhandling

UNIT - IV

EVENTS AND EVENT HANDLERS

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 - Onsubmit - Onunload.

UNIT - V

EXTENSIBLE MARKUP LANGUAGE (XML)

Extensible Markup Language (XML): Introduction - Creating XML Documents - XML style Sheet – Hyperlinks in XML Document Object Model - XML Query Language.

LAB WORK: CREATING A WEBSITE WITH DYNAMIC FUNCTIONALITY USING CLIENT-SIDE AND SERVER SIDE SCRIPTING.

Contents

UNIT - I

UNIT - II

Topic **Page No.**

UNIT - IV

Topic **Page No.**

UNIT - V

Frequently Asked & Important Questions

UNIT - I

1. Why HTML is a Markup language ?

Ans : **(June-19, Imp.)**

Refer Unit-I, Q.No. 3.

---

2. Explain different types of HTML tags.

Ans : **(Imp.)**

Refer Unit-I, Q.No. 5.

---

3. Explain web designing principles.

Ans : **(Imp.)**

Refer Unit-I, Q.No. 7.

---

4. Explain the elements of forms?

Ans : **(June-19, Imp.)**

Refer Unit-I, Q.No. 15.

---

5. How can you insert an image in to your web document ?

Ans : **(Imp.)**

Refer Unit-I, Q.No. 19.

---

6. What is meant by List ? How we can create the list in html ?

Ans : **(Dec.-20, July-19,Dec.-18, Imp.)**

Refer Unit-I, Q.No. 22.

---

7. What is a table ? What is the use of table in HTML ?

Ans : **(June-19, Dec.-18)**

Refer Unit-I, Q.No. 23.

---

8. What is meant by Frames ? Explain dis-advantages of frames?

Ans : **(Dec.-18, Imp.)**

Refer Unit-I, Q.No. 28.

UNIT - II

1. What is DHTML ? What are the features of DHTML ?

Ans : **(Dec.-20, Dec.-18, Imp.)**

Refer Unit-II, Q.No. 1.

2. What is CSS? What are the advantages of CSS?

Ans : **(July-19, Dec.-18, Imp.)**

Refer Unit-II, Q.No. 5.

3. Explain the different types of CSS with an examples.

Ans : **(Dec.-20, July-19, Dec.-18, Imp.)**

Refer Unit-II, Q.No. 9.

4. Explain briefly dynamic changes to text, and graphics.

Ans : **(Imp.)**

Refer Unit-II, Q.No. 13.

5. Explain filters and transactions for multimedia effects.

Ans : **(Dec.-20, Dec.-18, June-19, Imp.)**

Refer Unit-II, Q.No. 14.

UNIT - III

1. What are the advantages and limitations of java script ?

Ans : **(Imp.)**

Refer Unit-III, Q.No. 2.

2. What are the data types supported by javascript?

Ans : **(June-19, Dec-18, Imp.)**

Refer Unit-III, Q.No. 9.

3. What is meant by operator ? how many types of operators supported by java script?

Ans : **(Dec.-20, June-19, Imp.)**

Refer Unit-III, Q.No. 12.

4. How java script supports conditional statements? Explain with example programs.

Ans : **(Imp.)**

Refer Unit-III, Q.No. 14.

5. How java script supports functions? Explain in detail?

Ans : **(Dec.-20, Dec.-18, Imp.)**

Refer Unit-III, Q.No. 17.

6. What is Document object model?

Ans : **(Imp.)**

Refer Unit-III, Q.No. 21.

UNIT - IV

1. What is an event? List out the events supported by javascript?

Ans : **(June-19, Imp.)**

Refer Unit-IV, Q.No. 3.

2. Explain in detail about Onerror event with suitable example.

Ans : **(June-19, Imp.)**

Refer Unit-IV, Q.No. 9.

3. Explain briefly about Onkey Press event?

Ans : **(Dec.-20, June-19, Dec.-18, Imp.)**

Refer Unit-IV, Q.No. 11.

4. Explain briefly about Onkey Up event?

Ans : **(Dec.-20, June-19, Dec.-18, Imp.)**

Refer Unit-IV, Q.No. 12.

5. Explain briefly about Onmouse Down event?

Ans : **(Dec.-20, Dec.-18, Imp.)**

Refer Unit-IV, Q.No. 14.

6. Explain briefly about Onmouse Move event?

Ans : **(Dec.-20, Dec.-18, Imp.)**

Refer Unit-IV, Q.No. 15.

7. Explain briefly about Onmouse Out event?

Ans : **(Dec.-20, Dec.-18, Imp.)**

Refer Unit-IV, Q.No. 16.

8. Explain about Onmouse Over event?

Ans : **(Dec.-20, Dec.-18)**

Refer Unit-IV, Q.No. 17.

9. Explain about On unloadevent?

Ans : **(Imp.)**

Refer Unit-IV, Q.No. 23.

UNIT - V

1. What are the features of XML?

Ans : **(Imp.)**

Refer Unit-V, Q.No. 3.

2. Distinguish between XML and HTML.

Ans : **(Imp.)**

Refer Unit-V, Q.No. 4.

3. Explain in detail about XML style sheets?

Ans : **(June-19, Imp.)**

Refer Unit-V, Q.No. 7.

4. Explain types of hyperlinks in xml? and Give a basic syntax ?

Ans : **(Imp.)**

Refer Unit-V, Q.No. 9.

5. Explain about DOM in XML ?

Ans : **(Dec.18, Imp.)**

Refer Unit-V, Q.No. 10.

6. What is XQuery?explain in detail ?

Ans : **(Dec.-20, June-19, Dec.18, Imp.)**

Refer Unit-V, Q.No. 11.

7. State the uses of XML Query.

Ans : **(Imp.)**

Refer Unit-V, Q.No. 12.

UNIT I

INTRODUCTION :

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

1.1 INTRODUCTION TO WEB TECHNOLOGY

Q1. What is web technology?

Ans :

Web Technologies is a programming concept that allows users to create interactive web pages. These pages can be created by writing code using various web programming languages like HTML, JAVAScript, and XML. Web programming is one of the easiest programming that rune directly on the computer.

A web document can be created using client side and server isde programming languages. In client side programming, the execution of program/ script is doen on client machine. Some of the client side programming languages are HTML. CSS, JavaScript etc. In server side programming, the execution of program/script is doen on server, some of the server side programming languages are PHP, and ASP etc.

1.2 HTML

Q2. What is HTML ?

Ans : **(Imp.)**

Meaning

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as “heading”, “paragraph”, “table”, and so on
- Browsers do not display the HTML tags, but use them to render the content of the page.

The definition of HTML is HyperText Markup Language.

- HyperText is the method by which you move around on the web - by clicking on special text called own which bring you to the next page. The fact that it is hyper just means it is not linear - i.e. you can go to any place on the Internet whenever you want by clicking on links - there is no set order to do things in.
- Markup is what HTML tags do to the text inside them. They mark it as a certain type of text.
- HTML is a Language, as it has code-words and syntax like any other language.

Q3. Why HTML is a Markup language ?

(OR)

HTML is a Markup language - Explain.

Ans : **(June-19, Imp.)**

Hypertext means machine readable text and Markup means to structure it in a specific format. So, HTML is called hypertext markup language because it is a language that allows users to organize, improve the appearance of, and link text with data on the internet

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user.

Q4. Explain different types of markup languages.

Ans :

There are three types of electronic markup language :

(i) **Presentational Markup:** Used by traditional word processing systems with WYSIWYG; it is hidden from human users.

(ii) **Procedural Markup:** Integrated with text to provide text processing instructions to programs. Such text is visibly manipulated by the author. Procedural markup systems include programming constructs, where macros or subroutines are defined and invoked by name.

(iii) **Descriptive Markup:** Used to label parts of a document as to how they should be treated. For example, the HTML <cite> tag is used to label citations in text.

Gencode was the first public markup language presentation in computer text processing. Some other major markup languages include:

- LaTex
- Extensible Markup Language (XML)
- Generalized Markup Language (GML)
- Standard Generalized Markup Language (SGML)
- HyperText Markup Language (HTML)

1.2.1 Types of HTML tags

Q5. Explain different types of HTML tags.

Ans : **(Imp.)**

Tags can be defined as the instructions which are being directly embedded in the text of an HTML document. The types of tags used in the HTML document are responsible to tell a web browser to do something (follow the instruction) instead of just displaying text. In an HTML document, all tag names are differentiated from other simple text. The tag names are enclosed in between angle brackets or a 'less than' and a 'greater than' symbol, (<) and (>).

1. Paired and Unpaired Tags

Following are the paired and unpaired tags in HTML explained in detail with the help of examples.

Paired Tags

An HTML tag is known as a paired tag when the tag consists of an opening tag and a closing tag as its companion tag. An HTML Paired tag starts with an opening tag: the tag name enclosed inside the angle brackets; for example, a paragraph opening tag is written as ' '. The content follows the opening tag, which ends with an ending tag: the tag name starting with a forward slash; for example, an ending paragraph tag is written as ' '. The first tag can be referred to as the 'Opening Tag', and the second tag can be called Closing Tag.

Example 1

 This text is a paragraph .

Output

Note: Here, the opening tag is, and the closing tag is .

Example 2

Another example of a paired tag is italic and/or bold tags:

 This is a bold and italicized text

Output

This is a bold and italicized text

Note: These paired tags are also called Container Tags.

Unpaired Tags

An HTML tag is called an unpaired tag when the tag only has an opening tag and does not have a closing tag or a companion tag. The Unpaired HTML tag does not require a closing tag; an opening tag is sufficient in this type. Unpaired tags are sometimes also named as Standalone Tags or Singular Tags since they do not require a companion tag.

Example

```
<p> This is a paragraph </p>
<hr>
<i><b> This is a bold and italicized text </b> </i>
```

Output

Note

Here, the <hr> is the unpaired tag used to create a horizontal line. In older versions, you might see hr tag written as <hr/> instead of <hr>. These tags are also called Empty Tag.

2. Self-Closing Tags

Self-Closing Tags are those HTML tags that do not have a partner tag, where the first tag is the only necessary tag that is valid for the formatting. The main and important information is contained WITHIN the element as its attribute. An image tag is the classic example of a self-closing tag. Let's see it in action below:

Example

```
<imgsrc="a.jpg" alt="This is an alternate text">
```

Note

In the older versions, the self-closing tags use a 'forward slash' before the ending or closing 'greater than' sign/symbol, as written below:

```
<imgsrc="a.jpg" alt="This is an alternate text" />
```

3. Utility-Based Tags

The HTML tags can be widely differentiated on the basis of their utility, that is, on the basis of the purpose they serve. We can divide them basically into three categories as discussed below:

i) Formatting Tags

The HTML tags that help us in the formatting of the texts like the size of the text, font styles, making a text bold, etc. This is done using tags like <font>, <b>, <u>, etc. Tables, divisions, and span tags are also those tags that help format a web page or document and set the layout of the page. Below is a small program using divisions for formatting the page along with some other formatting tags.

Example

```
<body>
<div class="container">
<div class="row">
<div class="col-25">
<label for="email"><b>Name</b></label>
</div>
<div class="col-35">
<input type="text" placeholder="First" name="fname" required>
</div>
<div class="col-35">
<input type="text" placeholder="Last" name = "lname" required>
</div>
</div>
</div>
</body>
```

Output

ii) Structure Tags

The HTML tags that help in structuring the HTML document are called Structure Tags. Description, head, html, title, body, etc., form the group of the page structure tags. The structure tags only assist in creating or forming the basic html page from the root; that is, they do not affect or has any hand in the formatting of texts. So a basic HTML program is the basic group of structural tags:

Example

```
<!doctype html>
```

```
<html>
<head>
<meta charset="utf-8">
<title>Types of Tags Demo</title>
</head>
<body>
<p> This is a paragraph </p>
<i><b> This is a bold and italicized text </b></i>
</body>
</html>
```

Output

This is a paragraph

This is abold and italicized text

iii) Control Tags

Another category of tags that can be created is 'Control Tags'. The Script tags, radio buttons or checkboxes, the Form tags, etc., forms the control tags. These are the tags that are used in managing content or managing scripts or libraries that are external. All the form tags, drop-down lists, input text boxes, etc., are used in interacting with the visitor or the user.

1.2.2 Basic Structure of HTML

Q6. Explain the basic Structure of HTML program.

Ans : **(Imp.)**

```
<html>
    <head>
        <title>Page title</title>
    </head>
    <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    </body>
</html>
```

Mainly we have 4 basic elements to develop any html program

1. <HTML>–</HTML>

This tag is used to initiate the html program. It has opening and closing tag. In between opening and closing mainly we should maintain 2sub tags called <head> and<body > tags.by using <html> we can able to create web documents /pages.

Syntax :

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

2. <head>–</head>

This tag is used to maintain header section for the web documents or webpages. This tag can capable to declare many styles and rules for the document . this tag supports

syntax:

```
<head>
</head>
```

3. <title>–</title>

This tag is used to give a specific name for the web document

And which is displayed top left of the document. We should remember that always title tag should under head section only.

Syntax:

```
<head>
    <title>
    </title>
</head>
```

4. <body>–</body>

This tag is used to write actual body of the program. And which display the elements up on the web page which have been declared in between opening and closing body tags

Always we should remember that body tag has to declare after head tag only

Syntax

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

1.3 WEB DESIGN PRINCIPLES

Q7. Explain web designing principles.

(OR)

List out various web design principles.

Ans : **(Imp.)**

1. Purpose

Good web design always caters to the needs of the user. Are your web visitors looking for information, entertainment, some type of interaction, or to transact with your business? Each page of your website needs to have a clear purpose, and to fulfill a specific need for your website users in the most effective way possible.

2. Communication

People on the web tend to want information quickly, so it is important to communicate clearly, and make your information easy to read and digest. Some effective tactics to include in your web design include: organising information using headlines and sub headlines, using bullet points instead of long windy sentences, and cutting the waffle.

3. Typefaces

In general, Sans Serif fonts such as Arial and Verdana are easier to read online (Sans Serif fonts are contemporary looking fonts without decorative finishes). The ideal font size for reading easily online is 16px and stick to a maximum of 3 typefaces in a maximum of 3 point sizes to keep your design streamlined.

4. Colours

A well thought out colour palette can go a long way to enhance the user experience. Complementary colours create balance and harmony. Using contrasting colours for the text and background will make reading easier on the eye. Vibrant colours create emotion and should be used sparingly (e.g. for buttons and call to actions). Last but not least, white space/ negative space is very effective at giving your website a modern and uncluttered look.

5. Images

A picture can speak a thousand words, and choosing the right images for your website can help with brand positioning and connecting with your target audience. If you don't have high quality professional photos on hand, consider purchasing stock photos to lift the look of your website. Also consider using infographics, videos and graphics as these can be much more effective at communicating than even the most well written piece of text.

6. Navigation

Navigation is about how easy it is for people to take action ands move around your website. Some tactics for effective navigation include a logical page hierarchy, using bread crumbs, designing clickable buttons, and following the 'three click rule' which means users will be able to find the information they are looking for within three clicks.

7. Grid based layouts

Placing content randomly on your web page can end up with a haphazard appearance that is messy. Grid based layouts arrange content into sections, columns and boxes that line up and feel balanced, which leads to a better looking website design.

8. "F" Pattern design

Eye tracking studies have identified that people scan computer screens in an "F" pattern. Most of what people see is in the top and left of the screen and the right side of the screen is rarely seen. Rather than trying to force the viewer's visual flow, effectively designed websites will work with a reader's natural behaviour and display information in order of importance (left to right, and top to bottom).

9. Load time

Everybody hates a website that takes ages to load. Tips to make page load times more effective include optimizing image sizes (size and scale), combining code into a central CSS or JavaScript file (this reduces HTTP requests) and minify HTML, CSS, JavaScript (compressed to speed up their load time).

10. Mobile friendly

It is now commonplace to access websites from multiple devices with multiple screen sizes, so it is important to consider if your website is mobile friendly. If your website is not mobile friendly, you can either rebuild it in a responsive layout (this means your website will adjust to different screen widths) or you can build a dedicated mobile site (a separate website optimised specifically for mobile users).

1.4 HTML ATTRIBUTES

Q8. Explain various types of HTML Attributes.

Ans : **(Imp.)**

HTML attributes can be said as special words which are used inside opening tags and are responsible for controlling an element's behavior. They are a modifier of HTML element type. This modifier can do two jobs. It can either modify the default functionality of an element or provide the functionality to any element that cannot function properly without them. Syntactically, an attribute is added to the HTML start tag. There are different types of attributes that have been identified. These include the required attributes, optional attributes, standard attributes, and event attributes.

Different HTML Attributes

Let us have a look at different attributes and how they work in detail.

1. Core Attributes

There are four core attributes that are majorly used. To name these, they are as follows:

- **ID:** This HTML attribute can be easily used to identify an element present within an HTML page uniquely. The user id can be used either when an element carries an id attribute as a unique identifier, making it possible to identify the element and its content or this is used when there are two elements with the same name within a web page. The ID attribute can help in easily identifying the difference between the elements which have the same name.
- **The Title Attribute:** This attribute does the work of giving a suggested title for a particular element. It depends on the carrier how it will behave, and although it is not usually displayed as a tooltip whenever the cursor comes over the element or if it has to display it when the element is loading. It is also used to explain an element on hovering the mouse over it. The behavior can differ with different elements, and generally, its value is displayed on loading or on hovering the mouse pointer over it.
- **The Class Attribute:** The association of this attribute is done with an element of the style sheet. The user needs to specify the class of the element. More can be learned about this attribute when Cascading Style sheet is being learned. The value here may also be a space-separated list of class names. For example: class = "className1 className2 className3"
- **The Style Attribute:** This attribute allows you to specify the Cascading Style Sheet rules within any element. It can provide various Cascading Style sheet effects to the HTML elements like increasing the font size. It can also change the font family and coloring

After the core attributes, we have some internationalization attributes. They are as below:

2. Internationalization Attribute

- **Dir:** The dir attribute helps you in indicating to the browser the direction in which a text should follow. This attribute can usually take up two values. These can be LTR and RTL. LTR means left to right, and this is the default value, while RTL stands for the right to left. When this attribute is used within the <html> tag, it determines how a text should be

represented within the entire document. It can also be used to control text's direction from just the content of the tag.

- **The Lang Attribute:** This attribute helps in showcasing the main language that is used in a document. This attribute can be kept in HTML so that it is backward compatible with earlier versions of HTML. It can also be replaced by the XML: lang attribute in new XHTML documents. The values of lang attributes are ISO-639 standard and have two-character language codes. Declaring a language is important in order to access the application and different search engines.
- **The XML-Lang Attribute:** This attribute is supposed to be the replacement for the lang attribute. The value of the XML-lang attribute must have the country code as mentioned previously.

In addition to these, there are many generic attributes that are described below.

3. Generic Attribute

- **Align Attribute:** This attribute is useful when you wish to have some elements of your page at your disposition. You can change the alignment to left, right or the center of the page. The default alignment for all elements is set to the left. This can be changed by using this align attribute.
- **Src Attribute:** If a user needs to insert an image into a web page, we need to use the <img> tag with the src attribute. We can specify the address of the image as the attribute's value inside the double quote. You can use the src attribute as below to include the image on the webpage.
- **Alt Attribute:** This attribute is used as an alternate tag that can be used to show something if the primary attribute that is the <img> tag which fails to display the original value which is assigned to it. This can describe the image to a developer who is using it at the coding end. If the main image fails, then the alternate image can be used to display.
- **The Width and Height Attribute:** This attribute can be used to adjust the height and width of an image.
- **The Href Attribute:** This attribute is used when the user wants to direct to a specific link to any address. This attribute is used along with the <a> tag. When the link is placed in the href attribute where it is to be directed, it is linked to the text displayed within the <a> tag.

4. Global Attributes

HTML also provides global attributes which can be used with any HTML element. The attributes are as below.

- **Accesskey:** It specifies a shortcut key to activate or focus on any element.
- **Translate:** If this attribute is used, it specifies whether the content of the element is to be translated or not.
- **Class:** It specifies one or more class names for an element.
- **Title:** This attribute specifies extra information about an element.
- **Contenteditable:** In order to specify whether the content is editable or not, this attribute can be used.
- **Tabindex:** It specifies the tabbing order of an element.
- **Dir:** It specifies the text direction for any content of an element.
- **Spellcheck:** User can explicitly specify if they have to check the spelling and grammar checked or not.
- **Draggable:** It specifies if an element should be draggable or not.
- **Dropzone:** It specifies whether the dragged data is copied, moved or linked when dropped.

5. Event Attributes

HTML has the ability to trigger actions when some events take place. The event attributes can be as below.

- **Onload:** It fires after the page has finished loading.
- **Onmessage:** This can be said as a script that runs when the message is triggered.

- **Onstorage:** This is a script to be run when a web storage area is updated.
- **Onerror:** This script is run when an error occurs.
- **Onpagehide:** This script can be used when a user can navigate away from a page.

6. Form Event Attributes

Actions inside an HTML form trigger these events.

- **Onblur:** It is triggered as soon as the element loses focus.
- **Onchange:** It is triggered as soon as the value of an element changes.
- **Oncontextmenu:** This is run when a context menu is triggered.
- **Onfocus:** It is triggered as soon as the element gets focus.
- **Oninput:** The script has to run when the element receives an input.
- **Onsearch:** This is triggered when the user writes something in the search field.
- **Oninvalid:** This is triggered when an element that is entered is invalid.

7. Key Event Attributes

- **Onkeydown:** It is triggered when a key is being pressed.
- **Onkeypress:** This is triggered when a key is pressed.
- **Onkeyup:** This is triggered when a user releases a key.

8. Mouse Event Attributes

- **Onclick:** This is triggered when the mouse clicks on an element.
- **Onmousemove:** This is fired when the mouse pointer is moving while it is over an element.
- **Onmouseip:** It is triggered when a mouse button is released from over an element.
- **Onwheel:** It is triggered when the mouse wheel rolls up or down over an element

9. Drag Event Attributes

- **Ondrag:** This is triggered when an element is dragged.
- **Ondragleave:** The script is run when an element leaves a valid drop target.
- **Ondrop:** It is triggered when the dragged element is being dropped
- **Onscroll:** The script is run when an element's scroll bar is being scrolled.

1.5 HTML STYLES

Q9. Explain various types of HTML Styles.

Ans : **(Imp.)**

He HTML style attribute is used to add styles to an element, such as color, font, size, and more.

The HTML Style Attribute

Setting the style of an HTML element, can be done with the style attribute.

The HTML style attribute has the following syntax:

<tagname style="property:value;">

The ***property*** is a CSS property. The ***value*** is a CSS value.

1. Background Color

The CSS background-color property defines the background color for an HTML element.

Example

Set the background color for a page to powderblue:

```
<body style="background-color:powderblue;">
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

2. Text Color

The CSS color property defines the text color for an HTML element:

Example

<h1 style="color:blue;">This is a heading</h1>

This is a paragraph.

3. Fonts

The CSS font-family property defines the font to be used for an HTML element:

Example

<h1 style="font-family:verdana;">This is a heading</h1>

This is a paragraph.

4. Text Alignment

The CSS text-align property defines the horizontal text alignment for an HTML element:

Example

<h1 style="text-align:center;">Centered Heading</h1>

Centered paragraph.

1.6 HYPERTEXT

Q10. What is Hyper Text ?

Ans :

The term "hypertext" identifies an electronic document containing several parts that the electronically linked together unlike the traditional linear media. For example, bound books.

The term "hypermedia" identifies an hypertext that contains additional text, graphics, sound and animation. The user can use both hypertext and hypermedia to access information in the desired manner irrespective of database structure or computer program limitations. A hypertext screen allows users to click on any of its highlighted text, maps or photographs to obtain their corresponding details. These clicks open different screens which include more hypertext links to other screens as a chain. The chain continues until all the available screens have been reached.

1.7 FORMATTING TEXT

Q11. How can you format the text in HTML ?

Ans :

(i) HTML Formatting Elements

In the previous chapter, you learned about the HTML style attribute.

HTML also defines special elements for defining text with a special meaning.

HTML uses elements like and for formatting output, like bold or italic text.

Formatting elements were designed to display special types of text :

- - Bold text
- - Important text
- - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Small text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text.

**1. and **

The HTML element defines **bold** text, without any extra importance.

Example

This text is bold

The HTML element defines **strong** text, with added semantic "strong" importance.

Example

This text is strong

**2. and **

The HTML element defines italic text, without any extra importance.

Example

This text is italic

The HTML element defines emphasized text, with added semantic importance.

Example

This text is emphasized

3. <small>

The HTML <small> element defines smaller text:

Example

<h2>HTML <small>Small</small> Formatting</h2>

4. <mark>

The HTML <mark> element defines marked or highlighted text:

Example

<h2>HTML <mark>Marked</mark> Formatting</h2>

5. <del>

The HTML <del> element defines ~~deleted~~ (removed) text.

Example

My favorite color is ~~blue~~ red.

6. <ins>

The HTML <ins> element defines inserted (added) text.

Example

My favorite <ins>color</ins> is red.

7. <sub>

The HTML <sub> element defines subscripted text.

Example

This is _{subscripted} text.

8. <sup>

The HTML <sup> element defines superscripted text.

Example

This is ^{superscripted} text.

(ii) HTML Text Formatting Elements

Tag	Description
	Defines bold text
	Defines emphasized text
	Defines italic text
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<mark>	Defines marked/highlighted text

Q12. List and explain some Text Formatting Tags.

Ans :

The following HTML tags are used to format the appearance of the text on your web page.

(i) Header - <h?></h?>

There are 6 levels of headings available, from h1 for the largest and most important heading, down to h6 for the smallest heading.

**(ii) Bold - **

The text in between the tags will be bold, and stand out against text around it, the same as in a word processor.

**(iii) Italic - **

Also working the same way as a word processor, italics displays the text at a slight angle.

(iv) Underline - <u></u>

Again, the same as underline in a word processor. Note that html links are already underlined and don't need the extra tag.

(v) Strike-out - <strike></strike>

Puts a line right through the centre of the text, crossing it out.Often used to show that text is old and no longer relevant. Also works by using <s></s> instead.

(vi) Preformatted Text - <pre></pre>

Any text between the pre tags, including spaces, carriage returns and punctuation, will appear in the browser as it would in a text editor (normally browsers ignore multiple spaces)

(vii) Source Code - <code></code>

Text is displayed in a fixed-width font, commonly used when showing source code. I have used it on this site, along with stylesheets, to show all tags.

(viii) Typewriter Text - <tt></tt>

The text appears to have been typed by a typewriter, in a fixed-width font.

(ix) Block Quote - <blockquote></blockquote>

Defines a long quotation, and the quote is displayed with an extra wide margin on the left hand side of the block quote.

(x) Small - <small></small>

Instead of having to set a font size, you can use the small tag to render text slightly smaller than the text around it. Useful for displaying the 'fine-print'.

**(xi) Font Colour - **

Change the colour of a few words or a section of text. The 6 question marks represent the hex color code,

**(xii) Font Size - **

Replace the ?with a number from 1 to 7 to change the size of the font. One being the smallest and seven the largest.

**(xiii) Font Size Change - **

For an immediate change of font size with respect to the font size preceding it, this tag increase or decreases the size of the font by the number you specify. Eg: Some Text

**(xiv) Change Font Face - **

To show text in a particular font, use the font name such "Helvetica" or "Arial" or "Courier". Be aware that using some fancy font from your computer means that the person viewing that page must also have that font installed on their computer too, otherwise it will look totally different to them.

**(xv) Centre - **

A useful tag, as it says, it makes everything in between the tags centred (in the middle of the page).

**(xvi) Emphasis - **

Used to emphasize text, which usually appears in italics, but can vary according to your browser.

**(xvii) Strong Emphasis - **

Used to emphasize text more, which usually appears in bold, but can vary according to your browser.

Text Formatting

<h?> ... </h?>	Heading (?= 1 for largest to 6 for smallest, eg h1)
...	Bold Text
...	Italic Text
<u> ... </u>	Underline Text
<strike> ... </strike>	Strikeout
^{...}	Superscript - Smaller text placed below normal text
_{...}	Subscript - Smaller text placed below normal text
...	Small - Fineprint size text
<tt> ... </tt>	Typewriter Text
<pre> ... </pre>	Pre-formatted Text
<blockquote> ... </blockquote>	Text Block Quote
...	Strong - Shown as Bold in most browsers
...	Emphasis - Shown as Italics in most browsers
 ... 	Font tag obsolete,

1.8 FORMS

Q13. What is meant by Form? How can you create a Form in html ?

Ans : **(June-19, Imp.)**

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like

- text fields,
- text area fields,
- drop-down menus,
- radio buttons,
- checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following

syntax :

<form action = "Script URL" method = "GET|POST">

form elements like input, textarea etc.

</form>

1.8.1 Forms formulating instructions and Formulation Elements

Q14. Explain attributes of form tag ?

Ans :

Apart from common attributes, following is a list of the most frequently used form attributes -

1. **action.** Backend script ready to process your passed data.
2. **method.** Method to be used to upload data. The most frequently used are GET and POST methods.
3. **target.** Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
4. **enctype.** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server.

Q15. Explain the elements of forms?

(OR)

Explain form control methods.

Ans : **(June-19, Imp.)**

HTML Form Controls

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

1. Text Input Controls
2. Checkboxes Controls
3. Radio Box Controls
4. Select Box Controls
5. File Upload Control
6. Button Controls
7. Hidden form Controls

1. Text Input Controls

There are three types of text input used on forms -

(i) Single-line text input controls " This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name -

```
<html>

<head>
<title>Text Input Control</title>
</head>

<body>
```

```
<form>
    First name: <inputtype="text"    name="first_name"/>
<br>
    Last name: <inputtype="text"    name="last_name"/>
</form>
</body>

</html>
```

This will produce the following result -

Attributes

Following is the list of attributes for <input> tag for creating text field.

Attribute	Description
type	Indicates the types of input control and for text output control it will be se to text.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	This can be used to provide an initial value inside the control.
size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.

(ii) Password input controls " This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl<input> tag.

(iii) Multi-line text input controls " This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description -

```
<html>

<head>
<title>Multiple-Line Input Control</title>
</head>

<body>
<form>
    Description : <br/>
<textarearows="5"    cols="50"    name="description">
```

```
        Enter description here...
</textarea>
</form>
</body>

</html>
```

Attributes

Following is the list of attributes for <textarea> tag.

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognised and get the value.
rows	Indicates the number of rows of text area box.
cols	Indicates the number of columns of text area box.

2. Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox.**.

Example

Here is an example HTML code for a form with two checkboxes

```
<html>
<head>
<title>Checkbox Control</title>
</head>
<body>
<form>
<inputtype="checkbox" name="maths" value="on">Maths
<inputtype="checkbox" name="physics" value="on"> Physics
</form>
</body>
</html>
```

This will produce the following result -

Attributes

Following is the list of attributes for <checkbox> tag.

Attribute	Description
type	Indicates the types of input control and for checkbox input control it will be set to checkbox.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to checked if you want to select it by default.

3. Radio Box Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

Example

```
<html>

<head>
<title>Radio Box Control</title>
</head>

<body>
<form>
<inputtype="radio" name="subject" value="maths">Maths
<inputtype="radio" name="subject" value="physics"> Physics
</form>
</body>
</html>
```

This will produce the following result -

Attributes

Following is the list of attributes for radio button.

Attribute	Description
type	Indicates the types of input control and for checkbox input control it will be set to radio.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the radio box is selected.
checked	Set to checked if you want to select it by default.

4. Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

```
<html>
<head>
<title>Select Box Control</title>
</head>
    <body>
<form>
<selectname="dropdown">
<optionvalue="Maths"selected>Maths</option>
<optionvalue="Physics">Physics</option>
</select>
</form>
</body>
</html>
```

Attributes

Following is the list of important attributes of <select> tag -

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
size	This can be used to present a scrolling list box.
multiple	If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag -

Attribute	Description
value	The value that will be used if an option in the select box, box is selected.
selected	Specifies that this option should be the initially selected value when the page loads.
label	An alternative way of labeling options.

5. File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to **file**.

</body>

</html>

Attributes

Following is the list of important attributes of file upload box -

Attribute	Description
name	Used to give a name to the control which is sent to the server to be recognised and get the value.
accept	Specifies the types of files that the server accepts.

6. Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to **button**. The type attribute can take the following values.

Type	Description
submit	This creates a button that automatically submits a form.
reset	This creates a button that automatically resets form controls to their initial values.
button	This creates a button that is used to trigger a client-side script when the user clicks that button.
image	This creates a clickable button but we can use an image as background of the button.

Example

```
<html>

<head>
<title>File Upload Box</title>
</head>

<body>
<form>
<inputtype="submit" name="submit" value="Submit"/>
<inputtype="reset" name="reset" value="Reset"/>
```

```
<inputtype="button" name="ok" value="OK"/>
<inputtype="image" name="imagebutton" src="/html/images/logo.png"/>
</form>
</body>
</html>
```

7. Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

Example

```
<html>
<head>
<title>File Upload Box</title>
</head>
<body>
<form>
<p>This is page 10</p>
<inputtype="hidden" name="pagename" value="10"/>
<inputtype="submit" name="submit" value="Submit"/>
<inputtype="reset" name="reset" value="Reset"/>
</form>
</body>
</html>
```


1.9 COMMENTING CODE

Q16. What is the use of comment tags in html ?

Ans :

The **comment tag** is used to insert **comments** in the source code. **Comments** are not displayed in the browsers. You can **use comments** to explain your code, which can help you when you edit the source code at a later date. This is especially useful if you have a lot of code.

An HTML comment :

```
<!—This is a comment. Comments are not displayed in the browser—>
```

This element is used to add a comment to an HTML document. An HTML comment begins with <!— and the comment closes with —> . HTML comments are **visible** to anyone that views the page source code, but are not rendered when the HTML document is rendered by a browser.

The comment tag is used to **insert** comments in the source code. Comments are not displayed in the browsers. You can use comments to explain your code, which can help you when you **edit** the source code at a later date.

1.10 BACK GROUNDS

Q17. What is backgrouds? How can you set background for a html document ?

Ans :

By default, your webpage background is white in color. You may not like it, but no worries. HTML provides you following two good ways to decorate your webpage background.

- HTML Background with Colors
- HTML Background with Images

Html Background with Colors

The **bgcolor** attribute is used to control the background of an HTML element, specifically page body and table backgrounds.

Note : The *bgcolor* attribute deprecated in HTML5. Do not use this attribute.

Following is the syntax to use bgcolor attribute with any HTML tag.

```
<tagnamebgcolor = "color_value"...>
```

This color_value can be given in any of the following formats -

```
<!— Format 1 - Use color name —>
<table bgcolor = "lime" >

<!— Format 2 - Use hex value —>
<table bgcolor = "#f1f1f1" >

<!— Format 3 - Use color value in RGB terms —>
<table bgcolor = "rgb(0,0,120)" >
```

Example

```
<html>

<head>
<title>HTML Background Colors</title>
</head>

<body>
<!— Format 1 - Use color name —>
<tablebgcolor="yellow"  width="100%">
<tr>
<td>
        This background is yellow
</td>
</tr>
</table>

<!— Format 2 - Use hex value —>
```

```
<tablebgcolor=”#6666FF” width=”100%”>
<tr>
<td>
This background is sky blue
</td>
</tr>
</table>

<!— Format 3 - Use color value in RGB terms —>
<tablebgcolor=”rgb(255,0,255)” width=”100%”>
<tr>
<td>
This background is green
</td>
</tr>
</table>
</body>

</html>
```

The **background** attribute can also be used to control the background of an HTML element, specifically page body and table backgrounds. You can specify an image to set background of your HTML page or table.

Note - The *background* attribute deprecated in HTML5. Do not use this attribute.

Following is the syntax to use background attribute with any HTML tag.

Note - The *background* attribute is deprecated and it is recommended to use Style Sheet for background setting.

```
<tagname background = “Image URL”...>
```

The most frequently used image formats are JPEG, GIF and PNG images.

Example

Here are the examples to set background images of a table.

```
<html>

<head>
<title>HTML Background Images</title>
</head>

<body>
<!— Set table background —>
<tablebackground=”/images/html.gif” width=”100%” height=”100">
<tr><td>
```

This background is filled up with HTML image.

</td></tr>

</table>

</body>

</html>

Q18. Explain the Patterned & Transparent Backgrounds.

Ans :

You might have seen many pattern or transparent backgrounds on various websites. This simply can be achieved by using patterned image or transparent image in the background.

It is suggested that while creating patterns or transparent GIF or PNG images, use the smallest dimensions possible even as small as 1x1 to avoid slow loading.

Example

Here are the examples to set background pattern of a table -

<html>

<head>

<title>HTML Background Images</title>

</head>

<body>

<!— Set a table background using pattern —>

<tablebackground="/images/pattern1.gif" width="100%" height="100">

<tr>

<td>

This background is filled up with a pattern image.

</td>

</tr>

</table>

<!— Another example on table background using pattern —>

<tablebackground="/images/pattern2.gif" width="100%" height="100">

<tr>

<td>

This background is filled up with a pattern image.

</td>

</tr>

</table>

</body>

</html>

1.11 IMAGES

Q19. How can you insert an image in to your web document ?

Ans : **(Imp.)**

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

Insert Image

You can insert any image in your web page by using **<img>** tag. Following is the simple syntax to use this tag.

<imgsrc = "Image URL" ... attributes-list/>

The <img> tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

Example

To try following example, let's keep our HTML file test.htm and image file test.png in the same directory -

```
<html>
<head>
<title>Using Image in Webpage</title>
</head>

<body>
<p>Simple Image Insert</p>
<imgsrc="/html/images/test.png" alt="Test Image"/>
</body>

</html>
```

You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive.

The **alt** attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

Q20. Explain the properties (or) attributes of an image.

Ans :

Following are the basic properties of images.

1. Set Image Location

Usually we keep all the images in a separate directory. So let's keep HTML file test.htm in our home directory and create a subdirectory **images** inside the home directory where we will keep our image test.png.

Example

Assuming our image location is "image/test.png", try the following example -

```
<html>
```

```
<head>
<title>Using Image in Webpage</title>
</head>
<body>
<p>Simple Image Insert</p>
<imgsrc="/html/images/test.png" alt="Test Image"/>
</body>
</html>
```

2. Set Image Width/Height

You can set image width and height based on your requirement using **width** and **height** attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

```
<html>
<head>
<title>Set Image Width and Height</title>
</head>
<body>
<p>Setting image width and height</p>
<imgsrc="/html/images/test.png" alt="Test Image" width="150" height="100"/>
</body>

</html>
```

3. Set Image Border

By default, image will have a border around it, you can specify border thickness in terms of pixels using border attribute. A thickness of 0 means, no border around the picture.

```
<html>
<head>
<title>Set Image Border</title>
</head>
<body>
<p>Setting image Border</p>
<imgsrc="/html/images/test.png" alt="Test Image" border="3"/>
</body>
</html>
```

4. Set Image Alignment

By default, image will align at the left side of the page, but you can use **align** attribute to set it in the center or right.

```
<html>

<head>
<title>Set Image Alignment</title>
</head>
```

```
<body>
<p>Setting image Alignment</p>
<imgsrc="/html/images/test.png" alt="Test Image" border="3" align="right"/>
</body></html>
```

1.12 HYPERLINKS

Q21. What is the use of anchor tag in html ?

(OR)

What is hyperlink? How can you link a page with another page.

Ans : **(June-19, Dec.-18, Imp.)**

An anchor tag is an HTML tag. It is used to define the beginning and end of a hypertext link. It contains visible words within a text that can be clicked and the URL of the link's target.

The HTML code for creating a link to another page or to a particular section within a page. It is also commonly called an "h-ref." See HREF.

Links are specified in HTML using the <a>tag.

The <a> can be used in 2 ways

1. To create a link to another document, by using href attribute
2. To create a bookmark inside a document, by using name attribute.

Syntax:

```
<a href="url">link text</a>
```

The href attribute specifies the destination of a link.

Eg :

```
<a href="http//:www.w3cschools.com">visit w3c schools</a>
```

By click on above link current page is directed to w3cschools page. This kind of lining between pages done by using <a>only. This mechanism is also called as "hyperlink"

Sample program:

```
<html>
<head>
<title>creating hyperlink</title>
</head>
<body>
<center>
<h1>creating hyperlink</h1>
Here is a website to checkout
<a href=http://w3c.org>click here for w3cschools</a>
</center>
</body>
</html>
```

1.13 LISTS

Q22. What is meant by List ? How we can create the list in html ?

(OR)

Define Listing. Explain types of list with sample html code.

Ans : **(Dec.-20, July-19,Dec.-18, Imp.)**

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain -

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

(i) Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML **<ul>** tag. Each item in the list is marked with a bullet.

```
<html>
<head>
<title>HTML Unordered List</title>
</head>
<body>
<ul>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</body>
</html>
```

Output

- Beetroot
- Ginger
- Potato
- Radish

The type Attribute

You can use **type** attribute for <ul> tag to specify the type of bullet you like. By default, it is a disc. Following are the possible options "

```
<ul type = "square">
<ul type = "disc">
<ul type = "circle">
```

(ii) Ordered Lists

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using **<ol>** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.

```
<html>

<head>
<title>HTML Ordered List</title>
</head>

<body>
<ol>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

Output:

1. Beetroot
2. Ginger
3. Potato
4. Radish

The type Attribute

You can use **type** attribute for <ol> tag to specify the type of numbering you like. By default, it is a number. Following are the possible options "

```
<ol type = "1"> - Default-Case Numerals.
<ol type = "I"> - Upper-Case Numerals.
<ol type = "i"> - Lower-Case Numerals.
<ol type = "A"> - Upper-Case Letters.
<ol type = "a"> - Lower-Case Letters.
```

(iii) Definition Lists

HTML and XHTML supports a list style which is called **definition lists** where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition

List makes use of following three tags.

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

Example

```
<!DOCTYPE html>
<html>

<head>
<title>HTML Definition List</title>
</head>

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

</html>
```

Output

HTML : This stands for Hyper Text Markup Language

HTTP : This stands for Hyper Text Transfer Protocol

1.14 TABLES

Q23. What is a table ? What is the use of table in HTML ?

(OR)

Explain in detail about tables in html and its associated tags.

Ans : **(June-19, Dec.-18)**

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. The elements under <td> are regular and left aligned by default

Example

```
<html>
<head>
<title>HTML Tables</title>
</head>
<body>
<tableborder="1">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

Here, the **border** is an attribute of <table> tag and it is used to put a border across all the cells. If you do not need a border, then you can use border = "0".

Q24. Explain the properties (or) attributes of <table> in html ?

Ans : **(June-19, Dec.-18)**

1. Table Heading

Table heading can be defined using **<th>** tag. This tag will be put to replace <td> tag, which is used to represent actual data cell.

Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row. Headings, which are defined in <th>tag are centered and bold by default.

Example

```
<html>

<head>
<title>HTML Table Header</title>
</head>

<body>
<tableborder="1">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>

<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>

</html>
```

2. Tables Backgrounds

You can set table background using one of the following two ways -

- **bgcolor** attribute "You can set background color for whole table or just for one cell.
- **background** attribute "You can set background image for whole table or just for one cell.

You can also set border color also using **bordercolor** attribute.

Example

```
<html>

<head>
<title>HTML Table Background</title>
</head>

<body>
<tableborder="1" bordercolor="green"
bgcolor="yellow">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr>
<tdrowspan="2">Row 1 Cell 1</td>
<td>Row 1 Cell 2</td>
<td>Row 1 Cell 3</td>
</tr>
<tr>
<td>Row 2 Cell 2</td>
<td>Row 2 Cell 3</td>
</tr>
<tr>
<tdcolspan="3">Row 3 Cell 1</td>
</tr>
</table>
</body>

</html>
```

3. Table Caption

The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

Example

```
<html>

<head>
<title>HTML Table Caption</title>
</head>

<body>
<tableborder="1" width="100%">
<caption>This is the caption</caption>

<tr>
<td>row 1, column 1</td><td>row 1,
columnn 2</td>
```

```
</tr>

<tr>
<td>row 2, column 1</td><td>row 2, columnn 2</td>
</tr>
</table>
</body>

</html>
```

This is the caption	
row 1, column 1	row 1, column 2
row 2, column 1	row 2, column 2

4. Table Header, Body, and Footer

Tables can be divided into three portions "a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are -

- **<thead>** - to create a separate table header.
- **<tbody>** - to indicate the main body of the table.
- **<tfoot>** - to create a separate table footer.

A table may contain several <tbody> elements to indicate *different pages* or groups of data. But it is notable that <thead> and <tfoot> tags should appear before <tbody>

Example

```
<html>

<head>
<title>HTML Table</title>
</head>

<body>
<tableborder="1" width="100%">
<thead>
<tr>
<tdcolspan="4">This is the head of the table</
td>
</tr>
</thead>

<tfoot>
<tr>
<tdcolspan="4">This is the foot of the table</
td>
</tr>
</tfoot>

<tbody>
<tr>
<td>Cell 1</td>
<td>Cell 2</td>
<td>Cell 3</td>
<td>Cell 4</td>
</tr>
</tbody>

</table>
</body>

</html>
```

This will produce the following result -

This is the head of the table
This is the foot of the table

Q25. What is mean by Nested tables? How can you create ?

Ans : **(Imp.)**

You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

Example

Following is the example of using another table and other tags inside a table cell.

```
<html>

<head>
<title>HTML Table</title>
</head>

<body>
```

```
<tableborder = "1"width = "100%" >
<tr >
<td >
<tableborder = "1"width = "100%" >
<tr >
<th > Name </th >
<th > Salary </th >
</tr >
<tr >
<td > Ramesh Raman </td >
<td > 5000 </td >
</tr >
<tr >
<td > Shabbir Hussein </td >
<td > 7000 </td >
</tr >
</table >
</td >
</tr >
</table >
</body >
</html >
```

This will produce the following result -

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

Q26. Explain the following terms related to table.

(a) cell padding and cell spacing attributes

(b) colspan and rowspan attributes.

Ans :

(a) Cellpadding and Cellspacing Attributes

There are two attributes called cellpadding and cellspacing which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

Example

```
<html >
<head >
<title > HTML Table Cellpadding </title >
</head >
<body >
<tableborder = "1" cellpadding = "5" cellspacing = "5" >
<tr >
<th > Name </th >
<th > Salary </th >
</tr >
<tr >
<td > Ramesh Raman </td >
<td > 5000 </td >
</tr >
<tr >
<td > Shabbir Hussein </td >
<td > 7000 </td >
</tr >
</table >
</body >
</html >
```

(b) Colspan and Rowspan Attributes

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

Example

```
<html >
<head >
<title > HTML Table Colspan/Rowspan </title >
</head >
<body >
<tableborder = "1" >
<tr >
<th > Column 1 </th >
```

```
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr>
<tdrowspan="2">Row 1 Cell 1</td>
<td>Row 1 Cell 2</td>
<td>Row 1 Cell 3</td>
</tr>
<tr>
<td>Row 2 Cell 2</td>
<td>Row 2 Cell 3</td>
</tr>
<tr>
<tdcolspan="3">Row 3 Cell 1</td>
</tr>
</table>
</body>
</html>
```

Q27. Write a program to demonstrate rowspan and colspan attributes in table shown below.

Name	Marks		
	FM	FIT	BOM
Rahul	68	68	75
Rakesh	60	69	60
Ravi	68	66	60

Ans :

Program

```
<html>
<head>
<title> Tables </title>
<body bgcolour = “pink” align = “centre”?
<table border = “1” width = “50%”>
<col width = “75”>
<col width = “25”>
<tr>
<th rowspan = 2> Name </th>
<th colspan = 3> Marks </th>
</tr>
<tr>
<td> FM </td>
<td> FIT </td>
<td> BOM </td>
</tr>
<tr>
<align = “left”>
<td> Rahul </td>
<td> 68 </td>
<td> 68 </td>
<td> 75 </td>
</tr>
<tr>
<td> Rakesh </td>
<td> 60 </td>
<td> 69 </td>
<td> 60 </td>
</tr>
<tr>
<td> Ravi </td>
<td> 68 </td>
<td> 66 </td>
<td> 60 </td>
</tr>
</table>
</body>
</html>
```

Output

Name	Marks		
	FM	FIT	BOM
Rahul	68	68	75
Rakesh	60	69	60
Ravi	68	66	60

1.15 FRAMES

Q28. What is meant by Frames ? Explain disadvantages of frames?

(OR)

Define a frame.

(OR)

Explain in detail about frames in html with suitable example.

Ans : **(Dec.-18, Imp.)**

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages -

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's back button might not work as the user hopes.
- There are still few browsers that do not support frame technology.

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Note - The <frame> tag deprecated in HTML5. Do not use this element.

Example

Following is the example to create three horizontal frames

```
<html>
<head>
<title>HTML Frames</title>
</head>
<framesetrows="10%,80%,10%">
<framename="top"
    src="/html/top_frame.htm"/>
<framename="main"
    src="/html/main_frame.htm"/>
<framename="bottom"
    src="/html/bottom_frame.htm"/>
<noframes>
<body>Your browser does not support frames.</body>
</noframes>
</frameset>
</html>
```

Q29. What is the use of <frameset> tag in html ?

Ans :

The **frameset** is used to group a collection of windows together, sometimes horizontally (using the rows attribute), sometimes vertically (using the cols attribute), or even a combination of horizontal and vertical arrangements. Each child window inside the **frameset** is defined by the **frame** element.

Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically-

```
<html>
<head>
<title>HTML Frames</title>
</head>
<framesetcols="25%,50%,25%">
<framename="left"
    src="/html/top_frame.htm"/>
<framename="center"
    src="/html/main_frame.htm"/>
<framename="right"
    src="/html/bottom_frame.htm"/>
```

```
<noframes>
<body>Your browser does not support frames.</body>
</noframes>
</frameset>

</html>
```

Q30. Explain the attributes supported by <frameset> tag ?

Ans :

Following are important attributes of the <frameset> tag -

1. cols

Specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of the four ways -

Absolute values in pixels. For example, to create three vertical frames, use cols = "100, 500, 100".

A percentage of the browser window. For example, to create three vertical frames, use cols = "10%, 80%, 10%".

Using a wildcard symbol. For example, to create three vertical frames, use cols = "10%, *, 10%". In this case wildcard takes remainder of the window.

As relative widths of the browser window. For example, to create three vertical frames, use cols = "3*, 2*, 1*". This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.

2. rows

This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example, to create two horizontal frames, use rows = "10%, 90%". You can specify the height of each row in the same way as explained above for columns.

3. border

This attribute specifies the width of the border of each frame in pixels. For example, border = "5". A value of zero means no border.

4. frameborder

This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder = "0" specifies no border.

5. framespacing

This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing = "10" means there should be 10 pixels spacing between each frames.

Q31. Explain the attributes supported by <frame >tag briefly.

Ans : **(Dec.-18)**

Following are the important attributes of <frame> tag -

1. src

This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL.

For example, src = "/html/top_frame.htm" will load an HTML file available in html directory.

2. name

This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.

3. frameborder

This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).

4. marginwidth

This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth = "10".

5. marginheight

This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight = "10".

6. noresize

By default, you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize = "noresize".

7. scrolling

This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling = "no" means it should not have scroll bars.

8. longdesc

This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc = "framedescription.htm"

Short Question and Answers

1. HTML is a Markup Language - Explain.

Ans :

A markup language is a computer language that uses tags to define elements within a document. It is human-readable, meaning markup files contain standard words, rather than typical programming syntax. While several markup languages exist, the two most popular are HTML and XML.

HTML is a markup language used for creating webpages. The contents of each webpage are defined by HTML tags. Basic page tags,

such as <head>, <body>, and define sections of the page,

while tags such as <table>, <form>, <image>, and <a> define elements within the page. Most elements require a beginning and end tag, with the content placed between the tags.

2. HTML

Ans :

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

The definition of HTML is HyperText Markup Language.

- HyperText is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.

3. Heading Tags

Ans :

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. While displaying any heading, browser adds one line before and one line after that heading.

```
<html>
<head>
<title>Heading Example</title>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
</body>
</html>
```

4. Line Break Tag.

Ans :

Whenever you use the <br /> element, anything following it starts from the next line. This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

The <br /> tag has a space between the characters br and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <br> it is not valid in XHTML.

```
<html>

<head>
<title>Line Break Example</title>
</head>

<body>
<p>Hello<br/>
You delivered your assignment ontime.<br/>
Thanks<br/>
Mahnaz</p>
</body>
</html>
```

5. How can you format the text in HTML ?

Ans :

HTML Formatting Elements

In the previous chapter, you learned about the HTML **style attribute**.

HTML also defines special **elements** for defining text with a special **meaning**.

HTML uses elements like <b> and <i> for formatting output, like **bold** or italic text.

Formatting elements were designed to display special types of text :

- <b> - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Small text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text.

6. Define Form.

Ans :

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like

- text fields,
- text area fields,
- drop-down menus,
- radio buttons,
- checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following

syntax :

```
<form action = "Script URL" method = "GET|POST">
    form elements like input, textarea etc.
</form>
```

7. Explain Form Attributes.

Ans :

Apart from common attributes, following is a list of the most frequently used form attributes -

1. **action.** Backend script ready to process your passed data.
2. **method.** Method to be used to upload data. The most frequently used are GET and POST methods.
3. **target.** Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
4. **enctype.** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server.

8. Hyperlinks in HTML.

(OR)

Use of anchor tag.

Ans :

An **anchor tag** is an **HTML tag**. It is used to define the beginning and end of a hypertext link. It contains visible words within a text that can be clicked and the URL of the link's target.

The HTML code for creating a link to another page or to a particular section within a page. It is also commonly called an "h-ref."

Links are specified in HTML using the <a>tag.

The <a> can be used in 2 ways

1. To create a link to another document, by using href attribute
2. To create a bookmark inside a document, by using name attribute.

Syntax:

link text

The href attribute specifies the destination of a link.

9. Explain types of list with sample html code.

Ans :

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain -

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

10. HTML Definition Lists

Ans :

HTML and XHTML supports a list style which is called **definition lists** where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition

List makes use of following three tags.

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

Example

```
<!DOCTYPE html>
<html>

<head>
<title>HTML Definition List</title>
</head>

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

</html>
```

This will produce the following result.

HTML : This stands for Hyper Text Markup Language

HTTP : This stands for Hyper Text Transfer Protocol.

11. What is Table ?

Ans :

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. The elements under <td> are regular and left aligned by default

Example

Live Demo

```
<!DOCTYPE html>
<html>

<head>
<title>HTML Tables</title>
</head>
<body>
<tableborder="1">
```

```
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

12. Cellpadding and Cellspacing Attributes

Ans :

There are two attributes called cellpadding and cellspacing which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

Example

```
<html>

<head>
<title>HTML Table Cellpadding</title>
</head>

<body>
<tableborder="1"cellpadding="5"cellspacing="5">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>

</html>
```

13. Colspan and Rowspan Attributes

Ans :

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

Example

```
<html>

<head>
<title>HTML Table Colspan/Rowspan</title>
</head>

<body>
<tableborder="1">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr>
<tdrowspan="2">Row 1 Cell 1</td>
<td>Row 1 Cell 2</td>
<td>Row 1 Cell 3</td>
</tr>
<tr>
<td>Row 2 Cell 2</td>
<td>Row 2 Cell 3</td>
</tr>
<tr>
<tdcolspan="3">Row 3 Cell 1</td>
</tr>
</table>
</body>

</html>
```

14. What is the use of <frameset> tag in html ?

Ans :

The frameset is used to group a collection of windows together, sometimes horizontally (using the rows attribute), sometimes vertically (using the cols attribute), or even a combination of horizontal and vertical arrangements. Each child window inside the frameset is defined by the frame element.

Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically-

```
<html>
<head>
<title>HTML Frames</title>
</head>
<framesetcols="25%,50%,25%">
<framename="left" src="/html/top_frame.htm"/>
<framename="center" src="/html/main_frame.htm"/>
<framename="right" src="/html/bottom_frame.htm"/>
<noframes>
<body>Your browser does not support frames.</body>
</noframes>
</frameset>
</html>
```

15. How can you insert an image in to your web document ?

Ans :

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

Insert Image

You can insert any image in your web page by using **<img>** tag. Following is the simple syntax to use this tag.

```
<imgsrc = "Image URL" ... attributes-list/>
```

The <img> tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

Example

To try following example, let's keep our HTML file test.htm and image file test.png in the same directory -

```
<html>
<head>
<title>Using Image in Webpage</title>
</head>
<body>
<p>Simple Image Insert</p>
<imgsrc="/html/images/test.png" alt="Test Image"/>
</body>
</html>
```

You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive.

The **alt** attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

16. Write short notes on ordered lists with its attributes.

Ans :

Ordered Lists

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using **<ol>** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.

```
<html>

<head>
<title>HTML Ordered List</title>
</head>

<body>
<ol>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ol>
</body>
</html>
```

Output:

1. Beetroot
2. Ginger
3. Potato
4. Radish

The type Attribute

You can use **type** attribute for <ol> tag to specify the type of numbering you like. By default, it is a number. Following are the possible options "

<ol type = "1"> - Default-Case Numerals.

<ol type = "I"> - Upper-Case Numerals.

<ol type = "i"> - Lower-Case Numerals.

<ol type = "A"> - Upper-Case Letters.

<ol type = "a"> - Lower-Case Letters.

17. Write a short notes on marquee tag.

Ans :

The HTML <marquee> tag is used for scrolling piece of text or image displayed either horizontally across or vertically down your web site page depending on the settings.

```
<html>
    <head>
        <title>HTML marquee Tag</title>
        </head>
        <body>
    <marquee>This is basic example of marquee</marquee>
    <marquee direction = "up">The direction of text will be from bottom to top.</marquee>
</body>
</html>
```

Choose the Correct Answer

1. < frameset > is used for _____ [a]
   (a) to group a collection of windows (b) to display a single window
   (c) to display only two (d) none
2. Find which is not an attribute of <frameset> [d]
   (a) cols (b) rows
   (c) border (d) href
3. How many types of list we have [b]
   (a) 4 (b) 3
   (c) 2 (d) 5
4. Order list is Represented by which tag [a]
   (a) <OL> (b) <UL>
   (c) <DL> (d) <DT>
5. ⟨img Src="abc.jpg", height =100⟩ where Src stands for _____ [b]
   (a) Seriece (b) Source
   (c) Service (d) None
6. ⟨tr⟩ is used to create _____ [b]
   (a) table (b) a single Row in a table
   (c) multiple Rows in a table (d) column in a table
7. Following tag is apart of a table [d]
   (a) <thead> (b) <tbody>
   (c) <tfoot> (d) all
8. "bgcolor" is used for _____ [b]
   (a) for attractions (b) for Back ground color's
   (c) for Borders (d) all
9. Find which is not a property of table [d]
   (a) <th> (b) <td>
   (c) <tr> (d) <hr>
10. < ol type ="1" > comes under [a]
    (a) order list (b) un-order list
    (c) definition list (d) all

Fill in the blanks

1. HTML stands for _____.
2. SGML stands for _____.
3. HTML program contains _____.
4. <title> is used to declare _____ of the web document.
5. How many heading tags we have _____.
6. _____ tag displays big & thick heading.
7. _____ tag displays small & thin heading.
8. _____ is used for line braak.
9. To display horizontal line _____ will be useful.
10. get & post methods are a part of _____ tag.

ANSWERS

1. "Hyper Text Mark up Language"
2. "Standard Genralized Mark up Language"
3. tags
4. Name
5. 6
6. h_1
7. h_6
8.
9. <hr>
10. <form>

One Mark Answers

1. What is HTML?

Ans :

HTML stands for Hyper text Markup Language. It is used to create web documents. It supports many tags that's why named as a Markup Language.

2. Define form.

Ans :

Form's are used when you want to collect some data from the visitors and will post it to a back end.

3. Define List.

Ans :

List is used to display the information in a Line - by - Line format. HTML support's Basically 3 types of list they are Order list <oL>, Un-order list <uL>, definition list <dL>.

4. Define frame.

Ans :

Frames are used to divide your Browser window into multiple sections where each section can load a separate HTML document.

5. What is the use of hyper links ?

Ans :

<a> used for hyper links, It creates a link to another document by using "href" attribute.

UNIT II

AN OVER VIEW OF DYNAMIC WEB PAGES & DYNAMIC WEB PAGE :

An over view of dynamic web pages – technologies: Introduction to Dynamic HTML programing - Cascading style sheets (CSS) – types and advantages of CSS – CSS basic syntax and structure - Changing Text and Attributes - Dynamically changing style - Text Graphics and placements - Creating multimedia effects with filters and Transactions.

2.1 INTRODUCTION TO DYNAMIC HTML PROGRAMMING

Q1. What is DHTML ? What are the features of DHTML ?

(OR)

Give a brief introduction about DHTML and its features.

Ans : **(Dec.-20, Dec.-18, Imp.)**

Meaning

DHTML stands for Dynamic hypertext mark up language. DHTML is not a language but a term used to describe the way of making dynamic and interactive web pages.

It is a combination of HTML, JavaScript, Cascading Style Sheets (CSS) and Document Object Model (DOM). Dynamic content is added to static HTML pages using scripts and styles. DHTML uses client side scripting languages like JavaScript to change the static attributes of a HTML page to generate a dynamic effect. This means all DHTML effects achieved are after loading of content on a page without interacting with server again.

Features

All features of dynamic HTML can be classified into four categories as below :

- **Dynamic Content:** Content on the page is modified dynamically based on the user input. Below is the example of the content change when hovering the mouse over the text.
- **Dynamic Style :** The appearance of an element on a web page is modified dynamically like color change or font change. Below is the example of dynamic font change on mouse hover.
- **Dynamic Positioning :** The position of an element is dynamically changed relative to other elements on the page.
- **Dynamic Binding :** Linking an object at run time based on the conditions at that moment.

Q2. What are the advantages of DHTML?

Ans :

Advantages

There are many advantages of using DHTML especially when the content is to be modified dynamically.

- DHTML supports adding styles to static content in various manners.
- It is dynamic so it can be changed even during the run time execution.
- Webmasters are often limited to use default fonts such as Arial or Times Roman. DHTML allows downloadable fonts which make the web pages looking more attractive.
- DHTML page is also saved as an .html file.

It is worth to note here that using multiple scripts on a web page will reduce the page loading speed and slower the site. Also dynamic pages may not perform well on search engines compared to static HTML pages.

Some of the general examples where DHTML is used include :

- Generating animated text
- Content slide from one position to another
- Collect user inputs through a form and processing through client site JavaScript
- Dynamic menu and mouse over events.

Q3. Give a basic introduction about DHTML?

Ans :

DHTML is the art of combining HTML, JavaScript, DOM, and CSS.

Should have a basic understanding of the following Languages :

(i) HTML

(ii) JavaScript

(iii) CSS

- DHTML is NOT a Language
- DHTML stands for Dynamic HTML.
- DHTML is NOT a language or a web standard.

To most people DHTML means the combination of HTML, JavaScript, DOM and CSS.

According to the World Wide Web Consortium (W3C):

"Dynamic HTML is a term used by some vendors to describe the combination of HTML, style sheets and scripts that allows documents to be animated."

(i) HTML

The W3C HTML 4 standard has rich support for dynamic content :

- HTML supports JavaScript
- HTML supports the Document Object Model (DOM)
- HTML supports HTML Events
- HTML supports Cascading Style Sheets (CSS)

DHTML is about using these features, to create dynamic and interactive web pages.

(ii) JavaScript

JavaScript is the most popular scripting language on the internet, and it works in all major browsers.

DHTML is about using JavaScript to control, access and manipulate HTML elements.

HTML DOM

The HTML DOM is a W3C standard. It describes the Document Object Model for HTML.

The HTML DOM defines a standard way for accessing and manipulating HTML documents.

DHTML is about using the DOM to access and manipulate HTML elements.

HTML Events

HTML events are a part of the HTML DOM.

DHTML is about creating web pages that reacts to (user)events.

(iii) CSS

CSS defines how to display HTML elements.

DHTML is about using JavaScript and the HTML DOM to change the style and positioning of HTML elements.

Q4. What are the differences between HTML and DHTML?

(OR)

Explain briefly the differences between HTML and DHTML.

Ans : **(July-19, Imp.)**

S.No.	HTML	S.No.	DHTML
1.	HTML is Hypertext Markup Language	1.	DHTML is Dynamic Hypertext Markup Language.
2.	HTML stands for only static pages.	2.	DHTML is Dynamic HTML means HTML + JavaScript.
3.	It is referred as a static HTML and static in nature.	3.	It is referred as a dynamic HTML and dynamic in nature.
4.	A plain page without any styles and scripts called as HTML.	4.	A page with HTML, CSS, DOM and Scripts called as DHTML.
5.	HTML sites will be slow upon client-side technologies.	5.	DHTML sites will be fast enough upon client-side technologies.

2.2 CASCADING STYLE SHEETS (CSS)

2.2.1 Advantages

Q5. What is CSS? What are the advantages of CSS?

Ans : **(July-19, Dec.-18, Imp.)**

Meaning

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

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.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages

- **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.

2.2.2 Basic Syntax and Structure

Q6. Give a basic syntax and structure of CSS?

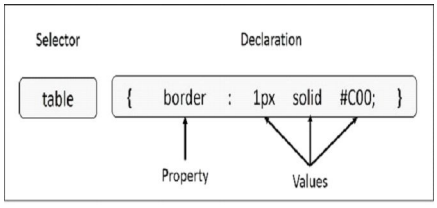
Ans :

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts.

- **Selector** - A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property** - A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
- **Value** - Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.

Syntax

```
selector
{
property: value
 }
```



Example

You can define a table border as follows-

```
Table
{
border :1px solid #C00;
 }
```

Here table is a selector and border is a property and given value 1px solid #C00 is the value of that property.

You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

Q7. Explain briefly about
(i) Class Selectors
(ii) ID Selectors

Ans :

(i) Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black
{
  color: #000000;
}
```

This rule renders the content in black for every element with class attribute set to black in our document. You can make it a bit more particular. For example:

```
h1.black
{
  color: #000000;
}
```

This rule renders the content in black for only <h1> elements with class attribute set to black.

You can apply more than one class selectors to given element. Consider the following example:

This para will be styled by the classes center and bold.

(ii) ID Selectors

You can define style rules based on the id attribute of the elements. All the elements having that id will be formatted according to the defined rule.

```
#black
{
  color: #000000;
}
```

This rule renders the content in black for every element with id attribute set to black in our document. You can make it a bit more particular. For example -

```
h1#black
{
  color: #000000;
}
```

This rule renders the content in black for only <h1> elements with id attribute set to black.

The true power of id selectors is when they are used as the foundation for descendant selectors, For example:

```
#black h2
 {
  color: #000000;
}
```

In this example all level 2 headings will be displayed in black color when those headings will lie with in tags having id attribute set to black.

Q8. What is meant by style sheet ?

Ans :

Cascading Style Sheets (CSS) provide easy and effective alternatives to specify various attributes for the HTML tags. Using CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

```
<html>

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

  <body>
    <p><font color = “green” size = “5”>Hello, World!</font></p>
  </body>

</html>
```

We can re-write above example with the help of Style Sheet as follows

```
<html>
```

```
<head>
  <title>HTML CSS</title>
</head>
<body>
  <p style = "color:green; font-size:24px;" >Hello, World!</p>
</body>
</html>
```

OUTPUT :

Hello, World!

2.2.3 Types of CSS

Q9. Explain the different types of CSS with an examples.

(OR)

Explain in detail about various types of CSS.

(OR)

List out various types of CSS.

Ans : **(Dec.-20, July-19, Dec.-18, Imp.)**

CSS supports mainly 3 types of style sheets.

1. External style sheet
2. Internal style sheet (embedded style sheet)
3. Inline style sheet

1. External Style Sheet

If you need to use your style sheet to various pages, then its always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as **.css** and it will be included in HTML files using <link> tag.

Example

Consider we define a style sheet file **style.css** which has following rules

```
.red {
   color: red;
}
.thick {
   font-size:20px;
}
.green {
   color:green;
}
```

Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. I suggest you should not bother about how these rules are being defined because you will learn them while studying CSS. Now let's make use of the above external CSS file in our following HTML document -

```
<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>
```

OUTPUT :

This is red

This is thick

This is green

This is thick and green

2. Internal Style Sheet (or) Embedded Style Sheet

If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using <style> tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

Example

Let's re-write above example once again, but here we will write style sheet rules in the same HTML document using <style> tag

```
<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>
```

OUTPUT :

This is red

This is thick

This is green

This is thick and green

3. Inline Style Sheet

You can apply style sheet rules directly to any HTML element using style attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

Example

Let's re-write above example once again, but here we will write style sheet rules along with the HTML elements using style attribute of those elements.

```
<!DOCTYPE html>
<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>
```

Output

This is red

This is thick

This is green

This is thick and green.

2.3 EVENT HANDLING

Q10. What is meant by Event Handling?

Ans :

Meaning

Event handling is the receipt of an event at some event handler from an event producer and subsequent processes. The processes involved in event handling include: Identifying where an event should be forwarded. Making the forward.

Event handling is the receipt of an event at some event handler from an event producer and subsequent processes. The processes involved in event handling include:

- Identifying where an event should be forwarded.
- Making the forward.
- Receiving the forwarded event.
- Taking some kind of appropriate action in response, such as writing to a log, sending an error or recovery routine or sending a message.
- The event handler may ultimately forward the event to an event consumer.

In programming, an event is an action that occurs as a result of the user or another source, such as a mouse being clicked, or a key being pressed. An event handler is a routine that is used to deal with the event, allowing a programmer to write code that will be executed when the event occurs.

Examples

- A web browser completely loading a web page.
- A file being created or modified on a filesystem.
- A hardware sensor such as a webcam or microphone receiving sensory input.
- The arrival of incoming network traffic.
- The occurrence of an error at the program or system level.

Events on the Web

Another example of an event is a user clicking on a button within a web page. This action creates what is known as a "click" event. JavaScript could then be used to program a reaction to the event, for instance, you could use the onclick event handler shown in the following box:

```
<form action="#" method="post">
<input type="button" value="Click for a Message" onclick="window.alert('Hello!');">
</form>
```

The code reacts to a click event on the button, sending the "Hello" message the viewer.

2.4 CHANGING TEXT AND ATTRIBUTES

Q11. How can we format the text in CSS? What are the attributes of text? Explain in detail with sample program.

Ans :

Following are the properties of TEXT

1. Text Color

The color property is used to set the color of the text. The color is specified by:

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

Look at CSS Color Values for a complete list of possible color values.

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

Example

```
Body
 {
     color: blue;
}

h1 {
     color: green;
}
```

OUTPUT :

This is heading 1

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

2. Text Alignment

The text-align property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

The following example shows center aligned, and left and right aligned text (left alignment is default if text direction is left-to-right, and right alignment is default if text direction is right-to-left):

Example

```
h1 {
    text-align: center;
}

h2 {
    text-align: left;
}

h3 {
    text-align: right;
}
```

OUTPUT :

Heading 1 (center)

Heading 2 (left)

Heading 3 (right)

The three headings above are aligned center, left and right.

3. Text Decoration

The text-decoration property is used to set or remove decorations from text.

The value text-decoration: none; is often used to remove underlines from links:

Example

```
A
 {
    text-decoration: none;
}
```

OUTPUT :

A link with no underline: W3Schools.com

4. Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

Example

```
p.uppercase {
    text-transform: uppercase;
}
p.lowercase {
    text-transform: lowercase;
}
p.capitalize {
    text-transform: capitalize;
}
```

OUTPUT :

This is some text.

This is some text.

This is some text.

5. Text Indentation

The text-indent property is used to specify the indentation of the first line of a text:

Example

```
p {
    text-indent: 50px;
}
```

6. Letter Spacing

The letter-spacing property is used to specify the space between the characters in a text.

The following example demonstrates how to increase or decrease the space between characters:

Example

```
h1 {
    letter-spacing: 3px;
}

h2 {
    letter-spacing: -3px;
}
```

7. Line Height

The line-height property is used to specify the space between lines:

Example

```
p.small {
    line-height: 0.8;
}

p.big {
    line-height: 1.8;
}
```

8. Text Direction

The direction property is used to change the text direction of an element:

Example

```
p {
    direction: rtl;
}
```

9. Word Spacing

The word-spacing property is used to specify the space between the words in a text.

The following example demonstrates how to increase or decrease the space between words:

Example

```
h1 {
    word-spacing: 10px;
}

h2 {
    word-spacing: -5px;
}
s
```

The text-shadow property adds shadow to text.

The following example specifies the position of the horizontal shadow (3px), the position of the vertical shadow (2px) and the color of the shadow (red):

Example

```
h1 {
    text-shadow: 3px 2px red;
}
```

2.5 Dynamically Changing Style

Q12. How can you make changes dynamically?

Ans :

The CSS Object Model (CSSOM), part of the DOM, exposes specific interfaces allowing manipulation of a wide amount of information regarding CSS. Initially defined in the DOM Level 2 Style recommendation, these interfaces forms now a specification, CSS Object Model (CSSOM) which aims at superseding it.

In many cases, and where possible, it really is best practice to dynamically manipulate classes via the class Name property since the ultimate appearance of all of the styling hooks can be controlled in a single stylesheet. One's JavaScript code also becomes cleaner since instead of being dedicated to styling details, it can focus on the overall semantics of each section it is creating or manipulating, leaving the precise style details to the stylesheet.

However, there are cases where actually obtaining or manipulating the rules can be useful (whether for whole stylesheets or individual elements), and that is described in further detail below. Note also that, as with individual element's DOM styles, when speaking of manipulating the stylesheets, this is not actually manipulating the physical document(s), but merely the internal representation of the document.

The basic style object exposes the Stylesheet and the CSS Stylesheet interfaces. Those interfaces contain members like insertRule, selectorText, and parentStyleSheet for accessing and manipulating the individual style rules that make up a CSS stylesheet.

To get to the style objects from the document, you can use the document.styleSheets property and access the individual objects by index.

2.6 Text Graphics and Placements

Q13. Explain briefly dynamic changes to text, and graphics.

Ans : **(Imp.)**

Dynamic Changes to Text

In DHTML, a user can modify the content of a HTML document using Document Object Model (DOM). The DOM allow user to access all the element of HTML document. That is using DOM the text and attribute of HTML document can be created, inserted, deleted or modified.

Moreover, the dynamic and interactive content of DHTML can be viewed with the help of less capable browsers like interest explorer. This is

because interest explorer is built with rich set of properties and methods with which the DHTML document can be dynamically constructed and already.

Example

The following document uses some lines of IE-specific script to replace and change some elements.

```
<DOCTYPE html>
<html>
<head>
<title>Changing Text</title>
</head>
<body>
<h1 id="header">Header2<\h1>
<script type="text/javascript">
document.getElementById("header"),innerHTML="Header3";
</script>
<p>The header text has not been changed</p>
</body>
</html>
```

Header2

The header test has not been changed from Header2 to Header3

Dynamic Changes to Style

In DHTML, dynamic changes to styles can be made with the help of Cascading Style Sheet (CSS). CSS basically allow to modify the appearance of html document as well as format it This modification can be done without adding/deleting any elements of HTML, document. The main advantage of CSS is that, it helps in writing small HTML document which can be manipulated easily and quickly.

Styles in HTML can be accessed by using object model. For instance, for modifying the inline style sheet a simple script based programming languages like Javascript, microJavaScript, microsoft Visual Basic Scripting Edition(VBScript) can be used.

Example

```
<html>
<head>
<title>Changing Styles</title>
<script language="Javascript">
    function ChangeStyle(i,f,s)
    {
        if(Idocument.getElementById) return;
      obj=document.getElementById(i);
      obj.style[f]=s;
    }
```

```
</script>
<body>
    <h1 ID="head1">Dynamic Styles</h1>
    <p ID="Paragraph"> A user can change the font style of the text by simply clicking on the
        link of provided font for example
    <a href="javascript:ChangeStyle('Paragraph', 'fontFamily', 'cursive');"> cursive </a>
    <a href="javascript:ChangeStyle('Paragraph', 'fontFamily', 'arial');"> arial </a>
    </p>
</body>
</html>
```

Dynamic Changes to Graphics

In DHTML, dynamic changes to graphics involves changing of elements by its size and positions, or by applying effects and animations etc. This can be done using DHTML technologies such as CSS, DOM and script etc.

Positioning is a CSS component that can help in placing an html element to some specified location in a html page. This positioning is done either on browser window or in another page using keywords like absolute, relative and fixed. Elements such as images, controls and text can be placed at the desired location by defining the 'top' and 'left' coordinates.

Positioning Using CSS

The positioning of an element can be done by setting certain CSS attributes. Some of these attributes may include, top, left, right, bottom, display etc.

Consider the below code to understand how to set the absolute position of a graphic with CSS.

Example

```
<html>
<head>
<title>Positioning Using CSS</title>
<style type="text/css">
#someSpan {position:absolute; left:30px; top:50px}
</style>
</head>
<body>
<div id="someDiv" style="position:absolute; left:0px; top:0px">
Rahul Publications
<span id="someSpan">
Rahul Publications
</spn>
</div>
</body>
</html>
```

Output

Rahul Publications

2.7 Creating Multimedia Effects with Filters and Transactions

Q14. Explain filters and transactions for multimedia effects.

(OR)

Explain in detail about filters and transitions.

Ans : **(Dec.-20, Dec.-18, June-19, Imp.)**

CSS filters to add special effects to text, images and other aspects of a webpage without using images or other graphics. Filters only work on Internet Explorer 4.0+,. If you are developing your site for multi browsers, then it may not be a good idea to use CSS filters because there is a possibility that it would not give any advantage.

(i) Filters

Following are the filters/effects generally in use.

1. Alpha Channel

The Alpha Channel filter alters the opacity of the object, which makes it blend into the background. The following parameters can be used in this filter.

Parameter	Description
opacity	Level of the opacity. 0 is fully transparent, 100 is fully opaque.
finishopacity	Level of the opacity at the other end of the object.
style	The shape of the opacity gradient. 0 = uniform 1 = linear 2 = radial 3 = rectangular
startX	X coordinate for opacity gradient to begin.
startY	Y coordinate for opacity gradient to begin.
finishX	X coordinate for opacity gradient to end.
finishY	Y coordinate for opacity gradient to end.

Example

```
<html>
  <head>
  </head>
   <body>
    <p>Image Example:</p>
   <img src="/css/images/logo.png" alt="CSS Logo"
     style="Filter: Alpha(Opacity=100,
     FinishOpacity=0,
     Style=2,
     StartX=20,
```

```
StartY=40,
FinishX=0,
FinishY=0)" />
<p>Text Example:</p>
<div style="width: 357;
height: 50;
font-size: 30pt;
font-family: Arial Black;
color: blue;
Filter: Alpha(Opacity=100, FinishOpacity=0, Style=1, StartX=0, StartY=0, FinishX=580,
FinishY=0)">CSS Tutorials</div>
</body>
</html>
```

2. Motion Blur

Motion Blur is used to create blurred pictures or text with the direction and strength. The following parameters can be used in this filter.

Parameter	Description
add	True or false. If true, the image is added to the blurred image; and if false, the image is not added to the blurred image.
direction	The direction of the blur, going clockwise, rounded to 45-degree increments. The default value is 270 (left). 0 = Top 45 = Top right 90 = Right 135 = Bottom right 180 = Bottom 225 = Bottom left 270 = Left 315 = Top left
strength	The number of pixels the blur will extend. The default is 5 pixels.

Example

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

```
<p>Image Example:</p>
<img src="/css/images/logo.png" alt="CSS Logo"
  style="Filter: Blur(Add = 0, Direction = 225, Strength = 10)">
<p>Text Example:</p>
<div style="width: 357;
  height: 50;
  font-size: 30pt;
  font-family: Arial Black;
  color: blue;
  Filter: Blur(Add = 1, Direction = 225, Strength = 10)">CSS Tutorials</div>
</body>
</html>
```

3. Chroma Filter

Chroma Filter is used to make any particular color transparent and usually it is used with images. You can use it with scrollbars also. The following parameter can be used in this filter "

Parameter	Description
color	The color that you'd like to be transparent.

4. Drop Shadow Effect

Drop Shadow is used to create a shadow of your object at the specified X (horizontal) and Y (vertical) offset and color.

The following parameters can be used in this filter -

Parameter	Description
color	The color, in #RRGGBB format, of the dropshadow.
offX	Number of pixels the drop shadow is offset from the visual object, along the x-axis. Positive integers move the drop shadow to the right, negative integers move the drop shadow to the left.
offY	Number of pixels the drop shadow is offset from the visual object, along the y-axis. Positive integers move the drop shadow down, negative integers move the drop shadow up.
positive	If true, all opaque pixels of the object have a dropshadow. If false, all transparent pixels have a dropshadow. The default is true.

5. Flip Effect

Flip effect is used to create a mirror image of the object. The following parameters can be used in this filter -

Parameter	Description
FlipH	Creates a horizontal mirror image
FlipV	Creates a vertical mirror image

6. Glow Effect

Glow effect is used to create a glow around the object. If it is a transparent image, then glow is created around the opaque pixels of it. The following parameters can be used in this filter "

Parameter	Description
Color	The color you want the glow to be
Strength	The intensity of the glow (from 1 to 255)

7. Grayscale Effect

Grayscale effect is used to convert the colors of the object to 256 shades of gray. The following parameter is used in this filter -

Parameter	Description
gray	Converts the colors of the object to 256 shades of gray.

8. Invert Effect

Invert effect is used to map the colors of the object to their opposite values in the color spectrum, i.e., to create a negative image. The following parameter is used in this filter "

Parameter	Description
Invert	Maps the colors of the object to their opposite value in the color spectrum.

9. Mask Effect

Mask effect is used to turn transparent pixels to a specified color and makes opaque pixels transparent. The following parameter is used in this filter -

Parameter	Description
Color	The color that the transparent areas will become

10. Shadow Filter

Shadow filter is used to create an attenuated shadow in the direction and color specified. This is a filter that lies in between Dropshadow and Glow. The following parameters can be used in this filter -

Parameter	Description
color	The color that you want the shadow to be.
direction	The direction of the blur, going clockwise, rounded to 45-degree increments. The default value is 270 (left). 0 = Top 45 = Top right 90 = Right 135 = Bottom right 180 = Bottom 225 = Bottom left 270 = Left 315 = Top left

11. Wave Effect

Wave effect is used to give the object a sine wave distortion to make it look wavy. The following parameters can be used in this filter -

Parameter	Description
Add	A value of 1 adds the original image to the waved image, 0 does not.
Freq	The number of waves.
Light	The strength of the light on the wave (from 0 to 100).
Phase	At what degree the sine wave should start (from 0 to 100).
strength	The intensity of the wave effect.

12. X-Ray Effect

X-Ray effect grayscales and flattens the color depth. The following parameter is used in this filter:

Parameter	Description
xray	Grayscales and flattens the color depth.

(ii) Transition

Transitions are filters that provide effects to the context of web page. They are responsible for visually changing the control/moving a page from one state to another. They provide time-based effects thereby allowing the user to create animation in images.

1. Reveal transition filter
2. Blend transition filter

1. Reveal transition filter

The reveal transition filter written as Reveal? Trans is applied to multiple visual objectives together inorder to show/hide them.

Syntax

STYLE = "filter:revealtrans(duration = duration, transition = transitionshape)

Here, duration is the value/time taken duration by the transition. It is exposed in "seconds.milliseconds".

2. Blend transition filter

The blend transition filter written as BlendTrans is applied to a visual object to fadein/fadeout for certain time limit.

Syntax

STYLE = "filter: blendtrans(duration = duration";

Here, duration is the value/time duration taken by the transition. It is also expressed in "seconds.milliseconds".

Short Question and Answers

1. What is DHTML?

Ans :

DHTML stands for Dynamic hypertext mark up language.DHTML is not a language but a term used to describe the way of making dynamic and interactive web pages.

It is a combination of HTML, JavaScript, Cascading Style Sheets (CSS) and Document Object Model (DOM). Dynamic content is added to static HTML pages using scripts and styles. DHTML uses client side scripting languages like JavaScript to change the static attributes of a HTML page to generate a dynamic effect. This means all DHTML effects achieved are after loading of content on a page without interacting with server again.

2. List out the Features of DHTML.

Ans :

All features of dynamic HTML can be classified into four categories as below :

- **Dynamic Content:** Content on the page is modified dynamically based on the user input. Below is the example of the content change when hovering the mouse over the text.
- **Dynamic Style :** The appearance of an element on a web page is modified dynamically like color change or font change. Below is the example of dynamic font change on mouse hover.
- **Dynamic Positioning :** The position of an element is dynamically changed relative to other elements on the page.
- **Dynamic Binding :** Linking an object at run time based on the conditions at that moment.

3. Write some Advantages of DHTML.

Ans :

There are many advantages of using DHTML especially when the content is to be modified dynamically.

- DHTML supports adding styles to static content in various manners.
- It is dynamic so it can be changed even during the run time execution.
- Webmasters are often limited to use default fonts such as Arial or Times Roman. DHTML allows downloadable fonts which make the web pages looking more attractive.
- DHTML page is also saved as an .html file.

It is worth to note here that using multiple scripts on a web page will reduce the page loading speed and slower the site. Also dynamic pages may not perform well on search engines compared to static HTML pages.

4. Differences between HTML and DHTML.

Ans :

	HTML	DHTML
1.	HTML is Hypertext Markup Language	DHTML is Dynamic Hypertext Markup Language.
2.	HTML stands for only static pages.	DHTML is Dynamic HTML means HTML + JavaScript.
3.	It is referred as a static HTML and static in nature.	It is referred as a dynamic HTML and dynamic in nature.
4.	A plain page without any styles and scripts called as HTML.	A page with HTML, CSS, DOM and Scripts called as DHTML.
5.	HTML sites will be slow upon client-side technologies.	DHTML sites will be fast enough upon client-side technologies.

5. Define CSS.

Ans :

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

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.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

6. Define External style sheet.

Ans :

External Style Sheet

If you need to use your style sheet to various pages, then its always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as **.css** and it will be included in HTML files using <link> tag.

Example

Consider we define a style sheet file **style.css** which has following rules

```
.red {
   color: red;
}
.thick {
   font-size:20px;
}
.green {
   color:green;
}
```

Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. I suggest you should not bother about how these rules are being defined because you will learn them while studying CSS. Now let's make use of the above external CSS file in our following HTML document -

```
<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>
```

OUTPUT :

This is red

This is thick

This is green

This is thick and green.

7. Define Internal Style Sheet.

Ans :

If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using <style> tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

Example

Let's re-write above example once again, but here we will write style sheet rules in the same HTML document using <style> tag

```
<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>
```

OUTPUT :

This is red
This is thick
This is green
This is thick and green

8. Define Inline Style Sheet.

Ans :

You can apply style sheet rules directly to any HTML element using **style** attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

Example

Let's re-write above example once again, but here we will write style sheet rules along with the HTML elements using **style** attribute of those elements.

```
<!DOCTYPE html>
<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>
```

Output

This is red

This is thick

This is green

This is thick and green.

9. What is Event Handling.

Ans :

Event handling

Event handling is the receipt of an **event** at some **event handler** from an **event** producer and subsequent processes. The processes involved in **event handling** include: Identifying where an **event** should be forwarded. Making the forward.

Event handling is the receipt of an event at some event handler from an event producer and subsequent processes. The processes involved in event handling include:

- Identifying where an event should be forwarded.
- Making the forward.
- Receiving the forwarded event.
- Taking some kind of appropriate action in response, such as writing to a log, sending an error or recovery routine or sending a message.
- The event handler may ultimately forward the event to an event consumer.

10. Explain filters and transactions for creating multimedia effects.

Ans :

CSS filters to add special effects to text, images and other aspects of a webpage without using images or other graphics. Filters only work on Internet Explorer 4.0+,. If you are developing your site for multi browsers, then it may not be a good idea to use CSS filters because there is a possibility that it would not give any advantage.

In this answer, we will see the details of each CSS filter. These filters may not work in your browser.

Alpha Channel

The Alpha Channel filter alters the opacity of the object, which makes it blend into the background. The following parameters can be used in this filter.

Parameter	Description
opacity	Level of the opacity. 0 is fully transparent, 100 is fully opaque.
finishopacity	Level of the opacity at the other end of the object.
style	The shape of the opacity gradient. 0 = uniform 1 = linear 2 = radial 3 = rectangular
startX	X coordinate for opacity gradient to begin.
startY	Y coordinate for opacity gradient to begin.
finishX	X coordinate for opacity gradient to end.
finishY	Y coordinate for opacity gradient to end.

Choose the Correct Answer

1. Find which of the following is a feature of the DHTML. [d]
   (a) Dynamic Style (b) Dynamic Positioning
   (c) Dynamic Binding (d) All
2. DHTML is a combination of ____ Languages. [b]
   (a) 2 (b) 3
   (c) 4 (d) 5
3. DOM ____ Defines a standard way for accessing and maniplating HTML Documents [b]
   (a) CSS (b) DOM
   (c) Event (d) Filters
4. DHTML creates ____ web pages ? [c]
   (a) Static (b) Private
   (c) Dynamic (d) None
5. How many types of css - we have ? [a]
   (a) 3 (b) 4
   (c) 5 (d) 2
6. Which one is an advantage of css? [d]
   (a) Time saving (b) Easy maintenance
   (c) Offline browsing (d) All
7. ____ style sheet is used to define style rules directly along with html elements [c]
   (a) Extermal (b) Embedded
   (c) Inline (d) All
8. Which one is not a properly of css text ? [d]
   (a) Text -color (b) Alignment
   (c) Decoration (d) Source
9. Find which one is a filter? [d]
   (a) Alpha channel (b) Motion Blue
   (c) Flip Effect (d) All
10. ____ is a Mechanism used to handle the events [b]
    (a) Event controlling (b) Event handling
    (c) Event Monitoring (d) None

Fill in the blanks

1. DHTML stands for _____.
2. DHTML is a combination of _____.
3. DOM stands for _____.
4. CSS stands for _____.
5. _____ sheet, rules in header section of the <html> document.
6. _____ to add special effects to web page, with out using graphics.
7. A style Rule is made up of _____.
8. We can define style rules based on _____ atributes of the Element.
9. <Link> is used in _____ style sheet.
10. Event handling is the receipt of on _____.

ANSWERS

1. "Dynamic Hyper Text Mark up Language"
2. HTML + Java Script + CSS
3. Document object Model
4. Cascading Style Sheets
5. Internal Style
6. CSS-filters
7. Sector Property and value
8. Class
9. Extermal
10. Event

One Mark Answers

1. DHTML

Ans :

DHTML is not a Language. But a term used to describe the way of making Dynamic and interactive web-pages.

2. DOM

Ans :

DOM stands for document object modd, It defines a standard way for accessing and manipulating HTML documents.

3. CSS

Ans :

It is a simple design language intended to simplify the process of making web pages presentable.

4. External style sheet

Ans :

Define style sheet Rules in a separate "CSS" file and then include that file in your HTML Documents using <Link> tag.

5. Filters

Ans :

Used to add special effect's to text, images and other aspects of web pages.

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 - Eventhandling

3.1 JAVA SCRIPT

3.1.1 Introduction

Q1. Give a brief introduction about java script ?

Ans :

Meaning

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

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform.

Q2. What are the advantages and limitations of java script ?

Ans : **(Imp.)**

Advantages

The merits of using JavaScript are -

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

Limitations

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multithreading or multiprocessor capabilities.

Q3. Give a syntax and basics of java script with simple program?

Ans :

JavaScript - Syntax

JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script>.

You can place the <script> tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the <head> tags.

The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

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

The script tag takes two important attributes-

- **Language** - This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- **Type** - This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

So your JavaScript segment will look like -

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

Let us take a sample example to print out "Hello World". We added an optional HTML comment that surrounds our JavaScript code. This is to save our code from a browser that does not support JavaScript. The comment ends with a "// —>". Here "//" signifies a comment in JavaScript, so we add that to prevent a browser from reading the end of the HTML comment as a piece of JavaScript code. Next, we call a function document.write which writes a string into our HTML document.

This function can be used to write text, HTML, or both. Take a look at the following code.

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

OUTPUT :

Hello World!

Whitespace and Line Breaks

JavaScript ignores spaces, tabs, and newlines that appear in JavaScript programs. You can use spaces, tabs, and newlines freely in your program and you are free to format and indent your programs in a neat and consistent way that makes the code easy to read and understand.

Semicolons are Optional

Simple statements in JavaScript are generally followed by a semicolon character, just as they are in C, C++, and Java. JavaScript, however, allows you to omit this semicolon if each of your statements are placed on a separate line. For example, the following code could be written without semicolons.

```
<script language = "javascript" type = "text/
javascript" >
<!—
var1 = 10
var2 = 20
 //—>
</script>
```

Case Sensitivity

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

So the identifiers Time and TIME will convey different meanings in JavaScript.

Comments in JavaScript

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

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

Example :

The following example shows how to use comments in JavaScript.

```
<script language="javascript" type="text/javascript">
        <!—
// This is a comment. It is similar to comments in C++
        /*
* This is a multiline comment in JavaScript
* It is very similar to comments in C Programming
        */
        //—>
        </script>
```

3.2 CLIENT SIDE JAVA SCRIPTING

Q4. What is client side java scripting ? explain?

Ans :

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Client-side means that the JavaScript code is run on the client machine, which is the browser. Server-side JavaScript means that the code is run on the server which is serving web pages. One runs in the browser (client side), the other runs on the server.

Client-side components usually consist of HTML, CSS, JavaScript, Ajax, JS libraries, images, and whatever other files that are to be downloaded to the browser. On the server, you need a listener to process requests, fetch resources or information, and manipulate them so that they can be sent back to the client.

3.3 SERVER SIDE JAVA SCRIPTING

Q5. What is server side java scripting ? explain?

Ans :

Although rich web sites and applications have become the norm in recent years, they can still be difficult to develop, maintain, and expand. Many of the challenges stem from the schism between client and server components. Client-side components usually consist of HTML, CSS, JavaScript, Ajax, JS libraries, images, and whatever other files that are to be downloaded to the browser. On the server, you need a listener to process requests, fetch resources or information, and manipulate them so that they can be sent back to the client.

This is usually accomplished using XML, JSON, or HTML-Formatted text, which is sent across the wire using Ajax. There are a number of competing technologies to choose from here. Depending on your traffic, hardware, O/S, bandwidth, IT expertise, and numerous other factors, there is a technology for every taste and occasion. Popular server-side languages at this time include PHP, Java, and .NET, to name only a few.

There is presently a ServerJS movement whose goal is to eliminate the gap between client and server. Exponents of the group want to keep with HTML, JavaScript, and CSS, which are most familiar to the end-users. What makes server-side JavaScript possible is a web server than can process the code.

One such server is called Jaxer. Developed by Aptana, Jaxer is an open source Ajax web server for building rich web pages and applications using a unified Ajax model that can be written entirely using JavaScript. Writing code for Jaxer is the focus of this article.

Q6. What are the differences between client -side and server-side scripting?

Ans :

Client-side Environment

The client-side environment used to run scripts is usually a browser. The processing takes place on the end users computer. The source code is transferred from the web server to the users computer over the internet and run directly in the browser.

The scripting language needs to be enabled on the client computer. Sometimes if a user is conscious of security risks they may switch the scripting facility off. When this is the case a message usually pops up to alert the user when script is attempting to run.

Server-side Environment

The server-side environment that runs a scripting language is a web server. A user's request is fulfilled by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser. It is usually used to provide interactive web sites that interface to databases or other data stores on the server.

This is different from client-side scripting where scripts are run by the viewing web browser, usually in JavaScript. 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.

3.4 Core Features

Q7. How java is differ from java script?

Ans :

The JavaScript programming language, developed by Netscape, Inc., is not part of the Java platform.

JavaScript does not create applets or stand-alone applications. In its most common form, JavaScript resides inside HTML documents, and can provide levels of interactivity to web pages that are not achievable with simple HTML.

Key differences between Java and JavaScript:

- Java is an OOP programming language while Java Script is an OOP scripting language.
- Java creates applications that run in a virtual machine or browser while JavaScript code is run on a browser only.
- Java code needs to be compiled while JavaScript code are all in text.
- They require different plug-ins.

For additional information about JavaScript, visit Mozilla.org

JavaScript Error javascript.JSException:

Unknown name

When loading the webpage an error is displayed:

JavaScript Error javascript.JSException:

Unknown name

Possible causes:

1. JavaScript is not enabled in the browser.
2. The browser does not support JavaScript technology.
3. The web page has a JavaScript programming error.

If JavaScript is already enabled in the web browser, then the error message indicates that there is a programming error in the JavaScript code. Contact the webmaster of that site to report the error including:

1. Error message
2. Type and version of the browser used
3. URL of the web page that displayed the error message
4. Steps to reproduce the error

Q8. How/where can you write or insert java script in html code?

Ans :

Here is a flexibility given to include JavaScript code anywhere in an HTML document. However the most preferred ways to include JavaScript in an HTML file are as follows -

- Script in <head>...</head> section.
- Script in <body>...</body> section.
- Script in <body>...</body>and <head> ...</head> sections.
- Script in an external file and then include in <head>...</head> section.

In the following section, we will see how we can place JavaScript in an HTML file in different ways.

JavaScript in <head>...</head> section

If you want to have a script run on some event, such as when a user clicks somewhere, then you will place that script in the head as follows-

```
<html>
<head>
<script type="text/javascript">
<!—
function sayHello() {
alert("Hello World")
}
//—>
</script>
</head>
<body>
<input type="button" onclick="sayHello()"
value="Say Hello" />
</body>
</html>
```

This code will produce the following results -

JavaScript in <body>...</body> section

If you need a script to run as the page loads so that the script generates content in the page, then the script goes in the <body> portion of the document. In this case, you would not have any function defined using JavaScript. Take a look at the following code.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
<!—
document.write("Hello World")
//—>
</script>
<p>This is web page body </p>
</body>
</html>
```

This code will produce the following results -

JavaScript in <body> and <head> Sections

You can put your JavaScript code in <head> and <body> section altogether as follows -

```
<html>
<head>
<script type="text/javascript">
<!—
function sayHello() {
alert("Hello World")
}
//—>
</script>
</head>
<body>
<script type="text/javascript">
```

```
<!—
document.write("Hello World")
//—>
</script>
<input type="button" onclick="sayHello()"
value="Say Hello"
</body>
</html>
```

This code will produce the following result -

JavaScript in External File

As you begin to work more extensively with JavaScript, you will be likely to find that there are cases where you are reusing identical JavaScript code on multiple pages of a site.

You are not restricted to be maintaining identical code in multiple HTML files. The script tag provides a mechanism to allow you to store JavaScript in an external file and then include it into your HTML files.

Here is an example to show how you can include an external JavaScript file in your HTML code using script tag and its src attribute.

```
<html>
<head>
<scripttype="text/javascript"src="
filename.js" > </script>
</head>
<body>
.....
</body>
</html>
```

To use JavaScript from an external file source, you need to write all your JavaScript source code in a simple text file with the extension ".js" and then include that file as shown above.

For example, you can keep the following content in filename.js file and then you can use sayHello function in your HTML file after including the filename.js file.

```
function sayHello() {
alert("Hello World")
}
```

3.5 Data Types & Variables

Q9. What are the data types supported by javascript?

(or)

Explain the data types supported by Javascript.

Ans : **(June-19, Dec-18, Imp.)**

In computer science and computer programming, a data type or simply type is a classification of data which tells the compiler or interpreter how the programmer intends to use the data.

JavaScript includes data types similar to other programming languages like Java or C#. Data type indicates characteristics of data. It tells the compiler whether the data value is numeric, alphabetic, date etc., so that it can perform the appropriate operation.

JavaScript includes primitive and non-primitive data types as per latest ECMAScript 5.1.

Primitive Data Types:

1. String
2. Number
3. Boolean
4. Undefined
5. Null

Sl.No.	Data Type	Description
1.	String	represents sequence of characters e.g. “hello”
2.	Number	represents numeric values e.g. 100
3.	Boolean	represents boolean value either false or true
4.	Undefined	represents undefined value
5.	Null	represents null i.e. no value at all

Non-primitive Data Type

1. Object
2. Array
3. RegExp

Sl.No.	Data Type	Description
1.	Object	represents instance through which we can access members
2.	Array	represents group of similar values
3.	RegExp	represents regular expression

JavaScript is a dynamic or loosely-typed language because a variable can hold value of any data type at any point of time.

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

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

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

Q10. What is meant by variable? How java script allows variables?

Ans :

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

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

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

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

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

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

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

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

Q11. What is local variables and global variables?

(or)

What is the difference between local variables and global variables?

Ans : **(Imp.)**

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

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

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

Example

```
<html>
<body onload = checkscope();>
<script type = “text/javascript”>
        <!—
var myVar = “global”; // Declare a global
variable
    function checkscope( ) {
    var myVar = “local”; //
    Declare a local variable
    document.write(myVar);
}
        //—>
        </script>
        </body>
        </html>
```

OUTPUT :

Local

3.6 OPERATORS

Q12. What is meant by operator ? how many types of operators supported by java script?

Ans : **(Dec.-20, June-19, Imp.)**

(i) Arithmetic Operators

(ii) Comparison Operatorss

(iii) Logical (or Relational) Operators

(iv) Bitwise Operators

(v) Assignment Operators

(vi) Conditional (or ternary) Operators

Lets have a look on all operators one by one.

(i) Arithmetic Operators in Java’s cripts

JavaScript Xsupports the following arithmetic operators -

Assume variable A holds 10 and variable B holds 20, then -

Operator	Description
Addition	Adds two operandsEx: A + B will gve 30
Subtraction	Subtracts the second operand from the firstEx: A - B will gve -10
Multiplication	Multiply both operands Ex: A* B will give 200
Division	Divide the numerator by the denominatorEx: B/A will give 2
Modulus	Outputs the remainder of an integer divisionEx: B % A will give 0
++(Increment)	Increases an integer value by oneEx: A++ will give 11
--(Decrement)	Decreases an integer value by oneEx: A- will give 9

Note - Addition operator (+) works for Numeric as well as Strings. e.g. "a" + 10 will give "a10".

Example

```
<html>
 <body>
<script type="text/javascript">
 <!—
 var a = 33;
 var b = 10;
 var c = "Test";
 var linebreak = "<br />";
 document.write("a + b = ");
 result = a + b;
 document.write(result);
 document.write(linebreak);
 document.write("a - b = ");
 result = a - b;
 document.write(result);
 document.write(linebreak);
document.write("a / b = ");
result = a / b;
 document.write(result);
 document.write(linebreak);
document.write("a % b = ");
result = a % b;
 document.write(result);
 document.write(linebreak);
document.write("a + b + c = ");
result = a + b + c;
 document.write(result);
 document.write(linebreak);
a = ++a;
 document.write("++a = ");
result = ++a;
 document.write(result);
 document.write(linebreak);
b = —b;
 document.write("—b = ");
result = —b;
 document.write(result);
 document.write(linebreak);
 //—>
 </script>
```

Set the variables to different values and then try...

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

Output :

```
a + b = 43
a - b = 23
a / b = 3.3
a % b = 3
a + b + c = 43
++a = 35
– b = 8
```

Set the variables to different values and then try...

(ii) Comparison Operators

JavaScript supports the following comparison operators -

Assume variable A holds 10 and variable B holds 20, then-

Operator	Description
= = (Equal)	Checks if the value of two operands are equal or not, if yes, then the condition becomes true **Ex:** (A = = B) ios not true.
!= (Not Equal)	Checks if tie value of two operands are equal or not, if the values are not equal, then the condition becomes true **Ex:** (A!= B) is true
> (Greater than)	Checks if the value of the left operand is greater than the value of operand, if yes, then the condition becomes true. **Ex:** (A > B) is not true.
< (Less than)	Checks if tie value of the ldt operand is less than the value of the right operand, if yes, then the condition becomes true. **Ex:**(A < B) is true.
> = (Greater than or Equal to)	Checks if tie value of the ldt operand is greater than or equal to the value of the right operand, if yes, then the conditcn becomes **Ex :** (A > = B) is not true.
< = (Less than or Eqial to)	Checks if tie value of theldt operand is less than or equal to the value of the right operand, if yes, then the conditcn becomes true. **Ex :** (A < = B) is true.

Example

```
<html>
 <body>
<script type="text/javascript">
<!—
var a = 10;
var b = 20;
var linebreak = "<br />";
document.write("(a = = b) => ");
result = (a = = b);
document.write(result);
document.write(linebreak);
document.write("(a < b) => ");
result = (a < b);
document.write(result);
document.write(linebreak);
document.write("(a > b) => ");
result = (a > b);
document.write(result);
document.write(linebreak);
document.write("(a != b) => ");
result = (a != b);
document.write(result);
document.write(linebreak);
```

```
document.write("(a >= b) => ");
result = (a >= b);
document.write(result);
document.write(linebreak);
document.write("(a <= b) => ");
result = (a <= b);
document.write(result);
document.write(linebreak);
//—>
</script>
```

Set the variables to different values and different operators and then try...

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

Output

(a == b) => false
(a < b) => true
(a > b) => false
(a != b) => true
(a >= b) => false
a <= b) => true
Set the variables to different values and different operators and then try...

(iii) Logical Operators

JavaScript supports the following logical operators-

Assume variable A holds 10 and variable B holds 20, then

Operator	Description
&& (Logical AND)	If both the operands are non-zero, then the condition becomes true. **Ex:** (A && B) is true.
|| (Logical OR)	If any of the two operands are non-zero, then the condition becomes true. **Ex:** (A 11 B) is true.
! (Logical NOT)	Reverses the logical state of its operand. If a condition is true, then the Logical NOT operator will make it false. **Ex:** ! (A && B) is false.

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var a = true;
var b = false;
var linebreak = "<br />";
document.write("(a && b) => ");
```

```
result = (a && b);
document.write(result);
document.write(linebreak);
document.write("(a || b) => ");
result = (a || b);
document.write(result);
document.write(linebreak);
document.write("!(a && b) => ");
result = (!(a && b));
document.write(result);
document.write(linebreak);
        //—>
    </script>
<p>Set the variables to different values and different operators and then try...</p>
        </body>
        </html>
```

Output

```
(a && b) => false
(a || b) => true
!(a && b) => true
Set the variables to different values and different operators and then try...
```

(iv) Bitwise Operators

JavaScript supports the following bitwise operators -

Assume variable A holds 2 and variable B holds 3, then -

Operator	Description
& (Bitwise AND)	It performs a Boolean AND operation on each bit of it interger arguments **Ex: (A & B) is 2.**
| (BitWise OR)	It performs a Boolean OR operation on each bit of it integer are arguments.
∧ (BitWise OR)	It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both **Ex:** (A ∧ B) is 1.
~ (Bitwise Not)	It is a unary operator and operates by reversing all the bits in the operand. **Ex:** (~B) s -4.
<< (Left Shift)	It moves all the bits in its first operand to the left by the number of pleces specified in the second. New bits are filled with zeros. Shifting a value left by one position t equivalent to multiplying it by 2, shipting two position is equivalent to multiplying by 4, and so on. **Ex:** (A<<1) = 4.
> > (Right Shift)	Binary Right Shift Operator. The left operand's value is moved right by the number of bits specified by the right operand. **Ex:** (A >> 1) is 1.
> > > (Right shift with Zero)	This operator is just like the > > operator, except that bits shifted in on the left are always zero. **Ex:** (A >>> 1) is 1.

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var a = 2; // Bit presentation 10
var b = 3; // Bit presentation 11
var linebreak = "<br />";
document.write("(a & b) => ");
result = (a & b);
document.write(result);
document.write(linebreak);
document.write("(a | b) => ");
result = (a | b);
document.write(result);
document.write(linebreak);
document.write("(a ^ b) => ");
result = (a ^ b);
document.write(result);
document.write(linebreak);
document.write("(~b) => ");
result = (~b);
document.write(result);
document.write(linebreak);
document.write("(a << b) => ");
result = (a << b);
document.write(result);
document.write(linebreak);
document.write("(a >> b) => ");
result = (a >> b);
document.write(result);
document.write(linebreak);
//—>
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
(a & b) => 2
(a | b) => 3
```

(a ^ b) = > 1

(~b) = > -4

(a < < b) = > 16

(a > > b) = > 0

Set the variables to different values and different operators and then try...

(v) Assignment Operators

JavaScript supports the following assignment operators -

Operator	Description
= (Simple Assignment)	Assigns values from the right side operand to the leftside operand **Ex:** C = A + B will assign the value of A + B into C
+ = (Add and Assignment)	It adds the right operand to the left operand and assigns the result to the left operand. **Ex:** C + = A is equivalent to C = C + A
– = (Subtract and Assignment)	It subtracts the right operand from the operand and assign the result to the left operand. **Ex:** C – = A is equivalent to C = C – A
***= (Multiply and Assignment)**	It multiplies the right operand with the left operand and assign the result to the left operand. **Ex:** C *= A is equivalent to C = C*A
/= (Divide and Assignment)	It divides the left operand with the right operand and assigns the result to the left operand. **Ex:** C /= A is equivalent to C = C / A
% = (Modules and Assignment)	It takes modulus using two operands and assigns the result to the left operand **Ex:** C %= A is equivalent to C = C % A

Note - Same logic applies to Bitwise operators so they will become like < < =, > > =, > > =, &=, |= and ^ =.

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var a = 33;
var b = 10;
var linebreak = "<br />";
document.write
("Value of a => (a = b) => ");
result = (a = b);
document.write(result);
document.write(linebreak);
document.write
("Value of a => (a += b) => ");
```

```
result = (a += b);
document.write(result);
document.write(linebreak);
document.write
("Value of a => (a -= b) => ");
result = (a -= b);
document.write(result);
document.write(linebreak);
document.write
("Value of a => (a *= b) => ");
result = (a *= b);
document.write(result);
document.write(linebreak);
document.write
("Value of a => (a /= b) => ");
result = (a /= b);
document.write(result);
document.write(linebreak);
document.write
("Value of a => (a %= b) => ");
result = (a %= b);
document.write(result);
document.write(linebreak);
//—>
</script>
<p>Set the variables to different values and
different operators and then try...</p>
</body>
</html>
```

Output

Value of a => (a = b) => 10

Value of a => (a += b) => 20

Value of a => (a -= b) => 10

Value of a => (a *= b) => 100

Value of a => (a /= b) => 10

Value of a => (a %= b) => 0

Set the variables to different values and different operators and then try...

(vi) Conditional Operator (? :)

The conditional operator first evaluates an expression for a true or false value and then executes one of the two given statements depending upon the result of the evaluation.

Operator	Description
? : (Conditional	Of Condition is true? Then value X Otherwise value Y

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var a = 10;
var b = 20;
var linebreak = "<br />";
document.write ("((a > b) ? 100 : 200) => ");
result = (a > b) ? 100 : 200;
document.write(result);
document.write(linebreak);
document.write ("((a < b) ? 100 : 200) => ");
result = (a < b) ? 100 : 200;
document.write(result);
document.write(linebreak);
//—>
</script>
<p>Set the variables to different values and
different operators and then try...</p>
</body>
</html>
```

Output

((a > b) ? 100 : 200) => 200

((a < b) ? 100 : 200) => 100

Set the variables to different values and different operators and then try...

type of Operator

The typeof operator is a unary operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.

The typeof operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.

Here is a list of the return values for the typeof Operator.

Type	String Returned by typeof
Number	"number"
String	"string"
Boolean	"boolean"
Object	"object"
Function	"function"
Undefined	"undefined"
Null	"object"

Example

The following code shows how to implement typeof operator.

```
<html>
<body>
<script type="text/javascript">
<!—
var a = 10;
var b = "String";
var linebreak = "<br />";
result = (typeof b = = "string" ? "B is String" : "B is Numeric");
document.write("Result = > ");
document.write(result);
document.write(linebreak);
result = (typeof a = = "string" ? "A is String" : "A is Numeric");
document.write("Result = > ");
document.write(result);
document.write(linebreak);
//—>
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
```

Output

Result = > B is String

Result = > A is Numeric

Set the variables to different values and different operators and then try..

3.7 EXPRESSIONS AND STATEMENTS

Q13. Define expression. List the various types of expressions in JavaScript.

Ans : **(Imp.)**

An expression consists of a single entity (such as a constant, a variable, an array element or a reference to a function) or combination of such entities are joined together using one or more operators.

An expression can also represent logical condition, that is either true or false, which can be represented using integer values like '1' and '0' respectively.

In general, there exist two types of variables one is expression that assigns a value to the given variable and expression that contains some value.

Types of Expressions

JavaScript supports the following types of expressions,

1. Arithmetic expressions
2. Logical expressions
3. Conditional expressions
4. String expressions.

1. Arithmetic Expression

An expression consisting of variables, numbers or function calls that are combined together by arithmetic operators is known as 'Arithmetic Expression'. . .

Example

(a - b) * (a + b)/4.

2. Logical Expression

An expression combining two or more expressions is known as 'logical expression'. It evaluates the expressions into true or false.

Example

(x>y&&a = =l)

3. Conditional Expression

An expression used in evaluating the statements based on the provided conditions is known as "Conditional Expression".

Example

X = (2 > 1>? "Yes" : "No".

4. String Expression

A string expression evaluates to a character string.

Example

"abc", "xyz".

Q14. How java script supports conditional statements? Explain with example programs.

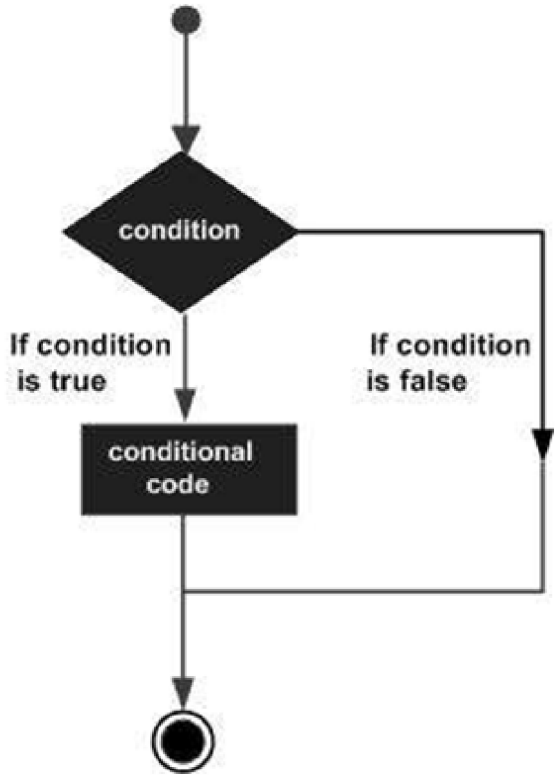
Ans : **(Imp.)**

While writing a program, there may be a situation when you need to adopt one out of a given set of paths. In such cases, you need to use conditional statements that allow your program to make correct decisions and perform right actions.

JavaScript supports conditional statements which are used to perform different actions based on different conditions. Here we will explain the if..else statement.

Flow Chart of if-else

The following flow chart shows how the if-else statement works.



JavaScript supports the following forms of if..else statement -

(i) if statement

(ii) if...else statement

(ii) if...else if... statement.

(i) if statement

The if statement is the fundamental control statement that allows JavaScript to make decisions and execute statements conditionally.

Syntax

The syntax for a basic if statement is as follows -

```
if (expression){
Statement(s) to be executed if
expression is true
}
```

Here a JavaScript expression is evaluated. If the resulting value is true, the given statement(s) are executed. If the expression is false, then no statement would be not executed. Most of the times, you will use comparison operators while making decisions.

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var age = 20;
if( age > 18 ){
document.write("<b>Qualifies for
driving</b>");
}
//—>
</script>
<p>Set the variable to different value and
then try...</p>
</body>
</html>
```

Output

Qualifies for driving

Set the variable to different value and then try...

(ii) if...else statement:

The 'if...else' statement is the next form of control statement that allows JavaScript to execute statements in a more controlled way.

Syntax

```
if (expression){
Statement(s) to be executed if
expression is true
}
else{
Statement(s) to be executed
if expression is false
}
```

Here JavaScript expression is evaluated. If the resulting value is true, the given statement(s) in the 'if' block, are executed. If the expression is false, then the given statement(s) in the else block are executed.

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var age = 15;
if( age > 18 ){
document.write("<b>Qualifies for
driving</b>");
}
else{
document.write("<b>Does not qualify for
driving</b>");
}
//—>
```

```
</script>
<p>Set the variable to different value and
then try...</p>
</body>
</html>
```

Output

Does not qualify for driving

Set the variable to different value and then try...

(iii) if...else if... statement

The if...else if... statement is an advanced form of if…else that allows JavaScript to make a correct decision out of several conditions.

Syntax

The syntax of an if-else-if statement is as follows "

```
if (expression 1){
Statement(s) to be executed if expression 1
is true
}
else if (expression 2){
Statement(s) to be executed if
expression 2 is true
}
else if (expression 3){
Statement(s) to be executed if
expression 3 is true
}
else{
Statement(s) to be executed if no
expression is true
}
```

There is nothing special about this code. It is just a series of if statements, where each if is a part of the else clause of the previous statement. Statement(s) are executed based on the true condition, if none of the conditions is true, then the else block is executed.

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var book = "maths";
if( book == "history" ){
document.write("<b>History Book</b>");
}
else if( book == "maths" ){
document.write("<b>Maths Book</b>");
}
else if( book == "economics" ){
document.write("<b>Economics Book
</b>");
}
else{
document.write("<b>Unknown Book</b>");
}
//—>
</script>
<p>Set the variable to different value and
then try...</p>
</body>
<html>
```

Output

Maths Book

Set the variable to different value and then try...

Q15. Explain about switch statement and Do... while loop in java script.

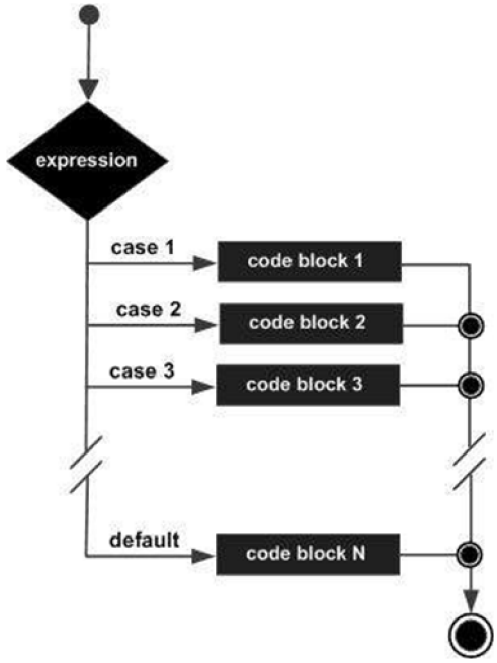
Ans : **(July.-19, Imp.)**

(i) Switch Statement

Switch statement which handles exactly this situation, and it does so more efficiently than repeated if...else if statements.

Flow Chart

The following flow chart explains a switch-case statement works.



Syntax

The syntax of while loop in JavaScript is as follows -

```
while (expression){
   Statement(s) to be executed if
    expression is true
}
```

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var count = 0;
document.write("Starting Loop ");
while (count < 10){
document.write("Current Count :
" + count + "<br />");
count++;
}
document.write("Loop stopped!");
//—>
</script>
<p>Set the variable to different value and
then try...</p>
</body>
</html>
```

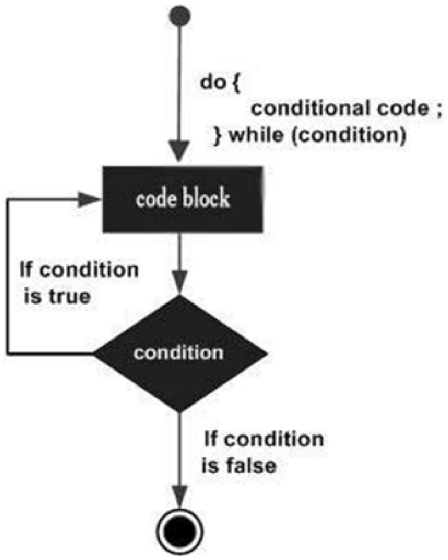
Output

```
Starting Loop
Current Count : 0
Current Count : 1
Current Count : 2
Current Count : 3
Current Count : 4
Current Count : 5
Current Count : 6
Current Count : 7
Current Count : 8
Current Count : 9
Loop stopped!
Set the variable to different value and then
try...
```

(ii) The do...while Loop

The do...while loop is similar to the while loop except that the condition check happens at the end of the loop. This means that the loop will always be executed at least once, even if the condition is false.

Flow Chart. The flow chart of a do-while loop would be as follows -



Syntax

The syntax for do-while loop in JavaScript is as follows -

```
do{
Statement(s) to be executed;
}  while (expression);
```

Note- Don't miss the semicolon used at the end of the do...while loop.

Example

```
<html>
<body>
<script type="text/javascript">
<!—
var count = 0;
document.write("Starting Loop"
+ "<br />");
do{
document.write("Current Count :
" + count + "<br />");
count++;
}
while (count < 5);
document.write ("Loop stopped!");
//—>
</script>
<p>Set the variable to different value and then try...</p>
</body>
</html>
```

Output

Starting Loop

Current Count : 0

Current Count : 1

Current Count : 2

Current Count : 3

Current Count : 4

Loop Stopped!

Set the variable to different value and then try...

Q16. Explain for-loop in java script with an example.

Ans : **(Imp.)**

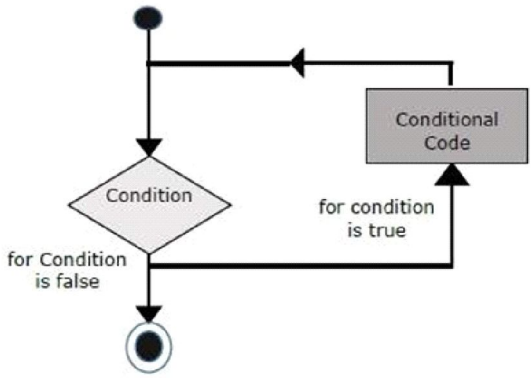
The 'for' loop is the most compact form of looping. It includes the following three important parts -

- The loop initialization where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
- The test statement which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.
- The iteration statement where you can increase or decrease your counter.

You can put all the three parts in a single line separated by semicolons.

Flow Chart

The flow chart of a for loop in JavaScript would be as follows-



Syntax

The syntax of for loop is JavaScript is as follows-

```
for (initialization; test condition; iteration statement){
Statement(s) to be executed if test condition is true
}
```

Example

Try the following example to learn how a for loop works in JavaScript.

```
<html>
<body>
<script type="text/javascript">
<!—
var count;
document.write("Starting Loop" + "<br />");
for(count = 0; count < 10; count++){
document.write("Current Count : " + count );
document.write("<br />");
}
document.write("Loop stopped!");
//—>
</script>
<p>Set the variable to different value and then try...</p>
</body>
</html>
```

Output

```
Starting Loop
Current Count : 0
Current Count : 1
Current Count : 2
Current Count : 3
Current Count : 4
Current Count : 5
Current Count : 6
Current Count : 7
Current Count : 8
Current Count : 9
Loop stopped!
Set the variable to different value and then try...
```

3.8 FUNCTIONS

Q17. How java script supports functions? Explain in detail?

(or)

Explain in detail how Javascript Supports function.

Ans : **(Dec.-20, Dec.-18, Imp.)**

A function is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions.

Like any other advanced programming language, JavaScript also supports all the features necessary to write modular code using functions. You must have seen functions like alert() and write() in the earlier chapters. We were using these functions again and again, but they had been written in core JavaScript only once.

JavaScript allows us to write our own functions as well.

(i) Function Definition

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

Syntax

The basic syntax is shown here.

```
<script type="text/javascript">
<!—
function functionname(parameter-list)
{
statements
}
//—>
</script>
```

Example

Try the following example. It defines a function called sayHello that takes no parameters -

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

(ii) Calling a Function

To invoke a function somewhere later in the script, you would simply need to write the name of that function as shown in the following code.

```
<html>
<head>
<script type="text/javascript">
function sayHello()
{
document.write ("Hello there!");
}
</script>
</head>
<body>
<p>Click the following button to call the function</p>
<form>
<input type="button" onclick="sayHello()" value="Say Hello">
</form>
<p>Use different text in write method and then try...</p>
</body>
</html>
```

(iii) Function Parameters

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

Example

Try the following example. We have modified our sayHello function here. Now it takes two parameters.

```
<html>
<head>
<script type="text/javascript">
function sayHello(name, age)
{
document.write (name + " is " + age + " years old.");
}
</script>
</head>
<body>
<p>Click the following button to call the function</p>
<form>
<input type="button" onclick="sayHello('Zara', 7)" value="Say Hello">
</form>
<p>Use different parameters inside the function and then try...</p>
</body>
</html>
```

(iv) The return Statement

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

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

Example

Try the following example. It defines a function that takes two parameters and concatenates them before returning the resultant in the calling program.

```
<html>
<head>
<script type="text/javascript">
function concatenate(first, last)
{
var full;
full = first + last;
return full;
}
function secondFunction()
{
var result;
```

```
        result = concatenate('Zara', 'Ali');
        document.write (result );
}
                </script>
                </head>
                <body>
<p>Click the following button to call the function</p>
                <form>
<input type="button" onclick =" second Function()" value="Call Function">
        </form>
<p>Use different parameters inside the function and then try...</p>
        </body>
        </html>
```

3.9 OBJECTS

Q18. What is meant by object? What kind of objects supports by java script?

Ans :

JavaScript is an Object Oriented Programming (OOP) language. A programming language can be called object-oriented if it provides four basic capabilities to developers -

- **Encapsulation** - the capability to store related information, whether data or methods, together in an object.
- **Aggregation** - the capability to store one object inside another object.
- **Inheritance** - the capability of a class to rely upon another class (or number of classes) for some of its properties and methods.
- **Polymorphism** - the capability to write one function or method that works in a variety of different ways.

Objects are composed of attributes. If an attribute contains a function, it is considered to be a method of the object, otherwise the attribute is considered a property.

Object Properties

Object properties can be any of the three primitive data types, or any of the abstract data types, such as another object. Object properties are usually variables that are used internally in the object's methods, but can also be globally visible variables that are used throughout the page.

The syntax for adding a property to an object is -

object Name.object Property = property Value;

For example - The following code gets the document title using the "title" property of the document object.

```
var str = document.title;
```

Object Methods

Methods are the functions that let the object do something or let something be done to it. There is a small difference between a function and a method – at a function is a standalone unit of statements and a method is attached to an object and can be referenced by the this keyword.

Methods are useful for everything from displaying the contents of the object to the screen to performing complex mathematical operations on a group of local properties and parameters.

For example - Following is a simple example to show how to use the write() method of document object to write any content on the document.

```
document.write("This is test");
```

User-Defined Objects

All user-defined objects and built-in objects are descendants of an object called Object.

The new Operator

The new operator is used to create an instance of an object. To create an object, the new operator is followed by the constructor method.

In the following example, the constructor methods are Object(), Array(), and Date(). These constructors are built-in JavaScript functions.

```
var employee = new Object();
```

```
var books = new Array("C++", "Perl",
 "Java");
var day = new Date("August 15, 1947");
```

The Object() Constructor

A constructor is a function that creates and initializes an object. JavaScript provides a special constructor function called Object() to build the object. The return value of the Object() constructor is assigned to a variable.

The variable contains a reference to the new object. The properties assigned to the object are not variables and are not defined with the var keyword.

```
<html>
<head>
<title>User-defined objects</title>
<script type="text/javascript">
var book = new Object();  // Create the object
book.subject = "Perl"; // Assign properties to the object
book.author  = "Mohtashim";
</script>
</head>
<body>
<script type="text/javascript">
document.write("Book name is :
 " + book.subject + "<br>");
document.write("Book author is :
 " + book.author + "<br>");
</script>
</body>
</html>
```

Output

```
Book name is : Perl
Book author is : Mohtashim
```

Q19. What is built in objects? Explain how many built-in objects supports by javascript?

Ans :

Java Script has several built-in or native objects. These objects are accessible anywhere in your program and will work the same way in any browser running in any operating system.

Here is the list of all important JavaScript Native Objects -

1. JavaScript Number Object
2. JavaScript Boolean Object
3. JavaScript String Object

4. JavaScript Date Object

5. JavaScript Math Object

1. JavaScript Number Object

The Number object represents numerical date, either integers or floating-point numbers. In general, you do not need to worry about Number objects because the browser automatically converts number literals to instances of the number class.

Syntax

The syntax for creating a number object is as follows -

```
var val = new Number(number);
```

In the place of number, if you provide any non-number argument, then the argument cannot be converted into a number, it returns NaN (Not-a-Number).

Number Properties

Here is a list of each property and their description.

Property	Description
MAX_VALUE	The largest possible value a number in JavaScript can have 1.7976931348623157E + 308.
MIN_VALUE	The smallest possible value a number in JavaScript can have 5E-324.
NaN	Equal to a value that is not a number.
NEGATIVE_INFINITY	A value that is less than MIN_VALUE
POSITIVE_INFINITY	A value that is greater than MAX_VALUE
Prototype	A stalic property of the Number object. Use the prototype property to assign new properties and methods to the Number object in the current document.
constructor	Returns the function that created this objects's By default this is the Number object

2. JavaScript Boolean Object

The Boolean object represents two values, either "true" or "false". If *value* parameter is omitted or is 0, -0, null, false, NaN, undefined, or the empty string (""), the object has an initial value of false.

Syntax

Use the following syntax to create a boolean object.

```
var val = new Boolean(value);
```

Boolean Properties

Here is a list of the properties of Boolean object

Property	Description
constructor	Returns a reference to the Boolean function that created the object.
prototype	The prototype property allows you to add properties and methods an object.

In the following sections, we will have a few examples to illustrate the properties of Boolean object.

Boolean Methods

Here is a list of the methods of Boolean object and their description.

Method	Description
toScource	Returns a string containing the source of the Boolean object; you can use this string to create and equivalent object.
toString()	Returns a string of either "true or false" depending upon the value of the object.
value Of()	Returns the primitive value of the Boolean object.

In the following sections, we will have a few examples to demonstrate the usage of the Boolean methods.

3. JavaScript Strings Object

The String object lets you work with a series of characters; it wraps Javascript's string primitive data type with a number of helper methods.

As JavaScript automatically converts between string primitives and String objects, you can call any of the helper methods of the String object on a string primitive.

Syntax

Use the following syntax to create a String object -

```
var val = new String(string);
```

The String parameter is a series of characters that has been properly encoded.

String Properties

Here is a list of the properties of String object and their description.

Property	Description
constructor	Returns a reference to the String function that created the object.
length	returns the length of the string.
prototype	The prototype property allows you to add properties and methods to an object.

In the following sections, we will have a few examples to demonstrate the usage of String properties.

String Methods

Here is a list of the methods available in String object along with their description.

Method	Description
charAt()	Returns the character at the specified index.
charCodeAt()	Returns a number indicating the Unicode value of the character at the given index.
concat()	Combines the text of two strings and returns a new string.
indexOf()	Returns the index within the calling String object of the first occurrence of the specified value, or -1 if not found.
lastIndexOf()	Returns the index within the calling String object of the last occurrence of the specified value, or -1 if not found.
localeCompare()	Returns a number indicating whether a reference string comes before or after or is the same as the given string in sort order.
match()	Used to match a regular expression against a string.
replace()	Used to find a match between a regular expression and a sting, and to replace the matched substring with a new substring.
search()	Executes the search for a match between a regular expression and a specified string.
slice()	Extracts a section of a string and returns a new string.
split()	Splits a String object into an array at strings by separating the string into substrings.
substr()	Returns the characters in a string beginning at the specified location through the specified number of characters.
substring()	Returns the characters in a string between two indexes into the string.
toLocaleLowerCase()	The characters within a string are converted to lower case while respecting the current locale.
toLocaleUpperCase()	The characters within a string are converted to upper case while respecting the current locale.
toLowerCase()	Returns the calling string value converted to lower case.
toString()	Returns a string representing the specified object.
toUpperCase()	Returns the calling string value converted to uppercase
valueOf()	Returns the primitive value at the specified object.

4. Java script date object

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

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

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

Syntax

You can use any of the following syntaxes to create a Date object using Date() constructor.

new Date( )

new Date(milliseconds)

new Date(datestring)

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

Note - Parameters in the brackets are always optional.

Here is a description of the parameters -

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

Date Properties

Here is a list of the properties of the Date object along with their description.

Property	Description
constructor	Specifies the function that create an objects's prototype. prototype The prototype property allows you to add properties and methods to an object.

5. JavaScript Math Object

The math object provides you properties and methods for mathematical constants and functions. Unlike other global objects, Math is not a constructor. All the properties and methods of Math are static and can be called by using Math as an object without creating it.

Thus, you refer to the constant pi as Math.PI and you call the *sine* function as Math.sin(x), where x is the method's argument.

Syntax

The syntax to call the properties and methods of Math are as follows

```
var pi_val = Math.PI;
var sine_val = Math.sin(30);
```

Math Properties

Here is a list of all the properties of Math and their description.

Property	Description
E	Euler's constant and the base of naturall logarithms, approximately 2.718.
LN2	Natural logarithm of 2, approximately 0.693.
LN10	Natural logarithm of 10, approximately 2.302.
LOG2E	Base 2 logarithm of E, approximately 1.442.
LOG10E	Base 10 logarithm of E, approximately 0.434.
PI	Ratio of the circumference of a circle to its diameter, approximately 3.14159.
SQRT1_2	Square root of 1/2; equivalently, 1 ov ert the square root of 2, approximate 0.707.
SQRT2	Square root of 2, approximately 1.414.

In the following sections, we will have a few examples to demonstrate the usage of Math properties.

3.10 ARRAYS

Q20. What is an array ? How java script implements arrays ?

Ans :

The Array object lets you store multiple values in a single variable. It stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Syntax

Use the following syntax to create an Array object -

var fruits = new Array(“apple”, “orange”, “mango”);

The Array parameter is a list of strings or integers. When you specify a single numeric parameter with the Array constructor, you specify the initial length of the array. The maximum length allowed for an array is 4,294,967,295.

You can create array by simply assigning values as follows “

var fruits = [“apple”, “orange”, “mango”];

You will use ordinal numbers to access and to set values inside an array as follows.

fruits[0] is the first element

fruits[1] is the second element

fruits[2] is the third element

Array Properties

Here is a list of the properties of the Array object along with their description.

Property	Description
constructor	Returns a reference to the array function that created the object.
index	The property represents the zero-based index of the match in the string
input	This property is only present in arrays created by regular expression matches.
length	Reflects the number of elements in an array.
prototype	The prototype property allows you to add properties and methods to an object.

In the following sections, we will have a few examples to illustrate the usage of Array properties.

3.11 DOCUMENT OBJECT MODEL

Q21. What is Document object model?

Ans : **(Imp.)**

The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

“The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.”

The W3C DOM standard is separated into 3 different parts:

- Core DOM - standard model for all document types
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

Q22. What is the HTML DOM?

Ans :

The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

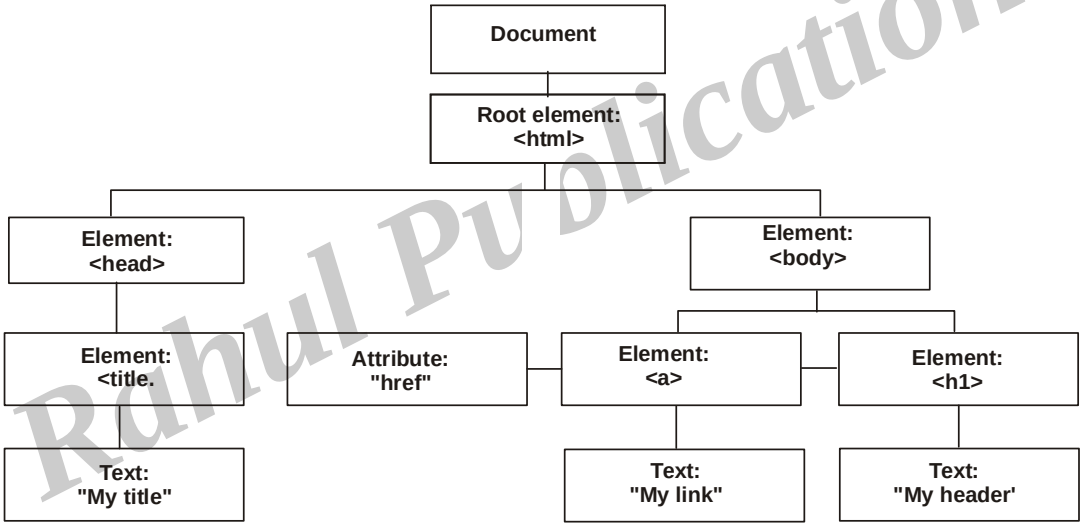
In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

The HTML DOM (Document Object Model)

When a web page is loaded, the browser creates a Document Object Model of the page.

The HTML DOM model is constructed as a tree of Objects:

The HTML DOM Tree of Objects



With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

3.12 Date Math Related Objects

Q23. Explain Math and Date objects in java script ?

Ans :

1. Math object

The math object provides you properties and methods for mathematical constants and functions. Unlike other global objects, Math is not a constructor. All the properties and methods of Math are static and can be called by using Math as an object without creating it.

Thus, you refer to the constant pi as Math.PI and you call the *sine* function as Math.sin(x), where x is the method's argument.

Syntax

The syntax to call the properties and methods of Math are as follows

```
var pi_val = Math.PI;
var sine_val = Math.sin(30);
```

Math Properties

Here is a list of all the properties of Math and their description.

Property	Description
E	Euler's constant and the base of natural logarithms, approximately 2.718.
LN2	Natural logarithm of 2, approximately 0.693.
LN10	Natural logarithm of 10, approximately 2.302.
LOG2E	Base 2 logarithm of E, approximately 1.442.
LOG10E	Base 10 logarithm of E, approximately 0.434.
PI	Ratio of the circumference of a circle to its diameter, approximately 3.14159.
SORT1_2	Square root of 1/2; equivalently, 1 over the square root of 2, approximately 0.707.
SORT2	Square root of 2. approximately 1.414.

In the following sections, we will have a few examples to demonstrate the usage of Math properties.

Math Methods

Here is a list of the methods associated with Math object and their description.

Method	Description
abs()	Returns the absolute value of a number.
acos()	Retums'ihe arccosine (in radians) of a number.
asin()	Returns the arcsine (in radians) of a number.
atan()	Returns the arctangent (in radians) of a number.
atan2()	Returns the arctangent of the quotient of its arguments.
ceil()	Returns the smallest integer greater than or equal to a number.
cos()	Returns the cosine of a number.
exp()	Returns E^N, where N is the argument, and E is Euler's constant, the base of the natural logarithm.
floor()	Returns the largest integer less than or equal to a number.
log()	Returns the natural logarithm (base E) of a number.
max()	Returns the largest of zero or more numbers.
min()	Returns the smallest of zero or more numbers.
pow()	Returns base to the exponent power, that is, base exponent
random()	Returns a pseudo-random number between 0 and 1.
round()	Returns the value of a number rounded to the nearest integer.
sin()	Returns the sine of a number.
sqrt()	Returns the square root of a number.
tan()	Returns the tangent of a number.
toSoorce()	Returns the string "Math".

In the following sections, we will have a few examples to demonstrate the usage of the methods associated with Math.

2. Date object

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

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

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

Syntax

You can use any of the following syntaxes to create a Date object using Date() constructor.

new Date( )

new Date(milliseconds)

new Date(datestring)

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

Note - Parameters in the brackets are always optional.

Here is a description of the parameters -

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

Date Properties

Here is a list of the properties of the Date object along with their description.

Property	Description
constructor	Specifies the function that creates an object's prototype.
prototype	The prototype property allows you to add properties and methods to an object

In the following sections, we will have a few examples to demonstrate the usage of different Date properties.

Date Methods

Here is a list of the methods used with Date and their description.

Method	Description
Date()	Returns today's date and time
getDate()	Returns the day of the month for the specified date according to local time.
getDay()	Returns the day of the week for the specified date according to local time.
getFullYear()	Returns the year of the specified date according to local time.
getHours()	Returns the hour in the specified date according to local time.
getMilliseconds()	Returns the milliseconds in the specified date according to local time.
getMinutes()	Returns the minutes in the specified date according to local time.
getMonth()	Returns the month in the specified date according to local time.
getSeconds()	Returns the seconds in the specified date according to local time.
getTime()	Returns the numeric value of the specified date as the number of milliseconds since January 1, 1970, 00:00:00 UTC.
getTimezoneOffset()	Returns the time-zone offset in minutes for the current locale.
getUTCDate()	Returns the day (date) of the month in the specified date according to universal time.
getUTCDay()	Returns the day of the week in the specified date according to universal time.
getUTCFullYear()	Returns the year in the specified date according to universal time.
getUTCHours()	Returns the hours in the specified date according to universal time.
getUTCMilliseconds()	Returns the milliseconds in the specified date according to universal time.
getUTCMinutes()	Returns the minutes in the specified date according to universal time.
getUTCMonth()	Returns the month in the specified date according to universal time.
getUTCSeconds()	Returns the seconds in the specified date according to universal time.
getYear()	Deprecated - Returns the year in the specified date according to local time. Use getFullYear instead.
setDate()	Sets the day of the month for a specified date according to local time.

setFullYear()	Sets the full year for a specified date according to local time.
setHours()	Sets the hours for a specified date according to local time.
setMilliseconds()	Sets the milliseconds for a specified date according to local time.
setMinutes()	Sets the minutes for a specified date according to local time.
setMonth()	Sets the month for a specified date according to local time.
setSeconds()	Sets the seconds for a specified date according to local time.
setTime()	Sets the Date object to the time represented by a number of milliseconds since January 1, 1970, 00:00:00 UTC.
setUTCDate()	Sets the day of the month for a specified date according to universal time.
setUTCFullYear()	Sets the full year for a specified date according to universal time.
setUTCHours()	Sets the hour for a specified date according to universal time.
setUTCMilliseconds()	Sets the milliseconds for a specified date according to universal time.
setUTCMinutes()	Sets the minutes for a specified date according to universal time.
setUTCMonth()	Sets the month for a specified date according to universal time.
setUTCSeconds()	Sets the seconds for a specified date according to universal time.
setYear()	Deprecated - Sets the year for a specified date according to local time. Use setFullYear instead.
toDateString()	Returns the “date” portion of the Date as a human-readable string.
toGMTString()	Deprecated - Converts a date to a string, using the Internet GMT conventions. Use toUTCString instead.
toLocaleDateString()	Returns the “date” portion of the Date as a string, using the current locale’s conventions.
toLocaleFormat()	Converts a date to a string, using a format string.
toLocaleString()	Converts a date to a string, using the current locale’s conventions.
toLocaleTimeString()	Returns the “time” portion of the Date as a string, using the current locale’s conventions.
toSource()	Returns a string representing the source for an equivalent Date object; you can use this value to create a new object.
toString()	Returns a string representing the specified Date object.
toTimeString()	Returns the “time” portion of the Date as a human-readable string.
toUTCString()	Converts a date to a string, using the universal time convention.
valueOf()	Returns the primitive value of a Date object.

Converts a date to a string, using the universal time convention.

Date Static Methods

In addition to the many instance methods listed previously, the Date object also defines two static methods. These methods are invoked through the Date() constructor itself.

Object Numbers	Method & Description
Date.parse( )	Parses a string representation of a date and time and returns the internal millisecond representation of that date.
Date.UTC( )	Returns the millisecond representation of the specified UTC date and time.

In the following sections, we will have a few examples to demonstrate the usages of Date Static methods.

3.13 EVENT HANDLING

Q24. What is an Event ? Explain event handling in Javascript ?

Ans : **(Imp.)**

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.

When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.

Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.

Please go through this small tutorial for a better understanding HTML Event Reference. Here we will see a few examples to understand a relation between Event and JavaScript-

onclick Event Type

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.

Example

```
<html>
<head>
<script type="text/javascript">
<!—
function sayHello() {
alert("Hello World")
}
```

```
//—>
</script>
</head>
<body>
<p>Click the following button and see result</p>
<form>
<inputtype="button" onclick=" sayHello()" value="Say Hello" />
</form>
</body>
</html>
```

on submit Event type

onsubmit is an event that occurs when you try to submit a form. You can put your form validation against this event type.

Example

The following example shows how to use onsubmit. Here we are calling a validate() function before submitting a form data to the webserver. If validate() function returns true, the form will be submitted, otherwise it will not submit the data.

Try the following example.

```
<html>
<head>
<script type="text/javascript">
<!—
function validation() {
all validation goes here
.....
return either true or false
}
//—>
</script>
</head>
<body>
<form method="POST" action="t.cgi" onsubmit="return validate()">
.....
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

on mouse over and on mouse out

These two event types will help you create nice effects with images or even with text as well. The onmouseover event triggers when you bring your mouse over any element and the onmouseout triggers when you move your mouse out from that element. Try the following example.

```
<html>
<head>
<script type="text/javascript">
<!—
function over() {
document.write ("Mouse Over");
}
function out() {
document.write ("Mouse Out");
}
//—>
</script>
</head>
<body>
<p>Bring your mouse inside the division to see the result:</p>
<div onmouseover="over()" onmouse out="out()">
<h2> This is inside the division </h2>
</div>
</body>
</html>
```

Attribute	Value	Description
Offline	script	Triggers when the document goes offline
Onabort	script	Triggers on an abort event
onafterprint	script	Triggers after the document is printed
onbeforeonload	script	Triggers before the document loads
onbeforeprint	script	Triggers before the document is printed
onblur	script	Triggers when the window loses focus
oncanplay	script	Triggers when media can start play, but might has to stop for buffering
oncanplaythrough	script	Triggers when media can be played to the end, without stopping for buffering
onchange	script	Triggers when an element changes
onclick	script	Triggers on a mouse click
oncontextmenu	script	Triggers when a context menu is triggered
ondblclick	script	Triggers on a mouse double-click
ondrag	script	Triggers when an element is dragged
ondragend	script	Triggers at the end of a drag operation
ondragenter	script	Triggers when an element has been dragged to a valid drop target
ondragleave	script	Triggers when an element is being dragged over a valid drop target
ondragover	script	Triggers at the start of a drag operation
ondragstart	script	Triggers at the start of a drag operation
ondrop	script	Triggers when dragged element is being dropped
ondurationchange	script	Triggers when the length of the media is changed
onemptied	script	Triggers when a media resource element suddenly becomes empty.
onended	script	Triggers when media has reach the end
onerror	script	Triggers when an error occur
onfocus	script	Triggers when the window gets focus
onformchange	script	Triggers when a form changes
onforminput	script	Triggers when a form gets user input
onhaschange	script	Triggers when the document has change
oninput	script	Triggers when an element gets user input
oninvalid	script	Triggers when an element is invalid
onkeydown	script	Triggers when a key is pressed
onkeypress	script	Triggers when a key is pressed and released
onkeyup	script	Triggers when a key is released
onload	script	Triggers when the document loads
onloadeddata	script	Triggers when media data is loaded
onloadedmetadata	script	Triggers when the duration and other media data of a media element is loaded

Short Question and Answers

1. List out the advantages of Java Script.

Ans :

The merits of using JavaScript are -

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

2. What is client side scripting.

Ans :

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

3. What is server side java scripting.

Ans :

Although rich web sites and applications have become the norm in recent years, they can still be difficult to develop, maintain, and expand. Many of the challenges stem from the schism between client and server components. Client-side components usually consist of HTML, CSS, JavaScript, Ajax, JS libraries, images, and whatever other files that are to be downloaded to the browser. On the server, you need a listener to process requests, fetch resources or information, and manipulate them so that they can be sent back to the client.

This is usually accomplished using XML, JSON, or HTML-Formatted text, which is sent across the wire using Ajax. There are a number of competing technologies to choose from here. Depending on your traffic, hardware, O/S, bandwidth, IT expertise, and numerous other factors, there is a technology for every taste and occasion. Popular server-side languages at this time include PHP, Java, and .NET, to name only a few.

There is presently a ServerJS movement whose goal is to eliminate the gap between client and server. Exponents of the group want to keep with HTML, JavaScript, and CSS, which are most familiar to the end-users. What makes server-side JavaScript possible is a web server than can process the code.

One such server is called Jaxer. Developed by Aptana, Jaxer is an open source Ajax web server for building rich web pages and applications using a unified Ajax model that can be written entirely using JavaScript. Writing code for Jaxer is the focus of this article.

4. Explain different types of variables.

Ans :

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

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

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

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

OUTPUT :

Local

5. List and Explain Bitwise Operators in Java.

Ans :

Bitwise Operators

JavaScript supports the following bitwise operators -

Assume variable A holds 2 and variable B holds 3, then -

Operator	Description
& (Bitwise AND)	It performs a Boolean AND operation on each bit of it interger arguments **Ex: (A & B) is 2.**
| (BitWise OR)	It performs a Boolean OR operation on each bit of it integer are arguments.
^ (BitWise OR)	It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both **Ex:** (A ^ B) is 1.
~ (Bitwise Not)	It is a unary operator and operates by reversing all the bits in the operand. **Ex:** (~B) s -4.
<< (Left Shift)	It moves all the bits in its first operand to the left by the number of pleces specified in the second. New bits are filled with zeros. Shifting a value left by one position t equivalent to multiplying it by 2, shipting two position is equivalent to multiplying by 4, and so on. **Ex:** (A<<1) = 4.
> > (Right Shift)	Binary Right Shift Operator. The left operand's value is moved right by the number of bits specified by the right operand. **Ex:** (A >> 1) is 1.
> > > (Right shift with Zero)	This operator is just like the > > operator, except that bits shifted in on the left are always zero. **Ex:** (A >>> 1) is 1.

6. Explain about Conditional Operator ?

Ans :

The conditional operator first evaluates an expression for a true or false value and then executes one of the two given statements depending upon the result of the evaluation.

Operator	Description
? : (Conditional	Of Condition is true? Then value X Otherwise value Y

Example

Try the following code to understand how the Conditional Operator works in JavaScript.

```
<html>
<body>
<script type="text/javascript">
<!—
var a = 10;
var b = 20;
var linebreak = "<br />";
document.write ("((a > b) ? 100 : 200) => ");
result = (a > b) ? 100 : 200;
document.write(result);
document.write(linebreak);
document.write ("((a < b) ? 100 : 200) => ");
result = (a < b) ? 100 : 200;
document.write(result);
document.write(linebreak);
//—>
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
```

Output

((a > b) ? 100 : 200) => 200

((a < b) ? 100 : 200) => 100

Set the variables to different values and different operators and then try...

7. Explain for-loop in java script with example.

Ans :

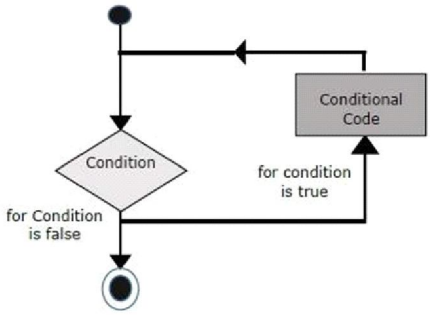
The 'for' loop is the most compact form of looping. It includes the following three important parts -

- The loop initialization where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
- The test statement which will test if a given condition is true or not. If the condition is true, then the code given inside the loop will be executed, otherwise the control will come out of the loop.
- The iteration statement where you can increase or decrease your counter.

You can put all the three parts in a single line separated by semicolons.

Flow Chart

The flow chart of a for loop in JavaScript would be as follows-



Syntax

The syntax of for loop is JavaScript is as follows-

```
for (initialization; test condition; iteration statement){
Statement(s) to be executed if test condition is true
}
```

Example

Try the following example to learn how a for loop works in JavaScript.

```
<html>
    <body>
<script type="text/javascript">
    <!—
    var count;
document.write("Starting Loop" + "<br />");
   for(count = 0; count < 10; count++){
 document.write("Current Count : " + count );
  document.write("<br />");
}
document.write("Loop stopped!");
    //—>
  </script>
<p>Set the variable to different value and then try...</p>
    </body>
    </html>
```

Output

Starting Loop
Current Count : 0
Current Count : 1
Current Count : 2
Current Count : 3
Current Count : 4
Current Count : 5
Current Count : 6
Current Count : 7
Current Count : 8
Current Count : 9

8. What is the HTML DOM.

Ans :

The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

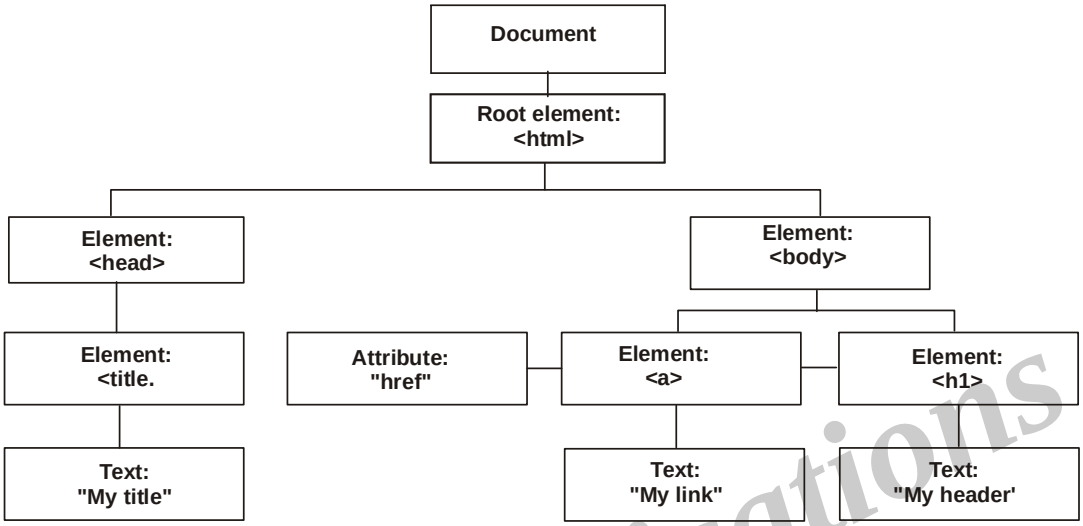
In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

The HTML DOM (Document Object Model)

When a web page is loaded, the browser creates a Document Object Model of the page.

The HTML DOM model is constructed as a tree of Objects:

The HTML DOM Tree of Objects



9. Define Date-Object

Ans :

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

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

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

Syntax

You can use any of the following syntaxes to create a Date object using Date() constructor.

new Date( )

new Date(milliseconds)

new Date(datestring)

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

Note - Parameters in the brackets are always optional.

Here is a description of the parameters -

- **No Argument** - With no arguments, the Date() constructor creates a Date object set to the current date and time.

- **milliseconds** - When one numeric argument is passed, it is taken as the internal numeric representation of the date in milliseconds, as returned by the getTime() method. For example, passing the argument 5000 creates a date that represents five seconds past midnight on 1/1/70.
- **datestring** - When one string argument is passed, it is a string representation of a date, in the format accepted by the Date.parse() method.
- **7 agruments** - To use the last form of the constructor shown above. Here is a description of each argument "
  - **year** - Integer value representing the year. For compatibility (in order to avoid the Y2K problem), you should always specify the year in full; use 1998, rather than 98.
  - **month** - Integer value representing the month, beginning with 0 for January to 11 for December.
  - **date** - Integer value representing the day of the month.
  - **hour** - Integer value representing the hour of the day (24-hour scale).
  - **minute** - Integer value representing the minute segment of a time reading.
  - **second** - Integer value representing the second segment of a time reading.
  - **millisecond** - Integer value representing the millisecond segment of a time reading.

10. Explain briefly about Onclick Event.

Ans :

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.

Example

Try the following example.

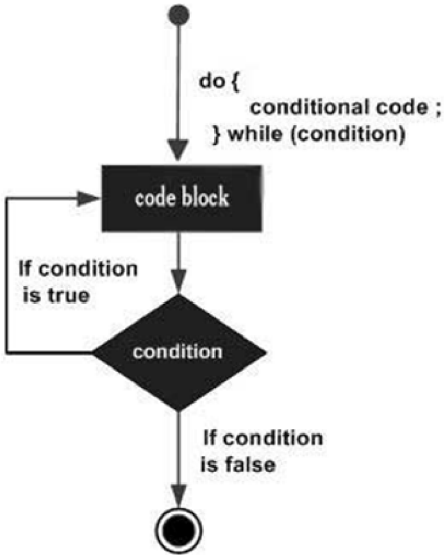
```
<html>
<head>
<script type="text/javascript">
<!—
function sayHello() {
alert("Hello World")
}
//—>
</script>
</head>
<body>
<p>Click the following button and see result</p>
<form>
<inputtype="button" onclick=" sayHello()" value="Say Hello" />
</form>
</body>
</html>
```

11. Explain the do...while Loop in Java.

Ans :

The do...while loop is similar to the while loop except that the condition check happens at the end of the loop. This means that the loop will always be executed at least once, even if the condition is false.

Flow Chart. The flow chart of a do-while loop would be as follows -



Syntax

The syntax for do-while loop in JavaScript is as follows -

```
do{
Statement(s) to be executed;
} while (expression);
```

Note- Don't miss the semicolon used at the end of the do...while loop.

12. Arithmetic Operators in Java's cripts

Ans :

JavaScript Xsupports the following arithmetic operators -

Assume variable A holds 10 and variable B holds 20, then -

Operator	Description
Addition	Adds two operandsEx: A + B will gve 30
Subtraction	Subtracts the second operand from the firstEx: A - B will give -10
Multiplication	Multiply both operands Ex: A* B will give 200
Division	Divide the numerator by the denominatorEx: B/A will give 2
Modulus	Outputs the remainder of an integer divisionEx: B % A will give 0
++(Increment)	Increases an integer value by oneEx: A+ + will give 11
--(Decrement)	Decreases an integer value by oneEx: A- will give 9

Note: Addition operator (+) works for Numeric as well as Strings. e.g. "a" + 10 will give "a10".

Choose the Correct Answer

1. ____ is an Event where mouse is frequently used. [a]
   (a) On click (b) On submit
   (c) On mouse out (d) On mouse out
2. Math. sin (30) is an object of [b]
   (a) Date (b) Math
   (c) Time (d) All
3. ____ is used to store collection of data ? [b]
   (a) datatype (b) array
   (c) variable (d) none
4. How many types of function's we have. [a]
   (a) 2 (b) 3
   (c) 4 (d) 5
5. For-loop includes how many statements. [c]
   (a) 2 (b) 4
   (c) 3 (d) 5
6. Type of operator is an ____ operator ? [a]
   (a) Unary (b) Binary
   (c) ternary (d) tertiary
7. How many types of operators are supported by Java script. [d]
   (a) 4 (b) 8
   (c) 6 (d) 5
8. How many types of variables we have [a]
   (a) 2 (b) 3
   (c) 1 (d) 4
9. Where we can insert the Java Script [c]
   (a) head (b) body
   (c) a & b (d) none
10. How many types of Data types supported by Java Script ? [a]
    (a) 5 (b) 6
    (c) 4 (d) 3

Fill in the blanks

1. Java Script is a _____ Programing Language.
2. Java Script code placed in between _____.
3. _____ which writers a string into our html document.
4. The script should be included at Browser side is called _____.
5. Where can you insert a script in html code _____.
6. Date is _____ data type.
7. _____ is used to represent true/false values.
8. _____ can be defined anywhere in your Java Script code.
9. 4 + 5 = 9. Here 4 and 5 are _____.
10. Conditional Operator is also called as _____ operator.

ANSWERS

1. Light weight & interpretted
2. <script >.....</script>
3. document write
4. client -side-Java -Script
5. <body> (or) <head>
6. Non-primitive
7. Boolean
8. Global variables
9. Operands
10. Ternary

One Mark Answers

1. Serverside Environment

Ans :

The Script which runs at server side.

2. Local variable

Ans :

It will be visible only with in a function where it is defined.

3. Operator

Ans :

Operator is a symbol used to define (or) perform some mathametical operations.

4. Function

Ans :

Is a group of Reusable code which can be called any where in your program.

5. Use of new operator

Ans :

Used to create instance of an object.

UNIT IV

EVENTS AND EVENT HANDLERS

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 - Onsubmit - Onunload.

4.1 General Information About Events

4.1.1 Introduction

Q1. What do you understand by HTML events?

Ans :

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

HTML allows event handler attributes, with JavaScript code, to be added to HTML elements.

With single quotes

```
<elementevent='some JavaScript'>
```

With double quotes

```
<elementevent="some JavaScript">
```

In the following example, an onclick attribute (with code), is added to a button element:

Example

```
<!DOCTYPE html>
<html>
<body>
<button
 onclick="document.get Element ByI
d('demo').inner
HTML=Date()">The time is?</button>
<p id="demo"></p>
</body>
</html>
```

Output

Tue Jun 05 2018 10:26:30 GMT+0530 (India Standard Time)

Q2. Explain the purpose of Javascript.

Ans :

Event handlers can be used to handle, and verify, user input, user actions, and browser actions:

- Things that should be done every time a page loads.
- Things that should be done when the page is closed.
- Action that should be performed when a user clicks a button.
- Content that should be verified when a user inputs data.

Many different methods can be used to let JavaScript work with events:

- HTML event attributes can execute JavaScript code directly.
- HTML event attributes can call JavaScript functions.
- You can assign your own event handler functions to HTML elements.
- You can prevent events from being sent or being handled.

4.2 EVENT

Q3. What is an event? List out the events supported by javascript?

(OR)

Define Event. Explain the various attributes supported by the JavaScript.

Ans : **(June-19, Imp.)**

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.

When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.

Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.

Events are actions or occurrences that happen in the system you are programming, which the system tells you about so you can respond to them in some way if desired. For example, if the user clicks a button on a webpage, you might want to respond to that action by displaying an information box.

Following are basic events supported by java script along with html

Attribute	Value	Description
Offline	script	Triggers when the document goes offline
Onabort	script	Triggers on an abort event
onafterprint	script	Triggers after the document is printed
onbeforeonload	script	Triggers before the document loads
onbeforeprint	script	Triggers before the document is printed
onblur	script	Triggers when the window loses focus
oncanplay	script	Triggers when media can start play, but might has to stop for buffering
oncanplaythrough	script	Triggers when media can be played to the end, without stopping for buffering
onchange	script	Triggers when an element changes
onclick	script	Triggers on a mouse click
oncontextmenu	script	Triggers when a context menu is triggered
ondblclick	script	Triggers on a mouse double-click
ondrag	script	Triggers when an element is dragged
ondragend	script	Triggers at the end of a drag operation
ondragenter	script	Triggers when an element has been dragged to a valid drop target
ondragleave	script	Triggers when an element is being dragged over a valid drop target
ondragover	script	Triggers at the start of a drag operation
ondragstart	script	Triggers at the start of a drag operation
ondrop	script	Triggers when dragged element is being dropped
ondurationchange	script	Triggers when the length of the media is changed

onemptied	script	Triggers when a media resource element suddenly becomes empty.
onended	script	Triggers when media has reach the end
onerror	script	Triggers when an error occur
onfocus	script	Triggers when the window gets focus
onformchange	script	Triggers when a form changes
onforminput	script	Triggers when a form gets user input
onhaschange	script	Triggers when the document has change
oninput	script	Triggers when an element gets user input
oninvalid	script	Triggers when an element is invalid
onkeydown	script	Triggers when a key is pressed
onkeypress	script	Triggers when a key is pressed and released
onkeyup	script	Triggers when a key is released
onload	script	Triggers when the document loads
onloadeddata	script	Triggers when media data is loaded
onloadedmetadata	script	Triggers when the duration and other media data of a media element is loaded
onloadstart	script	Triggers when the browser starts to load the media data
onmessage	script	Triggers when the message is triggered
onmousedown	script	Triggers when a mouse button is pressed
onmousemove	script	Triggers when the mouse pointer moves
onmouseout	script	Triggers when the mouse pointer moves out of an element
onmouseover	script	Triggers when the mouse pointer moves over an element
onmouseup	script	Triggers when a mouse button is released
onmousewheel	script	Triggers when the mouse wheel is being rotated
onoffline	script	Triggers when the document goes offline
onoine	script	Triggers when the document comes online
ononline	script	Triggers when the document comes online
onpagehide	script	Triggers when the window is hidden
onpageshow	script	Triggers when the window becomes visible
onpause	script	Triggers when media data is paused
onplay	script	Triggers when media data is going to start playing
onplaying	script	Triggers when media data has start playing
onpopstate	script	Triggers when the window's history changes
onprogress	script	Triggers when the browser is fetching the media data
onratechange	script	Triggers when the media data's playing rate has changed
onreadystatechange	script	Triggers when the ready-state changes
onredo	script	Triggers when the document performs a redo
onresize	script	Triggers when the window is resized
onscroll	script	Triggers when an element's scrollbar is being scrolled

onseeked	script	Triggers when a media element's seeking attribute is no longer true, and the seeking has ended
onseeking	script	Triggers when a media element's seeking attribute is true, and the seeking has begun
onselect	script	Triggers when an element is selected
onstalled	script	Triggers when there is an error in fetching media data
onstorage	script	Triggers when a document loads
onsubmit	script	Triggers when a form is submitted
onsuspend	script	Triggers when the browser has been fetching media data, but stopped before the entire media file was fetched
ontimeupdate	script	Triggers when media changes its playing position
onundo	script	Triggers when a document performs an undo
onunload	script	Triggers when the user leaves the document
onvolumechange	script	Triggers when media changes the volume, also when volume is set to "mute"
onwaiting	script	Triggers when media has stopped playing, but is expected to resume

Q4. Explain briefly about Event handling mechanism?

Ans :

Event Handling

Any program that uses GUI (graphical user interface) such as Java application written for windows, is event driven. Event describes the change in state of any object.

For Example : Pressing a button, Entering a character in Textbox, Clicking or Dragging a mouse, etc.

Event handling is the receipt of an event at some event handler from an event producer and subsequent processes. The processes involved in event handling include: Identifying where an event should be forwarded. Making the forward

Components of Event Handling

Event handling has three main components,

- **Events :** An event is a change in state of an object.
- **Events Source :** Event source is an object that generates an event.
- **Listeners :** A listener is an object that listens to the event. A listener gets notified when an event occurs.

4.3 ONABORT

Q5. Explain briefly about OnAbort event?

Ans :

The OnAbort event occurs when loading of an image is aborted.

The OnAbort event occurs when the user aborts the loading of an <img> or <input type="image"> element.

Syntax

onabort="SomeJavaScriptCode"

Parameter	Description
SomeJavaScript Code	Required. Specifies a JavaScript to be executed when the event occurs.

Supported by the following HTML tags:

Supported by the following JavaScript objects:

image

Example 1 :

In this example an alert box will be displayed if the loading of the image is aborted:

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

Example 2

In this example we will call a function if the loading of the image is aborted:

```
<html>
<head>
<script type="text/javascript">
function abortImage()
{
alert('Error: Loading of the image was aborted');
}
</script>
</head>
<body>
<img src="image_w3default.gif"
onabort="abortImage()">
</body>
</html>
```

4.4 ONCLICK

Q6. Explain in detail about OnClick event?

Ans :

The onclick event occurs when the user clicks on an element.

Syntax

In HTML :

```
<element onclick="myScript">
```

In JavaScript :

object.onclick = function(){myScript};

program:

```
<!DOCTYPE html>
<html>
<body>
<pid="demo"
 onclick="myFunction()">
 Click me to change my text color.</p>
<script>
function myFunction() {
document.getElementById("demo").
style.color = "red";
}
</script>
</body>
</html>
```

Output :

Click me to change my text color.

4.5 ONDBL CLICK

Q7. Explain in detail about Ondbl click ?

Ans :

The dblclick event occurs when an element is double-clicked. The dblclick() method triggers the dblclick event, or attaches a function to run when a dblclick event occurs. Tip: The dblclick event also generates a click event. This can cause problems if both events are applied to the same element.

Syntax

In HTML :

<element ondblclick="myScript">

In JavaScript :

object.ondblclick = function(){myScript};

program:

```
<!DOCTYPE html>
<html>
<body>
<pid="demo"
 ondblclick="myFunction()">
```

```
Double-click me to change my text color.</
p>
<p>A function is triggered when the p
element is double-clicked. The function sets
the color of the p element to red.</p>
    <script>
function myFunction() {
document.getElementById("demo").
    style.color = "red";
}
    </script>
    </body>
    </html>
```

Output :

Double-click me to change my text color.

A function is triggered when the p element is double-clicked. The function sets the color of the p element to red.

4.6 ONDRAG DROP

Q8. Explain in detail about Ondrag drop event ?

Ans :

The ondrag event occurs when an element or text selection is being dragged.

Drag and drop is a very common feature in HTML5. It is when you "grab" an object and drag it to a different location. For more information, see our HTML Tutorial on HTML5 Drag and Drop.

There are many events 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 - occurs when the user starts to drag an element
  - ondrag - occurs when an element is being dragged
  - ondragend - occurs when the user has finished dragging the element
- **Events fired on the drop target:**
  - ondragenter - occurs when the dragged element enters the drop target
  - ondragover - occurs when the dragged element is over the drop target
  - ondragleave - occurs when the dragged element leaves the drop target
  - ondrop - occurs when the dragged element is dropped on the drop target

Syntax

In HTML :

```
<element ondrag="myScript">
```

In JavaScript :

```
object.ondrag = function(){myScript};
```

4.7 ONERROR

Q9. Explain in detail about Onerror event with suitable example.

Ans : **(June-19, Imp.)**

The onerror event is fired whenever there is a script error in the page.

To use the onerror event, you must create a function to handle the errors. Then you call the function with the onerror event handler. The event handler is called with three arguments: msg (error message), url (the url of the page that caused the error) and line (the line where the error occurred).

```
    onerror=handleErr
    function handleErr(msg,url,l)
{
    //Handle the error here
    return true or false
}
```

The value returned by onerror determines whether the browser displays a standard error message. If you return false, the browser displays the standard error message in the JavaScript console. If you return true, the browser does not display the standard error message.

Example :

The following example shows how to catch the error with the onerror event:

```
<html>
<head>
<script type="text/javascript">
onerror=handleErr;
var txt="";
function handleErr(msg,url,l)
{
txt="There was an error on this
page.\n\n";
txt+="Error: " + msg + "\n";
txt+="URL: " + url + "\n";
txt+="Line: " + l + "\n\n";
txt+="Click OK to continue.\n\n";
alert(txt);
return true;
}
function message()
{
adddlert("Welcome guest!");
}
</script>
</head>
<body>
<input type="button" value="View
message" onclick="message()" />
</body>
</html>
```

4.8 ONFOCUS

Q10. Explain about briefly Onfocus event?

Ans :

The onfocus event occurs when an element gets focus.

The onfocus event is most often used with <input>, <select>, and <a>. The onfocus event is the opposite of the onblur event.The onfocus event is similar to the onfocusin event. The main difference is that the onfocus event does not bubble. Therefore, if you want to find out whether an element or its child gets the focus, you could use the onfocusin event. However, you can achieve this by using the optional *useCapture* parameter of the addEventListener() method for the onfocus event.

Syntax

In HTML:

```
<element onfocus="myScript">
```

In JavaScript:

```
object.onfocus = function(){myScript};
```

Program :

```
<!DOCTYPE html>
<html>
<body>
Enter your name: <input type="text"
onfocus="myFunction(this)">
<p>When the input field gets focus, a
function is triggered which changes the
background-color.</p>
<script>
function myFunction(x) {
x.style.background = "yellow";
}
</script>
</body>
</html>
```

Output:

Enter your name:

When the input field gets focus, a function is triggered which changes the background-color.

4.9 ONKEY PRESS

Q11. Explain briefly about Onkey Press event?

Ans : **(Dec.-20, June-19, Dec.-18, Imp.)**

The onkeypress event occurs when the user presses a key (on the keyboard).

The order of events related to the onkeypress event:

1. onkeydown
2. onkeypress
3. onkeyup

Note: The onkeypress event is not fired for all keys (e.g. ALT, CTRL, SHIFT, ESC) in all browsers. To detect only whether the user has pressed a key, use the onkeydown event instead, because it works for all keys.

Syntax

In HTML :

```
<element onkeypress="myScript">
```

In JavaScript:

```
object.onkeypress = function(){myScript};
```

Program :

```
<!DOCTYPE html>
<html>
<body>
<p>A function is triggered when the user is pressing a key in the input field.</p>
<inputtype="text"onkeypress="my
Function()">
<script>
function myFunction() {
alert("You pressed a key inside the input field");
}
</script>
</body>
</html>
```

Output:

A function is triggered when the user is pressing a key in the input field. K.MAHESH

4.10 ONKEY UP

Q12. Explain briefly about Onkey Up event?

Ans : **(Dec.-20, June-19, Dec.-18, Imp.)**

The onkeyup event occurs when the user releases a key (on the keyboard).

The order of events related to the onkeyup event:

1. Onkeydown
2. Onkeypress
3. Onkeyup

Syntax

In HTML:

```
<element onkeyup="myScript">
```

In JavaScript:

```
object.onkeyup = function(){myScript};
```

Program :

```
<!DOCTYPE html>
<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. get Element By Id("fname");
x.value = x.value.toUpperCase();
}
</script>
</body>
</html>
```

Output:

A function is triggered when the user releases a key in the input field. The function transforms the character to upper case.

Enter your name:

4.11 ONLOAD

Q13. Explain briefly about Onloadevent?

Ans :

The onload event occurs when an object has been loaded.

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.).

The onload event can be used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.

The onload event can also be used to deal with cookies

Syntax

In HTML:

<element onload="myScript">

In JavaScript:

object.onload = function(){myScript};

Syntax

In HTML:

<element onload="myScript">

In JavaScript:

object.onload = function(){myScript};

Program :

```
<!DOCTYPE html>
    <html>
<body onload="myFunction()">
<h1>Hello World!</h1>
    <script>
function myFunction() {
alert("Page is loaded");
}
    </script>
    </body>
    </html>
```

Output:

Page is loaded

Hello World

4.12 ONMOUSE DOWN

Q14. Explain briefly about Onmouse Down event?

Ans : **(Dec.-20, Dec.-18, Imp.)**

He onmousedown event occurs when a user presses a mouse button over an element.

The order of events related to the onmouse down event (for the left/middle mouse button):

1. onmousedown
2. onmouseup
3. onclick

The order of events related to the onmouse down event (for the right mouse button):

1. onmousedown
2. onmouseup
3. oncontextmenu

Syntax

In HTML:

<element onmousedown="myScript">

In JavaScript:

object.onmousedown= function(){myScript};

Program:

```
<!DOCTYPE html>
    <html>
    <body>
<pid="myP" on mouse down ="mouse
Down()"onmouse up=" mouse Up ()" >
```

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.get Element By Id ("myP"). style.color = "red";
}
function mouseUp() {document.getElementById("myP"). style.color = "green";
}
</script>
</body>
</html>
```

4.13 ONMOUSE MOVE

Q15. Explain briefly about Onmouse Move event?

Ans : **(Dec.-20, Dec.-18, Imp.)**

The onmousemove event occurs when the pointer is moving while it is over an element.

Occurs when the user moves the mouse over the element.

Use the onmouseover event to receive a notification when the user moves the mouse pointer into and the onmouseout event to receive a notification when the user moves the mouse pointer out of an element.

Syntax:

In HTML:

```
<element onmousemove="myScript">
```

In JavaScript:

```
object.onmousemove = function(){myScript};
```

Program:

```
<!DOCTYPE html>
    <html>
    <head>
    <style>
    div {
width: 200px;
height: 100px;
border: 1px solid black;
}
</style>
</head>
<body>
```

```
<div onmousemove="myFunction(event)" onmouseout="clearCoor()"></div>
<p>Mouse over the rectangle above, and get the coordinates of your mouse pointer.</p>
<p id="demo"></p>
<script>
function myFunction(e) {
var x = e.clientX;
var y = e.clientY;
var coor = "Coordinates: (" + x + "," + y + ")";
document.getElementById("demo").innerHTML = coor;
}
function clearCoor() {
document.getElementById("demo"). innerHTML = "";
}
</script>
</body>
</html>
```

Output:

Coordinates: (120,17)

4.14 ONMOUSE OUT

Q16. Explain briefly about Onmouse Out event?

Ans : **(Dec.-20, Dec.-18, Imp.)**

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, or onto one of its children.

Occurs when the user moves the mouse pointer out of the element.

Use the onmouseover event to receive a notification when the user moves the mouse pointer into and the onmousemove event to receive a notification when the user moves the mouse pointer over an element.

Note: the onmouseout event is not fired during a drag operation. For that case, use the ondragleave event.

Program:

```
<head>
<scripttype="text/javascript">
function OnMouseIn (elem) {
elem.style.border = "2px solid blue";
}
```

```
function OnMouseOut (elem) {
elem.style.border = "";
}
</script>
</head>
<body>
<divstyle="background-color:
#d0f0a0; width:200px"
onmouseover="OnMouseIn(this)"
onmouseout="OnMouseOut (this)">
Move your mouse pointer into and out of this element!
</div>
</body>
```

4.15 ONMOUSE OVER

Q17. Explain about Onmouse Over event?

Ans : **(Dec.-20, Dec.-18)**

The onmouseover event occurs when the mouse pointer is moved onto an element, or onto one of its children.

This event is often used together with the onmouseout event, which occurs when a user moves the mouse pointer out of an element.

Syntax

In HTML:

```
<element onmouseover="myScript">
```

In JavaScript:

```
object.onmouseover = function(){myScript};
```

Program:

```
<!DOCTYPE html>
<html>
<body>
<img onmouseover="bigImg(this)" on mouseout="normalImg(this)" border="0" src="smiley.gif"
alt="Smiley" width="32" height="32">
<p>The function bigImg() is triggered when the user moves the mouse pointer over the image.</p>
<p>The function normalImg() is triggered when the mouse pointer is moved out of the image.</p>
<script>
```

```
        function bigImg(x) {
        x.style.height = "64px";
        x.style.width = "64px";
    }
        function normalImg(x) {
        x.style.height = "32px";
        x.style.width = "32px";
    }
        </script>
        </body>
        </html>
```

Output :

The function bigImg() is triggered when the user moves the mouse pointer over the image.

The function normalImg() is triggered when the mouse pointer is moved out of the image.

4.16 ONMOVE

Q18. Explain about Onmove event?

Ans : **(Dec.18, Imp.)**

Occurs when the position of an element's top-left corner is changed.

To get the top-left corner of an element, use the clientTop, clientLeft, offsetTop and offsetLeft properties.

The placement of an element can be specified with the left, top, right, bottom, width and height style properties. The left, top, right and bottom properties only have effect on relative, absolute or fixed positioned elements. When the position of an absolute or fixed positioned element's top-left corner is changed, an onmove event is fired on the element. The onmove event is not fired on static and relative positioned elements.

The position of elements in an editable region (see the contentEditable and designMode properties) can be modified by the user through the user interface. When the user modifies the position of a control element (see form controls) or a relative or absolute positioned element in an editable region, an onmove event is fired on the element.

Action	Event order
Changing the position by script.	l.onmove
Resizing the browser window if itcauses the position of a positionedelement to change.	l.onmove
Dragging a control element.	l.onmove
Dragging a positioned element.	l.onmovestart 2.onmove 3.onmoveend

4.17 ONREST

Q19. Explain briefly about OnRest event?

Ans :

The onreset event occurs when the reset button in a form is clicked.

Syntax

onreset="SomeJavaScriptCode"

Parameter	Description
SomeJavaScriptCode	Required. Specifies a JavaScript to be executed when the event occurs.

Supported by the following HTML tags:

<form>

Supported by the following JavaScript objects

form

Example :

In this example the form changes back to the default values and displays an alert box when the reset button is clicked:

```
<form onreset="alert('The form will be
reset')">
Firstname: <input type="text" name="fname" value="John" /> <br />
Lastname: <input type="text" name="lname" />
<br /><br />
<input type="reset" value="Reset">
</form>
```

The output of the code above will be:

Firstname: John

Lastname:

Submit

4.18 ONRESIZE

Q20. Explain briefly about Onresize event?

Ans :

The onresize event occurs when the browser window has been resized.

To get the size of an element, use the clientWidth, clientHeight, innerWidth, innerHeight, outerWidth, outerHeight, offsetWidth and/or offsetHeight properties.

Syntax

In HTML:

```
<element onresize="myScript">
```

In JavaScript:

```
object.onresize = function(){myScript};
```

Program:

```
<!DOCTYPE html>
    <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>
```


4.19 ONSELECT

Q21. Explain about Onselect event?

Ans :

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.

Syntax

In HTML:

```
<element onselect="myScript">
```

In JavaScript:

```
object.onselect = function(){myScript};
```

Program:

```
<!DOCTYPE html>
```

```
<html>
<body>
Select some of the text: <input type="text" value="Helloworld!"onselect="my Function()">
<script>
function myFunction() {
alert("You selected some text!");
}
</script>
</body>
</html>
```

Output:

Select some of the text: K Mahesh

4.20 ON SUBMIT

Q22. Explain about On submit event?

Ans : **(Imp.)**

The onsubmit event occurs when a form is submitted.

Syntax

In HTML:

```
<element onsubmit="myScript">
```

In JavaScript:

```
object.onsubmit = function(){myScript};
```

Program:

```
<!DOCTYPE html>
<html>
<body>
<p>When you submit the form, a function is triggered which alerts some text.</p>
<formaction="/action_page.php"on submit="myFunction()">
Enter name: <input type="text" name=" fname">
<inputtype="submit" value="Submit">
</form>
<script>
function myFunction() {
alert("The form was submitted");
}
</script>
</body>
</html>
```

Output:

When you submit the form, a function is triggered which alerts some text.

Top of Form

Enter name: Mahesh K submit

4.21 On unload

Q23. Explain about On unloadevent?

Ans : **(Imp.)**

The onunload event occurs once a page has unloaded (or the browser window has been closed).

Onunload occurs when the user navigates away from the page (by clicking on a link, submitting a form, closing the browser window, etc.).

Note: The onunload event is also triggered when a user reloads the page (and the onload event).

Syntax

In HTML:

```
<element onunload="myScript">
```

In JavaScript:

```
object.onunload = function(){myScript};
```

Program:

```
<!DOCTYPE html>
    <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>
```

Output:

Welcome to my Home Page

Close this window or press F5 to reload the page.

Note: Due to different browser settings, this event may not always work as expected.

Short Question and Answers

Q1. Define Event handling Mechanism?

Ans :

Any program that uses GUI (graphical user interface) such as Java application written for windows, is event driven. Event describes the change in state of any object.

For Example : Pressing a button, Entering a character in Textbox, Clicking or Dragging a mouse, etc.

Event handling is the receipt of an event at some event handler from an event producer and subsequent processes. The processes involved in event handling include: Identifying where an event should be forwarded. Making the forward

Components of Event Handling

Event handling has three main components,

- **Events :** An event is a change in state of an object.
- **Events Source :** Event source is an object that generates an event.
- **Listeners :** A listener is an object that listens to the event. A listener gets notified when an event occurs.

Q2. Explain in detail about Ondbl click event.

Ans :

The dblclick event occurs when an element is double-clicked. The dblclick() method triggers the dblclick event, or attaches a function to run when a dblclick event occurs. Tip: The dblclick event also generates a click event. This can cause problems if both events are applied to the same element.

Syntax

In HTML :

```
<element ondblclick="myScript">
```

In JavaScript :

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

program:

```
<!DOCTYPE html>
<html>
<body>
<pid="demo"
 ondblclick="myFunction()">
Double-click me to change my text color.</
p>
<p>A function is triggered when the p
element is double-clicked. The function sets
the color of the p element to red.</p>
<script>
function myFunction() {
document.getElementById("demo").
style.color = "red";
}
</script>
</body>
</html>
```

Output :

Double-click me to change my text color.

Q3. Explain Onloadevent.

Ans :

The onload event occurs when an object has been loaded.

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.).

The onload event can be used to check the visitor's browser type and browser version, and load the proper version of the web page based on the information.

The onload event can also be used to deal with cookies

Syntax

In HTML:

```
<element onload="myScript">
```

In JavaScript:

```
object.onload = function(){myScript};
```

Syntax

In HTML:

```
<element onload="myScript">
```

In JavaScript:

```
object.onload = function(){myScript};
```

Program :

```
<!DOCTYPE html>
<html>
<body onload="myFunction()">
<h1>Hello World!</h1>
<script>
function myFunction() {
alert("Page is loaded");
}
</script>
</body>
</html>
```

Output:

Page is loaded

Hello World

Q4. Explain about On submit event?

Ans :

The onsubmit event occurs when a form is submitted.

Syntax

In HTML:

```
<element onsubmit="myScript">
```

In JavaScript:

```
object.onsubmit = function(){myScript};
```

Program:

```
<!DOCTYPE html>
<html>
<body>
<p>When you submit the form, a function is triggered which alerts some text.</p>
<formaction="/action_page.php"on submit="myFunction()">
Enter name: <input type="text" name="fname">
<inputtype="submit" value="Submit">
</form>
<script>
function myFunction() {
alert("The form was submitted");
}
</script>
</body>
</html>
```

Output:

When you submit the form, a function is triggered which alerts some text.

Top of Form

Enter name: Mahesh K submit

Q5. Ondrag drop event. Explain.

Ans :

The ondrag event occurs when an element or text selection is being dragged.

Drag and drop is a very common feature in HTML5. It is when you "grab" an object and drag it to a different location. For more information, see our HTML Tutorial on HTML5 Drag and Drop.

There are many events 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 - occurs when the user starts to drag an element
  - ondrag - occurs when an element is being dragged
  - ondragend - occurs when the user has finished dragging the element

➢ **Events fired on the drop target:**

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

Q6. What is an event ?

Ans :

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.

When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.

Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.

Events are actions or occurrences that happen in the system you are programming, which the system tells you about so you can respond to them in some way if desired. For example, if the user clicks a button on a webpage, you might want to respond to that action by displaying an information box.

Q7. List Out various event handlers.

Ans :

Event Handlers

Once the event is generated, there is often requirement of code to process these events. Such code is known as event handlers. In general, event handlers are of two types.

1. Interactive Event Handlers

The event handlers which solely relies on the users's activity for them to be invoked are interactive event handler. **Ex:** onClick, onBlur etc.

2. Non-Interactive Event Handlers

The event handlers which do not rely on the user's for them to be invoked are non-interactive event handlers. **Ex:** onPageLoad etc.

Q8. Illustrate the differences between onMouseMove and onMouseOver event handler.

Ans :

onMouseMove

onMouseMove event occurs when the pointer moves when it is on an element. This event is handled by onMouseMove event handler. The browser will be checking the mouse position constantly. If any changed in mouse positive a observed then the onMouseMove event is triggered.

<input type = "text" onMouseMove = " HandlerText value = " ">

onMouseOver

onMouseOver event will occur when the mouse pointer is moved over an element. This event is handled by the onMouseOver event handler. The programmer can code this event and even add an alert when mouse pointer is moved onto any element.

{Input type = "text" onMouseOver = "HandlerText" value = " ">

Q9. Write about onSubmit even handler.

Ans :

OnSubmit is a form based event. Whenever user fills in the given form (supplied on the internet) and ensures that his entries has to be loaded on a server then he clicks "submit" button which is provided at the end of each form. Hence, ONSUBMIT event fires whenever the button to which it is associated is clicked.

<input type = "button" onsubmit = "handlerText" value = "submit">

Choose the Correct Answer

1. Onblur is responsible for ? [b]
   (a) Off line (b) Loses focus
   (c) Mouse clict (d) Dropped
2. Following is not a keyboard Event ? [d]
   (a) Key up (b) Key press
   (c) Key dows (d) On load
3. ____ Event is fired when Ever there is a Script Error [b]
   (a) Enor message (b) On Error
   (c) On Err (d) None
4. Following Event is very similer to on mouse up Event [b]
   (a) Key up (b) On mousedown
   (c) Scroll (d) None
5. Which one is a Button Event ? [c]
   (a) resize (b) select
   (c) onsubmit (d) all
6. Which Event Occurs when Inset Button in a form is clicked [c]
   (a) On move (b) On resize
   (c) On reset (d) None
7. Which one is not a form Events [d]
   (a) Onreset (b) On submit
   (c) Onselect (d) On keypress
8. Load and unload works at [b]
   (a) Serverside (b) Client side
   (c) Both (d) None
9. HTML allows Event handles attributes with [c]
   (a) DOM (b) CSS
   (c) Java script (d) XML
10. Event handling has which component [d]
    (a) Events (b) Event source
    (c) Listoners (d) all

Fill in the blanks

1. On click is a _____ Event.
2. Events are _____ that happen to html Elements.
3. GVI stands for _____.
4. _____ Event occurs when loading of an image is aboarted.
5. _____ Event occurs when an Element is double clicked.
6. Onmore is used to change the _____ by script.
7. Hight, width comes under _____ Event .
8. Keydown is _____ Event.
9. Event handlers are used to verify _____.
10. Events are part of _____.

ANSWERS

1. Mouse
2. Things
3. “Graphical User Interface”
4. Onabort
5. db click
6. Position
7. Onresize
8. Key board
9. “Browser actions”
10. DOM

One Mark Answers

1. What is Ondbclick.

Ans :

It occurs when an element is double clicked.

2. Define Onload.

Ans :

It occur's when an object has been loaded.

3. What is Onsubmit.

Ans :

It occur's when a form is submitted.

4. What is On move.

Ans :

It occur's when the position of an Element top. Left corner is changed.

5. What is On mouseout.

Ans :

When the mouse pointer is moved out of an element.

UNIT V

EXTENSIBLE MARKUP LANGUAGE (XML) :

Extensible Markup Language (XML): Introduction - Creating XML Documents - XML style Sheet – Hyperlinks in XML Document Object Model - XML Query Language.

5.1 INTRODUCTION TO XML

Q1. Give a brief introduction about XML.

Ans :

XML stands for Extensible Markup Language and is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and solutions.

- XML is extensible - XML allows you to create your own self-descriptive tags, or language, that suits your application.
- XML carries the data, does not present it - XML allows you to store the data irrespective of how it will be presented.
- XML is a public standard - XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.

XML Usage

A short list of XML usage says it all -

- XML can work behind the scene to simplify the creation of HTML documents for large web sites.
- XML can be used to exchange the information between organizations and systems.
- XML can be used for offloading and reloading of databases.
- XML can be used to store and arrange the data, which can customize your data handling needs.
- XML can easily be merged with style sheets to create almost any desired output.
- Virtually, any type of data can be expressed as an XML document.

Q2. What is Markup? Is XML is a markup language ?

Ans :

XML is a markup language that defines set of rules for encoding documents in a format that is both human-readable and machine-readable. Markup is information added to a document that enhances its meaning in certain ways, in that it identifies the parts and how they relate to each other. More specifically, a markup language is a set of symbols that can be placed in the text of a document to demarcate and label the parts of that document.

Following example shows how XML markup looks, when embedded in a piece of text

```
<message>
<text>Hello, world!</text>
</message>
```

This snippet includes the markup symbols, or the tags such as <message>...</message> and <text>... </text>. The tags <message> and </message> mark the start and the end of the XML code fragment. The tags <text> and </text> surround the text Hello, world!.

A programming language consists of grammar rules and its own vocabulary which is used to create computer programs. These programs instruct the computer to perform specific tasks.

XML does not qualify to be a programming language as it does not perform any computation or algorithms. It is usually stored in a simple text file and is processed by special software that is capable of interpreting XML.

Q3. What are the features of XML?

(OR)

Explain in detail about features of XML.

Ans : **(Imp.)**

1. XML is very simple as that of HTML. It has very few syntax rules.
2. XML files are text files. They can be managed by all text editors.
3. XML is extensible. It specifies only the structural rules of tags, allowing the create their own tags or use the existing tags.
3. XML is extremely suitable for document storage and processing, both online and offline.
4. Markup language is used for describing the data.
5. XML is human and computer friendly format.
6. XML stores the data in the form of tree.

Q4. Distinguish between XML and HTML.

Ans : **(Imp.)**

S.No.	XML	S.No.	HTML
1.	XML stands for Extensible Markup Language.	1.	HTML stands for Hypertext Markup Language.
2.	It is a simple markup language that share text information on web.	2.	It is a predominant markup language used based for creating web page.
3.	It defines set of rules which helps in encoding web documents.	3.	It defines structure for creating web documents.
4.	It is a case sensitive language.	4.	It is not case sensitive.
5.	It is a software and hardware independent tool that is used for transferring data between application and database.	5.	It is a presentation language used for designing static web pages.
6.	It is both client side and server side language.	6.	It is a client side language.
7.	It is dynamic in nature. Since it create dynamic web pages.	7.	It is static in nature. Since it create static web pages.
8.	It is used to describe information	8.	It is used to display information
9.	XML document starts and ends with <? and ?>	9.	HTML document starts and ends with <html> and </html>
10.	It makes use of user defined tags.	10.	It makes use of predefined tags.

5.2 Creating XML Documents

Q5. What is the basic syntax of XML program? Give some rules to develop the documents ?

Ans :

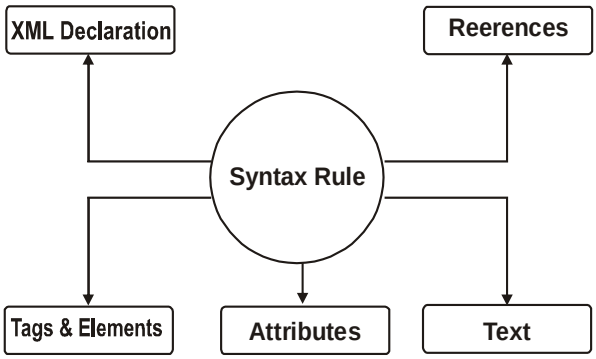
The simple syntax rules to write an XML document. Following is a complete XML document-

```
<?xml version = "1.0"?>

<contact-info>

<name>TanmayPatil</name>

<company>TutorialsPoint</company>

<phone>(011) 123-4567</phone>

</contact-info>
```

You can notice there are two kinds of information in the above example -

- Markup, like <contact-info>
- The text, or the character data, *Tutorials Point* and (040) 123-4567.

The following diagram depicts the syntax rules to write different types of markup and text in an XML document.



Let us see each component of the above diagram in detail.

i) XML Declaration

The XML document can optionally have an XML declaration. It is written as follows "

```
<?xml version = "1.0"
encoding = "UTF-8"?>
```

Where version is the XML version and encoding specifies the character encoding used in the document.

Syntax Rules for XML Declaration

- The XML declaration is case sensitive and must begin with "<?xml>" where "xml" is written in lower-case.
- If document contains XML declaration, then it strictly needs to be the first statement of the XML document.
- The XML declaration strictly needs be the first statement in the XML document.
- An HTTP protocol can override the value of encoding that you put in the XML declaration.

ii) Tags and Elements

An XML file is structured by several XML-elements, also called XML-nodes or XML-tags. The names of XML-elements are enclosed in triangular brackets < > as shown below-

```
<element>
```

Syntax Rules for Tags and Elements

Element Syntax- Each XML-element needs to be closed either with start or with end elements as shown below -

```
<element>....</element>
```

or in simple-cases, just this way "

```
<element/>
```

Nesting of Elements " An XML-element can contain multiple XML-elements as its children, but the children elements must not overlap. i.e., an end tag of an element must have the same name as that of the most recent unmatched start tag.

The Following example shows incorrect nested tags -

```
<?xml version = "1.0"?>
<contact-info>
<company>TutorialsPoint
<contact-info>
</company>
```

The Following example shows correct nested tags-

```
<?xml version = "1.0"?>
<contact-info>
<company>TutorialsPoint</company>
<contact-info>
```

Root Element " An XML document can have only one root element. For example, following is not a correct XML document, because both the x and y elements occur at the top level without a root element -

```
<x>...</x>
<y>...</y>
```

The Following example shows a correctly formed XML document -

```
<root>
<x>...</x>
<y>...</y>
</root>
```

Case Sensitivity - The names of XML-elements are case-sensitive. That means the name of the start and the end elements need to be exactly in the same case.

For example, <contact-info> is different from <Contact-Info>

iii) XML Attributes

An attribute specifies a single property for the element, using a name/value pair. An XML-element can have one or more attributes. For example -

```
<a href = "http://www.tutorialspoint.com/
">Tutorialspoint!</a>
```

Here href is the attribute name and http:// www.tutorialspoint.com/ is attribute value.

Syntax Rules for XML Attributes

- Attribute names in XML (unlike HTML) are case sensitive. That is, HREF and href are considered two different XML attributes.
- Same attribute cannot have two values in a syntax. The following example shows incorrect syntax because the attribute b is specified twice -

  `<ab="x"c="y"b="z">....</a>`
- Attribute names are defined without quotation marks, whereas attribute values must always appear in quotation marks. Following example demonstrates incorrect xml syntax -

  `<ab=x>....</a>`

In the above syntax, the attribute value is not defined in quotation marks.

iv) XML References

References usually allow you to add or include additional text or markup in an XML document. References always begin with the symbol "&" which is a reserved character and end with the symbol ";". XML has two types of references -

- **Entity References -** An entity reference contains a name between the start and the end delimiters. For example & where amp is name. The name refers to a predefined string of text and/or markup.
- **Character References -** These contain references, such as A, contains a hash mark ("#") followed by a number. The number always refers to the Unicode code of a character. In this case, 65 refers to alphabet "A".

v) XML Text

The names of XML-elements and XML-attributes are case-sensitive, which means the name of start and end elements need to be written in the same case. To avoid character encoding problems, all XML files should be saved as Unicode UTF-8 or UTF-16 files.

Whitespace characters like blanks, tabs and line-breaks between XML-elements and between the XML-attributes will be ignored.

Some characters are reserved by the XML syntax itself. Hence, they cannot be used directly. To use them, some replacement-entities are used, which are listed below -

Not Allowed Character	Replacement Entity	Character Description
<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

Q6. Explain about XML documents ?

Ans :

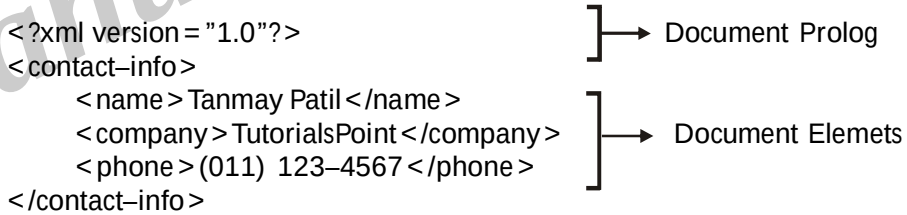
An XML document is a basic unit of XML information composed of elements and other markup in an orderly package. An XML document can contains wide variety of data. For example, database of numbers, numbers representing molecular structure or a mathematical equation.

Example

A simple document is shown in the following example-

```
<?xml version = "1.0"?>
<contact-info>
<name>TanmayPatil</name>
<company>TutorialsPoint</company>
<phone>(011) 123-4567</phone>
</contact-info>
```

The following image depicts the parts of XML document.



(i) Document Prolog Section

Document Prolog comes at the top of the document, before the root element. This section contains -

- XML declaration
- Document type declaration

You can learn more about XML declaration in this chapter " XML Declaration

(ii) Document Elements Section

Document Elements are the building blocks of XML. These divide the document into a hierarchy of sections, each serving a specific purpose. You can separate a document into multiple sections so that they can be rendered differently, or used by a search engine. The elements can be containers, with a combination of text and other elements.

5.3 XML Style Seets

Q7. Explain in detail about XML style sheets?

(OR)

What is XSL? how which helpful to XML?

Ans : **(June-19, Imp.)**

Style sheet can be defined as a form that specifies a set of layouts of document used in either word processing or desktop publishing.

XML stylesheet is a language that provides styles to XML document by using the following,

1. Extensible Stylesheet Language (XSL)
2. Extensible Stylesheet Language Transfor-mation (XSLT)
3. Cascading Style Sheet (CSS).

1. Extensible Stylesheet Language (XSL)

XSL is a language capable of transforming as well as formatting given XML documents. Hence, XSL can be referred as a language,

- It can format or structure XML data depending on the supplied values and
- It can sort or filter XML documents.

Now-a-days, XSL has emerged as one of the most powerful tools to format given XML documents into a standard which can be recognized by the browser. The most commonly used standard is HTML. The process of conversion is initi-ated by converting each single element of XML into its equivalent HTML. Besides this, various other tasks which can be accomplished by using XSL are listed below,

- Adding/deleting elements
- Sorting and filtering of XML elements
- Testing as well as making decisions about which elements to be included or deleted etc.

Consider a small XML document and convert it into its equivalent HTML file.

Demo.xml

```
<MENU>
<DVD>
    <NAME> The Moon Stone </NAME>
    <ACTOR> Harvey </ACTOR>
    <PLACE> Germany </PLACE>
    <AMOUNT> $20</AMOUNT>
    <YEAR OF_RELEASE> 2000
        </YEAR_OF_RELEASE>
```

```
</DVD>
<DVD>
    <NAME> Don</NAME>
    <ACTOR> Rajesh </ACTOR>
    <PLACE> India </PLACE>
    <AMOUNT> Rs 200 </AMOUNT>
    <YEAR_OF_RELEASE> 2005
        </YEAR_OF_RELEASE>
</DVD>
</MENU>
```

Now, following is an XSL document which can be considered as an HTML template used to acquire required HTML document (by considering XML data).

xsl_demo.xsl

```
<XSL:STYLESHEET>
<XSL: TEMPLATE MATCH = "/" >
    <HTML>
    <BODY>
        <TABLE BORDER = 'T' >
            <TR>
            <TH> NAME </TH>
            <TH> ACTOR</TH>
        </TR>

            <XSL: for each select = "MENU/DVD" > <TR>
            <TD> <XSL: Value of select = "NAME"/>
            </TD>
            <TD> <XSL: Value of select = "ACTOR"/> </TD>
            </TR>
</XSL: for each>
</TABLE>
</BODY>
</HTML>
</XSL: TEMPLATE>
</XSL:STYLESHEET>
```

The first line of XSL file declares that the current document is an XSL stylesheet. The second line indicates that the current template corresponds to the root (/) of the given XML document i.e., the current file system is traversed for XML source document where 7' refers to subdirectories. At line 10, new XSL element is encountered i.e., "XSL: for each" which searches for various XSL element and for each element a template is repeated. Hence, the name of the element, "for-each" element is followed by an attribute "select" which refers to the elements existing in the source XML document. At line number 12 and 14, an XSL element is observed, i.e., XSL value of which generally searches for all child elements and the data corresponding to that element is released into the template.

Now, in order to provide a reference of XSL file into the source XML file, include the following code into XML file.

<?XML Style sheet type = "text/XSL" href = "xsl_demo.xsl"?>

2. Extensible Stylesheet Language Transformation (XSLT)

Extensible Stylesheet Language Transformations (XSLT) is a declarative programming language that converts and restructures an XML document to some other form such as HTML or CSV. The need for such a transformation, may be to make an XML document conform to the schema of a business partner or to match with the structure of purchase orders and invoices.

The steps of transformations in XSLT are performed using an XSLT processors as follows.

1. Takes the source document as input.
2. Creates a "source tree" of the document, which is the in-memory representation of the document.
3. Processes the source tree as per the templates in the XSLT stylesheet.
4. Generates a "result tree".
5. Creates the result document from the result tree.

Step 4 is specifically called "transformation" whereas, step 5 is called "serialization".

Saxon is a widely used as XSLT processor. It has many versions such as Saxon-B used for basic, Saxon-SA used schema aware, and so on.

Example

In this example, an XML file is converted into an HTML file. The XML file named Books.xml is as follows,

Books.xml

```
<Books>
    <Book>
        <Title>Web Technologies</Title>
        <Author>ABC</Author>
        <Publisher>Rahul Publication</Publisher>
    </Book>
    <Book>
        <Title>Middleware Technologies</Title>
        <Author>XYZ</Author>
```

```
        <Publisher>Rahul Publications</Publisher>
    </Book>
    <Book>
        <Title>Computer Networks</Title>
        <Author>XYZ</Author>
        <Publisher>Rahul Publications
    </Book>
</Book>
```

Books.xslt

The XSLT file, Books.xslt is as follows,

```
<xsl:stylesheet
    xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
    version = "1.0">
    <xsl: template match = "/">
        <html>
            <head>
            <title>List of Books</title>
            </head>
            <body><hl>Books Details</h 1 ><br/>
            </body>
        </html>
    </xsl:template>
    <xsl template match = "Books">
        <h2><xsl:value_of select= "Title 7></h2>
            <xsl:value_of select= "Author"/>
            <xsl:value_of select= "Publisher"/>
        <br/>
    </xsl:template>
    </xsl:stylesheet>
```

To convert the Books.xml into Books.html using Books.xslt, first install either the Java version or .Net version of Saxon processor. The following command can be used for the java version,

Java - jar saxon8.jar -0 Books.xml Books.html Books.xslt.

For the .Net version, the transformation command is,

transform.exe -0 Books.xml Books.html Books.xslt.

The output of either of these commands is the Books.html file, which is as follows,

Books.html

```
<html>
    <head>
        <title>List of Books.</title>
    </head>
    <body>
        <hl>Books Details.</hl>
        <h2>Web Technologies</h2>
            ABC
            Rahul Publications
        <h2>Middleware Technologies<h/2>
            XYZ
            Rahul Publications
        <h2>DBMS<h2>
            PQR
            Rahul Publications
        <br>
    </body>
</html>
```

3. Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) can also be defined with XMT as XSL for a given XML file. After writing the XML file, it needs to be saved using XML extension as shown below,

Book.xml

```
<? xml version = "1.0" encoding = "ISO-8859-1"?>
<CATALOG>
<BQQK>
<BOOJCTITLE>WEB TECHNOLQGIES</<BOOKTITLE>
<AUTHbRNAME>A.A.PUNTAMBAKER<'/AUTHORNAME>
<AMOUNT>300</AMQUNT>
<YEAROFPUBLISH>2014</YEARQFPUBLISH>
</BOOK>
<BOOK>
<BOOKTITLE>DYNAMIC HTML</BOOKTITLE>
<AUTHORNAME>JEFF RULE</AUTHORNAME>
<AMOUNT> 400 AMOUNT>
```

```
<YEAROFPUBLISH>2012</YEAROFPUBLISH>
</BOOK>
```

The following code is written in a file and saved with .css extension,

catalog.css

```
CATALOG
{
    background-color: Red;
    width: 150%;
}
BOOK
{
    display: block;
    margin-bottom: 20pt;
    margin-left: 0.5;
}
BOOKTITLE
{
    colour; Pink
    font-size; 25pt;
}
AUTHORNAME
{
    color: fuchsia;
    font-size: 25pt;
}
AMOUNT, YEAROFPUBLISH
{
    display: Orange;
    color: Purple;
    margin-left: 25pt;
```

The linking of these two files i.e., xml file and css file can be done by the following below code,

```
<?xml-stylesheet type = "text/css" href = "book-catalog.css"?>
```

Q8. How is CSS different of XSL ?

Ans :

S.No.	CSS	S.No.	XSL
1.	CSS work by assigning a set of display properties to an HTML element.	1.	XSL provides means of transforming of XML documents
2.	CSS determines the visual appearance of a page.	2.	XSLT lets us to map a certain pattern in the source document.
3.	It does not change the structure of the document.	3.	It transforms XML into structures such as lists or tables

5.4 HYPERLINKS IN XML

Q9. Explain types of hyperlinks in XML? and Give a basic syntax ?

Ans : **(Imp.)**

There are two basic types of links in an XML document.

1. Internal links
2. External links

Internal links are links to a section within the same document, much like hash URLs and anchor tags in an HTML document.

External links, as the name implies, are links to other, external documents.

1. Internal Links

The two elements used for internal links in THL essays are <ref></ref> and <ptr/>. The difference being that the <ref> element can be wrapped around text including other elements, whereas a <ptr /> element has no text within it but points to another location in the same document. The more common of these is the <ref> element. The only attribute that the <ref> element requires is a target attribute which must be set to the ID of another element, most likely a , within the same document. For example,

.... See the section on <ref target="b2">the Dalai Lama</ref>.

where the div for that section has an id of "b2" as in:

<head>The Dalai Lama</head>

....

This will work for s that are a different section from where the link or <ref> element is. To link to another part of the same section, which appears on the same webpage, the markup is slightly different. To do this one must place an <anchor> tag with a unique ID just above the place where one wants to link to. An example is in Debreczeny article in JIATS 06. In the second section there are links to to mark these up, an <anchor> tag with an id of "img04" was placed above the <figure> tag thus:

<anchor id="img04" />

And the <ref> links set their target to that same ID, but add a rend attribute equal to "samepage", as in:

<reftarget="img04" rend=" samepage"> Fig. 4</ref>

Finally, if one wants to link to another section in the document *and* scroll to a specific anchor in that section, such as is done in Debreczeny section 3, one uses the section ID as the target value and the n attribute for the anchor's id, as in this link:

<ref target="b2" n="img04">

Fig. 4, no. 14</ref>

The <ptr /> has not been implemented for THL because there has been no call for it.

- **Element:** <ref> *{text of link}*</xref>
- **Attributes and Values:**
- **target:** must be the ID value of another element in the same document.
- **rend:** for internal links with in the same section that involve only scrolling to the position on the page, use rend="samepage"
- **n:** for internal links to different sections from where the <ref> tag is located, use the n attribute to indicate an anchor name to scroll to on the target page.

2. External Links

External links are marked up using the <xref type="url"> element. In THL essays all external links open up in popup "lightboxes" *unless* the rend attribute is set to either "_self" or "_top" (to open the content in the same window as the present essay) or "_blank" (to open up the content in a new window). The <xref> element has a doc attribute that is set to the name of a predefined entity, which contains the actual URL for the link. The entities for links are defined in one of two files:

1. http://texts.thlib.org/essays/external-links.dtd
2. http://texts.thlib.org/essays/internal-links.dtd

An example entity declaration is:

<!ENTITYdept-ed-us SYSTEM “http://www2.ed.gov/” NDATA HTML >

This defines an entity by the name of “dept-ed-us” to have the value of “http://www2.ed.gov/” which is HTML. The doc attribute of the <xref> element would be then set to “dept-ed-us”.

5.5 Document Object Model

Q10. Explain about DOM in XML ?

(or)

Explain in detail about XML-DOM

Ans : **(Dec.18, Imp.)**

The Document Object Model (DOM) is a W3C standard. It defines a standard for accessing documents like HTML and XML.

The Document Object Model (DOM) is an application programming interface (API) for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

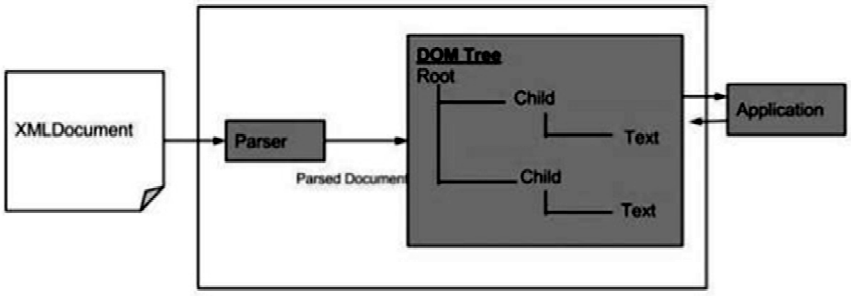
DOM defines the objects and properties and methods (interface) to access all XML elements. It is separated into 3 different parts / levels -

- Core DOM “ standard model for any structured document
- XML DOM “ standard model for XML documents
- HTML DOM “ standard model for HTML documents

XML DOM is a standard object model for XML. XML documents have a hierarchy of informational units called nodes; DOM is a standard programming interface of describing those nodes and the relationships between them.

As XML DOM also provides an API that allows a developer to add, edit, move or remove nodes at any point on the tree in order to create an application.

Following is the diagram for the DOM structure. The diagram depicts that parser evaluates an XML document as a DOM structure by traversing through each node.



Advantages

The following are the advantages of XML DOM.

- XML DOM is language and platform independent.
- XML DOM is traversable - Information in XML DOM is organized in a hierarchy which allows developer to navigate around the hierarchy looking for specific information.
- XML DOM is modifiable - It is dynamic in nature providing the developer a scope to add, edit, move or remove nodes at any point on the tree.

Disadvantages

- It consumes more memory (if the XML structure is large) as program written once remains in memory all the time until and unless removed explicitly.
- Due to the extensive usage of memory, its operational speed, compared to SAX is slower.

Now that we know what DOM means, let's see what a DOM structure is. A DOM document is a collection of nodes or pieces of information, organized in a hierarchy. Some types of nodes may have child nodes of various types and others are leaf nodes that cannot have anything under them in the document structure.

Following is a list of the node types, with a list of node types that they may have as children -

- **Document** - Element (maximum of one), ProcessingInstruction,Comment, Document Type (maximum of one)
- **Document Fragment** - Element, Processing Instruction, Comment, Text, CDATA Section, Entity Reference
- **Entity Reference**-Element, Processing Instruction, Comment, Text, CDATA Section, Entity Reference
- **Element**-Element,Text, Comment, Processing Instruction,CDATASection,Entity Reference
- **Attr** - Text, EntityReference
- **ProcessingInstruction** - No children
- **Comment** - No children
- **Text** - No children
- **CDATASection** " No children
- **Entity** - Element, ProcessingInstruction, Comment,Text, CDATASection, Entity Reference
- **Notation** - No children

Example :

Consider the DOM representation of the following XML document node.xml.

```
<?xml version = "1.0"?>
<Company>
<Employee category = "technical">
<FirstName>Tanmay</FirstName>
<LastName>Patil</LastName>
<ContactNo>1234567890</ContactNo>
</Employee>
<Employee category = "non-technical">
```

<FirstName>Taniya</FirstName>

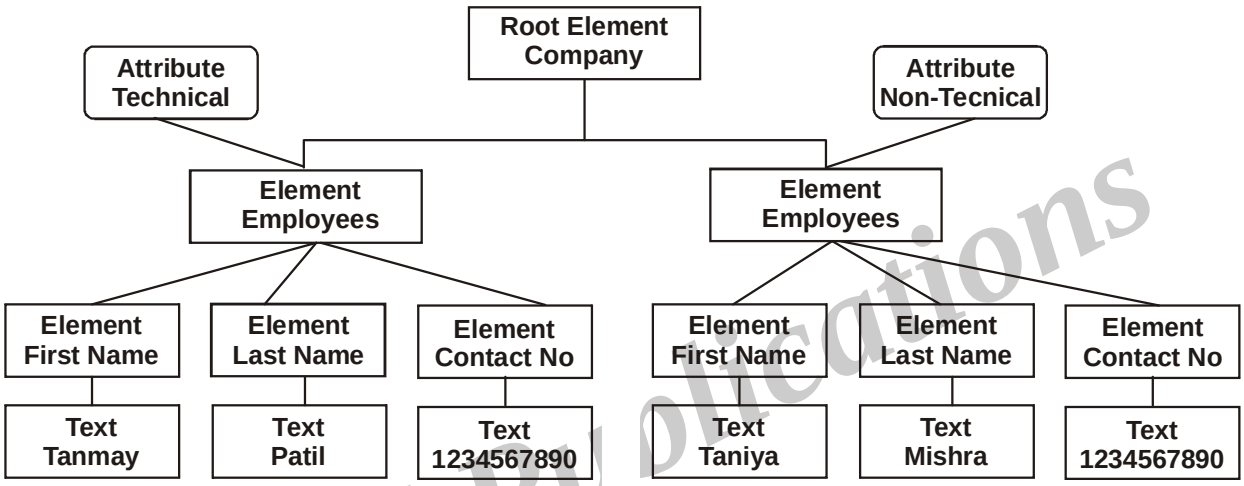
<LastName>Mishra</LastName>

<ContactNo>1234667898</ContactNo>

</Employee>

</Company>

The Document Object Model of the above XML document would be as follows -



From the above flowchart, we can infer -

- Node object can have only one parent node object. This occupies the position above all the nodes. Here it is Company.
- The parent node can have multiple nodes called the child nodes. These child nodes can have additional nodes called the attribute nodes. In the above example, we have two attribute nodes Technical and Non-technical. The attribute node is not actually a child of the element node, but is still associated with it.
- These child nodes in turn can have multiple child nodes. The text within the nodes is called the text node.
- The node objects at the same level are called as siblings.
- The DOM identifies -
  - the objects to represent the interface and manipulate the document.
  - the relationship among the objects and interfaces.

5.6 XML QUERY LANGUAGE

Q11. What is XQuery?explain in detail ?

(OR)

Explain in detail about X-Query (XML Query)

Ans : **(Dec.-20, June-19, Dec.18, Imp.)**

XQuery is a functional language that is used to retrieve information stored in XML format. XQuery can be used on XML documents, relational databases containing data in XML formats, or XML Databases. XQuery 3.0 is a W3C recommendation from April 8, 2014.

The definition of XQuery as given by its official documentation is as follows :

XQuery is a standardized language for combining documents, databases, Web pages and almost anything else. It is very widely implemented. It is powerful and easy to learn. XQuery is replacing proprietary middleware languages and Web Application development languages. XQuery is replacing complex Java or C++ programs with a few lines of code. XQuery is simpler to work with and easier to maintain than many other alternatives.

Characteristics

- **Functional Language** - XQuery is a language to retrieve/querying XML based data.
- **Analogous to SQL** - XQuery is to XML what SQL is to databases.
- **XPath based** - XQuery uses XPath expressions to navigate through XML documents.
- **Universally accepted** - XQuery is supported by all major databases.
- **W3C Standard** - XQuery is a W3C standard.

Benefits of XQuery

- Using XQuery, both hierarchical and tabular data can be retrieved.
- XQuery can be used to query tree and graphical structures.
- XQuery can be directly used to query webpages.
- XQuery can be directly used to build webpages.
- XQuery can be used to transform xml documents.
- XQuery is ideal for XML-based databases and object-based databases. Object databases are much more flexible and powerful than purely tabular databases.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
<bookcategory="JAVA">
<titlelang="en">Learn Java in 24 Hours
</title>
<author>Robert</author>
<year>2005</year>
<price>30.00</price>
</book>
<bookcategory="DOTNET">
<titlelang="en">Learn .Net in 24 hours
</title>
<author>Peter</author>
<year>2011</year>
<price>40.50</price>
</book>
<bookcategory="XML">
```

```
<titlelang="en">Learn XQuery in 24 hours</title>
<author>Robert</author>
<author>Peter</author>
<year>2013</year>
<price>50.00</price>
</book>
<bookcategory="XML">
<titlelang="en">Learn XPath in 24 hours
</title>
<author>Jay Ban</author>
<year>2010</year>
<price>16.50</price>
</book>
</books>
```

books.xqy

```
for $x in doc("books.xml")/books/book
where $x/price>30
return $x/title
```


Q12. State the uses of XML Query.

Ans :

1. It extract data rom database to use it in webservices.
2. It creates reports in a summarized form on data stored in data base.
3. It has the abiity of transforming xml data into XHTML.
4. It uses the extracted data for integrating applications.
5. It divides a single document that performs multiple transactions into multiple documents.
6. It defines syntaxes for creating new xml documents.
7. It searches data relevant text based documents and later compile them to obtain the result.

Short Question and Answers

Q1. Define XML.

Ans :

XML stands for Extensible Markup Language and is a text-based markup language derived from Standard Generalized Markup Language (SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and solutions.

- XML is extensible - XML allows you to create your own self-descriptive tags, or language, that suits your application.
- XML carries the data, does not present it - XML allows you to store the data irrespective of how it will be presented.
- XML is a public standard - XML was developed by an organization called the World Wide Web Consortium (W3C) and is available mas an open standard.

Q2. What is Markup? Is XML is a markup language ?

Ans :

XML is a markup language that defines set of rules for encoding documents in a format that is both human-readable and machine-readable. So *what exactly is a markup language?* Markup is information added to a document that enhances its meaning in certain ways, in that it identifies the parts and how they relate to each other. More specifically, a markup language is a set of symbols that can be placed in the text of a document to demarcate and label the parts of that document.

Following example shows how XML markup looks, when embedded in a piece of text.

```
<message>
<text>Hello, world!</text>
</message>
```

This snippet includes the markup symbols, or the tags such as <message>...</message> and <text>... </text>. The tags <message> and </message> mark the start and the end of the XML code fragment. The tags <text> and </text> surround the text Hello, world!.

Q3. What is XSL ?

Ans :

EXtensible Stylesheet Language Trans formation commonly known as XSLT is a way to transform the XML document into other formats such as XHTML.

XSL is a family of recommendations for defining XML document transformation and presentation. It consists of three parts:

1. XSL Transformations (XSLT)

   a language for transforming XML;

2. The XML Path Language (XPath)

   an expression language used by XSLT (and many other languages) to access or refer to parts of an XML document;

3. XSL Formatting Objects (XSL-FO)

   an XML vocabulary for specifying formatting semantics.

An XSLT stylesheet specifies the presentation of a class of XML documents by describing how an instance of the class is transformed into an XML document that uses a formatting vocabulary, such as (X)HTML or XSL-FO. For a more detailed explanation of how XSL works, see the What Is XSL page.

Q4. What is External Links?

Ans :

External links are marked up using the <xref type = "url"> element. In THL essays all external links open up in popup "lightboxes" *unless* the rend attribute is set to either "_self" or "_top" (to open the content in the same window as the present essay) or "_blank" (to open up the content in a new window). The <xref> element has a doc attribute that is set to the name of a predefined entity, which contains the actual URL for the link. The entities for links are defined in one of two files:

1. http://texts.thlib.org/essays/external-links.dtd
2. http://texts.thlib.org/essays/internal-links.dtd

An example entity declaration is:

<!ENTITYdept-ed-us SYSTEM "http://www2.ed.gov/" NDATA HTML >

This defines an entity by the name of "dept-ed-us" to have the value of "http://www2.ed.gov/" which is HTML. The doc attribute of the <xref> element would be then set to "dept-ed-us".

Q5. Advantages of XML DOM.

Ans :

The following are the advantages of XML DOM.

- XML DOM is language and platform independent.
- XML DOM is traversable - Information in XML DOM is organized in a hierarchy which allows developer to navigate around the hierarchy looking for specific information.
- XML DOM is modifiable - It is dynamic in nature providing the developer a scope to add, edit, move or remove nodes at any point on the tree.

Q6. What are the Benefits of XQuery.

Ans :

- Using XQuery, both hierarchical and tabular data can be retrieved.
- XQuery can be used to query tree and graphical structures.
- XQuery can be directly used to query webpages.
- XQuery can be directly used to build webpages.
- XQuery can be used to transform xml documents.
- XQuery is ideal for XML-based databases and object-based databases. Object databases are much more flexible and powerful than purely tabular databases.

Q7. State the uses of XQuery.

Ans :

1. It extract data rom database to use it in webservices.
2. It creates reports in a summarized form on data stored in data base.
3. It has the abiity of transforming xml data into XHTML.
4. It uses the extracted data for integrating applications.
5. It divides a single document that performs multiple transactions into multiple documents.
6. It defines syntaxes for creating new xml documents.
7. It searches data relevant text based documents and later compile them to obtain the result.

Q8. Define PI and entity.

Ans :

The processing instructions allow the XML documents to contain the special instructions which pass the parameters to the application. The special instructions the application how the XML document can be interpreted. The processing instructions start with <? andend with?>. XML parsers, rather than caring these instructions, they pass the parameters to the underlying application. These instructions are not the part of the document. Consider the processing instruction given below.

<?ml-style sheet bref="filename.xsl" type="text/xsl"?>

This instruction indicates that the document must be transformed by using "filename.xsl" stylesheet.

Entity

Entity is the statement that is used to define DTD entities in DTD or XML document. It repeats huge data block of text that needs to be stored i separate. files. It provides abbreviated entry to use in XML document.

Q9. XML attributes.

Ans :

XML attributes. Attributes are part of XML elements. An element can have multiple unique attributes. Attribute gives more information about XML elements. To be more precise, they define properties of elements. An XML attribute is always a name-value pair.

Syntax

An XML attribute has the following syntax "

<element-name attribute1 attribute2 >

....content..

< /element-name>

where attribute1 and attribute2 has the following form "

name = "value"

value has to be in double (" ") or single (' ') quotes. Here, attribute1 and attribute2 are unique attribute labels.

Attributes are used to add a unique label to an element, place the label in a category, add a Boolean flag, or otherwise associate it with some string of data. Following example demonstrates the use of attributes

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE garden [
  <!ELEMENT garden (plants)*>
  <!ELEMENT plants (#PCDATA)>
    <!ATTLIST plants category CDATA #REQUIRED>
]>
<garden>
  <plants category = "flowers" />
  <plants category = "shrubs">
  </plants>
</garden>
```

Attributes are used to distinguish among elements of the same name, when you do not want to create a new element for every situation. Hence, the use of an attribute can add a little more detail in differentiating two or more similar elements.

In the above example, we have categorized the plants by including attribute category and assigning different values to each of the elements. Hence, we have two categories of plants, one flowers and other shrubs. Thus, we have two plant elements with different attributes.

You can also observe that we have declared this attribute at the beginning of XML.

Choose the Correct Answer

1. XML is an Extension of ____. [b]
   (a) GML (b) SGML
   (c) HTML (d) CSS
2. ____ is used for off loading and Re loading of Databases. [c]
   (a) HTML (b) CSS
   (c) XML (d) None
3. XML supports ____ tags ? [b]
   (a) Predefined (b) User defined
   (c) Both (d) None
4. XML document transformations & Presentations Represented by ? [a]
   (a) XSLT (b) X path
   (c) XSLFO (d) XSL
5. Name and properties are differ by ? [a]
   (a) colon (:) (b) semicolm (i)
   (c) astrik (*) (d) slasls (/)
6. I it Represents [c]
   (a) > (b) > =
   (c) < (d) < =
7. DOM can have ____ parts [d]
   (a) core DOM (b) XML-DOM
   (c) HTML DOM (d) ALL
8. XML-DOM is ____ [d]
   (a) plat form independent (b) traversable
   (c) dynamic in nature (d) all
9. XQuery uses ____ to navigate through XML-documents [a]
   (a) X path (b) UTF-8
   (c) version (d) all
10. XQuery can be used to build ____ [b]
    (a) only Query (b) web pages
    (c) contents (d) none

Fill in the blanks

1. XML stands for _____ .
2. XML is a Extenssion of _____ .
3. SGML stands for _____ .
4. XML can be used for _____ of Data Bases.
5. XML is _____ language.
6. XML Root & child Elements should be _____ .
7. XSL stands for _____ .
8. How many kinds of hyperlinks supported by XML _____ .
9. XML - DOM contains parts _____ .
10. _____ Responsible for interaction with data Base.

ANSWERS

1. Extensible markup Language
2. SGML
3. Standard Generalized Markup Long
4. Off loading & reloading
5. Case sensitive
6. Differ
7. Extensible style sheet Language
8. 2
9. 3
10. XQuery

One Mark Answers

1. XML

Ans :

XML is Extensible , It allows you to create your own self Descriptive tags & applications.

2. XSLT

Ans :

Extensible style sheet language, commonly known as XSLT. It is a way to transform the XML document in to other formats.

3. XML DOM

Ans :

DOM Defines the objects and properties, methods to access all XML Elements.

4. XQuery

Ans :

Is a functional Language that is used to retrive information stored in XML-formats.

5. XML reference

Ans :

References usually allow you to add (or) include additional text (or) markup in an XML document.

LAB PROGRAMS

1. Create a Webpage with dynamic functionality using a Client Side Program.

```
<html>
<head>
    <title>Student Registration Form</title>
    <script type="text/javascript" src="validate.js"></script>
</head>
<body>
<center>
<Font color=Blue><h1> RAHUL PUBLICATIONS</h1></font>
<h4> Complete Solution for Computer Technology.....</h4>
</center>
<center>
<Font color=orange><h2> Student Form</h2></font>
</center>
<form action="student.php" method="post" name="StudentRegistration"onsubmit="return
(validate());">
    <table cellpadding="2" width="20%" bgcolor="99FFFF" align="center"
    cellspacing="2">
        <tr>
            <td colspan=2>
            <center><font size=4><b>Student Registration Form</b></font></center>
            </td>
        </tr>
        <tr>
            <td>Name</td>
            <td><input type="text" name="studentname" id="studentname" size="30"></td>
        </tr>
        <tr>
            <td>Father Name</td>
            <td><input type="text" name="fathername" id="fathername"
            size="30"></td>
        </tr>
        <tr>
            <td>Postal Address</td>
            <td><input type="text" name="paddress" id="paddress" size="30"></td>
        </tr>
```

```
<tr>
    <td>Personal Address</td>
    <td><input type="text" name="personaladdress"id="personaladdress" size="30">
    </td>
</tr>

<tr>
    <td>Sex</td>
    <td>
        <input type="radio" name="sex" value="male" size="10">Male
        <input type="radio" name="sex" value="Female" size="10">Female
    </td>
</tr>

<tr>
<td>City</td>
    <td>
    <select name="City">
        <option value="-1" selected>select..</option>
        <option value="New Delhi">NEW DELHI</option>
        <option value="Mumbai">MUMBAI</option>
        <option value="Goa">GOA</option>
        <option value="Patna">PATNA</option>
        <option value="Patna">Hyderabad</option>
    </select></td>
</tr>
<tr>
    <td>Course</td>
    <td><select name="Course">
    <option value="-1" selected>select..</option>
    <option value="B.Tech">B.TECH</option>
    <option value="MCA">MCA</option>
    <option value="MBA">MBA</option>
    <option value="BCA">BCA</option>
    </select></td>
</tr>
<tr>
    <td>District</td>
    <td><select name="District">
    <option value="-1" selected>select..</option>
    <option value="Nalanda">MAHARASTRA</option>
```

```
                <option value="UP">UP</option>
                <option value="Goa">KERALA</option>
                <option value="Patna">MADHYAPRADESH</option>
                <option value="Patna">TS/AP</option>
            </select></td>
            </tr>
            <tr>
            <td>State</td>
                <td><select Name="State">
                <option value="-1" selected>select..</option>
                <option value="New Delhi">NEW DELHI</option>
                <option value="Mumbai">MUMBAI</option>
                <option value="Goa">GOA</option>
                <option value="Bihar">BIHAR</option>
                </select></td>
            </tr>
            <tr>
                <td>PinCode</td>
                <td><input type="text" name="pincode" id="pincode" size="30"></td>
            </tr>
            <tr>
                <td>EmailId</td>
                <td><input type="text" name="emailid" id="emailid" size="30"></td>
            </tr>
            <tr>
                <td>DOB</td>
                <td><input type="text" name="dob" id="dob" size="30"></td>
            </tr>
            <tr>
                <td>MobileNo</td>
                <td><input type="text" name="mobileno" id="mobileno" size="30"></td>
            </tr>
            <tr>
                <td>
                    <input type="reset">
                </td>
                <td colspan="2">
                    <input type="submit" value="Submit Form" />
                </td>
            </tr>
        </table>
</form>
</body>
</html>
```

Output

2. Create a Webpage with dynamic functionality using a Server Side Program.

```
<html>
<head>
    <title>Student Registration Form</title>
    <script type="text/javascript" src="validate.js"> </script>
</head>
<body>
<center>
<Font color=Blue><h1> RAHUL PUBLICATIONS</h1></font>
<h4> Complete Solution for Computer Technology.....</h4>
</center>
<center>
<Font color=orange><h2> Student Form</h2></font>
</center>
<form action="student.php" method="post" name="StudentRegistration" onsubmit="return
(validate());">
    <table cellpadding="2" width="20%" bgcolor="99FFFF" align="center"
    cellspacing="2">
        <tr>
            <td colspan=2>
            <center><font size=4><b>Student Registration Form</b></font></center>
            </td>
        </tr>
```

```
<tr>
    <td>Name</td>
    <td><input type="text" name="studentname" id="studentname" size="30"></td>
</tr>

<tr>
    <td>Father Name</td>
    <td><input type="text" name="fathername" id="fathername"
    size="30"></td>
</tr>
<tr>
    <td>Postal Address</td>
    <td><input type="text" name="paddress" id="paddress" size="30"></td>
</tr>
<tr>
    <td>Personal Address</td>
    <td><input type="text" name="personaladdress"id="personaladdress" size="30">
    </td>
</tr>
<tr>
    <td>Sex</td>
    <td>
        <input type="radio" name="sex" value="male" size="10">Male
        <input type="radio" name="sex" value="Female" size="10">Female
    </td>
</tr>
<tr>
<td>City</td>
    <td>
    <select name="City">
        <option value="-1" selected>select..</option>
        <option value="New Delhi">NEW DELHI</option>
        <option value="Mumbai">MUMBAI</option>
        <option value="Goa">GOA</option>
        <option value="Patna">PATNA</option>
        <option value="Patna">Hyderabad</option>
    </select></td>
</tr>
<tr>
    <td>Course</td>
    <td><select name="Course">
```

```
        <option value="-1" selected>select..</option>
        <option value="B.Tech">B.TECH</option>
        <option value="MCA">MCA</option>
        <option value="MBA">MBA</option>
        <option value="BCA">BCA</option>
        </select></td>
</tr>
<tr>
        <td>District</td>
        <td><select name="District">
        <option value="-1" selected>select..</option>
        <option value="Nalanda">MAHARASTRA</option>
        <option value="UP">UP</option>
        <option value="Goa">KERALA</option>
        <option value="Patna">MADHYAPRADESH</option>
        <option value="Patna">TS/AP</option>
</select></td>
</tr>
<tr>
<td>State</td>
        <td><select Name="State">
        <option value="-1" selected>select..</option>
        <option value="New Delhi">NEW DELHI</option>
        <option value="Mumbai">MUMBAI</option>
        <option value="Goa">GOA</option>
        <option value="Bihar">BIHAR</option>
        </select></td>
</tr>
<tr>
        <td>PinCode</td>
        <td><input type="text" name="pincode" id="pincode" size="30"></td>
</tr>
<tr>
        <td>EmailId</td>
        <td><input type="text" name="emailid" id="emailid" size="30"></td>
</tr>
<tr>
        <td>DOB</td>
        <td><input type="text" name="dob" id="dob" size="30"></td>
</tr>
```

```
            <tr>
                <td>MobileNo</td>
                <td><input type="text" name="mobileno" id="mobileno" size="30"></td>
            </tr>
            <tr>
                <td>
                    <input type="reset">
                </td>
                <td colspan="2">
                    <input type="submit" value="Submit Form" />
                </td>
            </tr>

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

Validate Program on Server Side(validate.js)

```
function validate()
    {
    if(document.StudentRegistration.studentname.value == "" )
    {
        alert( "Please provide your Name!" );
        document.StudentRegistration.studentname.focus() ;
        return false;
    }
    if(document.StudentRegistration.fathername.value == "" )
    {
        alert( "Please provide your Father Name!" );
        document.StudentRegistration.fathername.focus() ;
        return false;
    }
    if(document.StudentRegistration.paddress.value == "" )
    {
        alert( "Please provide your Postal Address!" );
        document.StudentRegistration.paddress.focus() ;
    return false;
    }
    if(document.StudentRegistration.personaladdress.value == "" )
```

```
{
alert( “Please provide your Personal Address!” );
document.StudentRegistration.personaladdress.focus() ;
return false;
}
if ( ( StudentRegistration.sex[0].checked = = false ) && ( StudentRegistration.sex[1].checked = = false ) )
{
alert ( “Please choose your Gender: Male or Female” );
return false;
}
if(document.StudentRegistration.City.value = = “-1” )
{
alert( “Please provide your City!” );
document.StudentRegistration.City.focus() ;
return false;
}
if(document.StudentRegistration.Course.value = = “-1” )
{
alert( “Please provide your Course!” );
return false;
}
if(document.StudentRegistration.District.value = = “-1” )
{
alert( “Please provide your Select District!” );
return false;
}
if(document.StudentRegistration.State.value = = “-1” )
{
alert( “Please provide your Select State!” );
return false;
}
if(document.StudentRegistration.pincode.value = = “” ||
isNaN(document.StudentRegistration.pincode.value) ||
document.StudentRegistration.pincode.value.length != 6 )
{
      alert( “Please provide a pincode in the format ######.” );
      document.StudentRegistration.pincode.focus() ;
      return false;
}
```

```
        var email = document.StudentRegistration.emailid.value;
        atpos = email.indexOf("@");
        dotpos = email.lastIndexOf(".");
        if (email == "" || atpos< 1 || ( dotpos - atpos< 2 ))
    {
        alert("Please enter correct email ID")
        document.StudentRegistration.emailid.focus() ;
        return false;
    }
    if(document.StudentRegistration.dob.value == "" )
    {
        alert( "Please provide your DOB!" );
        document.StudentRegistration.dob.focus() ;
        return false;
    }
        if(document.StudentRegistration.mobileno.value == "" ||
        isNaN(document.StudentRegistration.mobileno.value) ||
        document.StudentRegistration.mobileno.value.length != 10 )
    {
        alert( "Please provide a Mobile No in the format 123." );
        document.StudentRegistration.mobileno.focus() ;
        return false;
    }
    return( true );
}
```

Output


Student Registration Form
This page says
Please enter correct email ID
OK
Student Form
Student Registration Form
Name
Subba Reddy
Father Name
Rama Subba Reddy
Postal Address
DSNR
Personal Address
DSNR
Sex
Male
Female
City
Hyderabad
Course
B.TECH
District
TS/AP
State
GOA
PinCode
500060
EmailId
DOB
MobileNo
Reset
Submit Form
Type here to search

FACULTIES OF COMMERCE

B.Com. V - Semester (CBCS) Examination

November / December - 2020

WEB TECHNOLOGY

(Common Paper for Computers / Computer Applications Courses)

Time : 2 Hours] **[Max. Marks : 80**

PART - A (4 × 5 = 20 Marks)

Note: Answer any Four questions

		ANSWERS
1.	Write short notes on ordered lists along with its attributes.	**(Unit-I, SQA-16)**
2.	What is DHTML? What are the technologies included in DHTML?	**(Unit-II, SQA-1)**
3.	Write about for loop in java script.	**(Unit-III, SQA-7)**
4.	Write short notes on Event and Event handlers in java script.	**(Unit-IV, SQA-6, 7)**
5.	Write short notes on XML attributes.	**(Unit-V, SQA-9)**
6.	Write short notes on <marquee> tag.	**(Unit-I, SQA-17)**
7.	What is a Form in html?	**(Unit-I, SQA-6)**
8.	Write about Arithmetic operations in java script.	**(Unit-II, SQA-12)**

PART - B (4 × 15 = 60 Marks)

Note: Answer any four questions.

9.	Explain in detail about tables in html and its associated tags.	**(Unit-I, Q.No. 23)**
10.	Explain in detail about Frames in html with suitable example.	**(Unit-I, Q.No. 28)**
11.	Explain in detail about various types of CSS.	**(Unit-II, Q.No. 9)**
12.	Explain in detail about filters and transitions.	**(Unit-II, Q.No. 14)**
13.	Explain in detail about how to create user defined functions in Java Script with example.	**(Unit-III, Q.No. 17)**
14.	Explain in detail about Data Object in Java Script.	**(Unit-III, Q.No. 9)**
15.	Explain about Mouse related events with examples.	**(Unit-IV, Q.No. 14, 15, 16, 17)**
16.	Explain in detail Keyboard related events with examples.	**(Unit-IV, Q.No. 11, 12)**
17.	Explain in detail about features of XML.	**(Unit-V, Q.No. 3)**
18.	Explain in detail about XML Query language.	**(Unit-V, Q.No. 11)**

FACULTIES OF COMMERCE

B.Com. V Semester (CBCS) Examination

June / July - 2019

WEB TECHNOLOGY

Time : 3 Hours] **[Max. Marks : 80**

PART - A (5 × 4 = 20 Marks)

Note: Answer any FIVE of the following questions

ANSWERS

1. HTML is a markup language - Explain. **(Unit-I, SQA-1)**
2. What is Cascading Style Sheets (CSS)? List types of Style Sheets. **(Unit-II, SQA-5, 6, 7, 8)**
3. Explain DO... While loop in Java Script. **(Unit-III, SQA-11)**
4. Explain onload Event. **(Unit-IV, SQA-3)**
5. Advantages of XML DOM. **(Unit-V, SQA-5)**
6. What is the use of Anchor tag? **(Unit-I, SQA-8)**
7. Explain OndbL click event. **(Unit-IV, SQA-2)**
8. Explain types of lists in HTML. **(Unit-I, SQA-9)**

PART - B (5 × 12 = 60 Marks)

Note: Answer ALL the questions

9. (a) What is meant by Form? Explain the elements of forms. **(Unit-I, Q.No. 13, 15)**

OR

(b) What is a Table? Explain the attributes of <table> in HTML. **(Unit-I, Q.No. 23, 24)**

10. (a) Explain briefly the difference between HTML and DHTML. **(Unit-II, Q.No. 4)**

OR

(b) Explain the concept of Filters and Transitions. **(Unit-II, Q.No. 14)**

11. (a) Explain the Data types supported by Java Script. **(Unit-III, Q.No. 9)**

OR

(b) What is meant by an Operator? Explain the types of operators supported by Java script. **(Unit-III, Q.No. 12)**

12. (a) What is an Event? Explain the Key Board events. **(Unit-IV, Q.No. 3, 11, 12)**

OR

(b) Explain in detail ON-ERROR event with a suitable program. **(Unit-IV, Q.No. 9)**

13. (a) Explain about XML - Style Sheets. **(Unit-V, Q.No. 7)**

OR

(b) Explain in detail about X-Query (XML query). **(Unit-V, Q.No. 11)**

FACULTIES OF COMMERCE

B.Com. III Year - V Semester (CBCS) Examination

Nov. / Dec. - 2018

WEB TECHNOLOGY

Time : 3 Hours] **[Max. Marks : 80**

PART - A (5 × 4 = 20 Marks)

Note: Answer any FIVE of the following questions

ANSWERS

1. Hyper Links in HTML. **(Unit-I, SQA-8)**
2. Explain filters and transactions for creating multimedia effects. **(Unit-II, SQA-10)**
3. Explain for Loop in Java script. **(Unit-II, SQA-7)**
4. Ondrag drop event, Explain. **(Unit-IV, SQA-5)**
5. What is XSL? **(Unit-V, SQA-3)**
6. Explain On load event. **(Unit-IV, SQA-3)**
7. Explain types of lists in HTML. **(Unit-I, SQA-9)**
8. How do you insert an image in a Web Page? **(Unit-I, SQA-15)**

PART - B (5 × 12 = 60 Marks)

Note: Answer ALL the questions

9. (a) What is a table? Explain the attributes of <table> in HTML. **(Unit-I, Q.No. 23, 24)**

OR

(b) Define a frame. Explain the attributes supported by the <frame> tag briefly. **(Unit-I, Q.No. 28, 30)**

10. (a) What is CSS? Explain the types of CSS. **(Unit-II, Q.No. 5, 9)**

OR

(b) Give a brief introduction about DHTML, its features. **(Unit-II, Q.No. 1)**

11. (a) Explain in detail how Java script supports functions. **(Unit-III, Q.No. 17)**

OR

(b) Explain the data types supported by Java script. **(Unit-III, Q.No. 12)**

12. (a) List and explain Mouse – events. **(Unit-IV, Q.No. 14, 15, 16, 17)**

OR

(b) List and explain key board events. **(Unit-IV, Q.No. 11, 12)**

13. (a) Explain in detail about XML – DOM. **(Unit-V, Q.No. 10)**

OR

(b) Explain in detail about X-Query (XML Query). **(Unit-V, Q.No. 11)**

KAKATIYA UNIVERSITY

FACULTY OF COMMERCE AND BUSINESS MANAGEMENT

B.Com. VI - Semester Examination

November - 2020

WEB TECHNOLOGY

Paper - V

Time : 2 Hours] [Max. Marks 60

Answer any four questions. Including Q.No. 1, which is compulsory

(5 × 4 = 20 Marks)

ANSWERS

1. **Attempt any three questions in short-form :**

   (a) What is Markup Language ? **(Unit-I, SQA-1)**

   (b) What do you mean by <BR> tag. **(Unit-I, SQA-4)**

   (c) What is CSS? **(Unit-II, SQA-6)**

   (d) What are the data types of JavaScript Language ? **(Unit-III, Q.No. 9)**

   (e) Define event. **(Unit-IV, SQA-6)**

   (f) What is event handling in JavaScript ? **(Unit-III, Q.No. 24)**

2. Explain the basic structure of HTML, document. **(Unit-I, Q.No. 6)**

3. What is a form ? Explain <FORM> and <INPUT> tags used in HTML forms. **(Unit-I, Q.No. 13, 15)**

4. What is DHTML? Explain various components of DHTM. **(Unit-II, Q.No. 4)**

5. Explain various CSS embedding techniques. **(Unit-II, Q.No. 9)**

6. Explain the various conditional statements used in JavaScript. **(Unit-III, Q.No. 14)**

7. What is an object and explain various member functions of math object and data object ? **(Unit-III, Q.No. 18, 19)**

8. Explain with suitable examples :

   (a) onClick and onDbiClick **(Unit-IV, Q.No. 6)**

   (b) onLoad and onUnload **(Unit-IV, Q.No. 13)**

   (c) onFocus **(Unit-IV, Q.No. 10)**

9. Explain with suitable examples :

   (a) onAbort **(Unit-IV, Q.No. 5)**

   (b) onError **(Unit-IV, Q.No. 9)**

   (c) onSubmit **(Unit-IV, Q.No. 22)**

   (d) OnMouseOver and onMouseOut. **(Unit-IV, Q.No. 16, 17)**

10. What is XML document ? What are the different rules to write XML document? **(Unit-V, Q.No. 5,6)**

MAHATMA GANDHI UNIVERSITY

FACULTY OF COMMERCE AND BUSINESS MANAGEMENT

B.Com. (CBCS) III Year V - Semester Examination

November / December - 2019

WEB TECHNOLOGY

Time : 3 Hours] [Max. Marks 60

PART-A (5 × 3 = 15 MARKS)

(Short Answer Type)

ANSWERS

I. **Answer all the questions.**

1. Explain Anchor Tag. **(Unit-I, SQA-8)**
2. What is Dynamic Web Page ? **(Unit-II, SQA-1)**
3. What is Client side JavaScript ? **(Unit-III, SQA-2)**
4. Explain OnClick event. **(Unit-IV, Q.No. 6)**
5. What is the purpose of XML ? **(Unit-V, Q.No. 1)**

PART-B (5 × 9 = 45 MARKS)

(Essay Answer Type)

II. Answer all the questions.

6. (a) What is Frame ? Explain working the frames with examples. **(Unit-I, Q.No. 28)**

OR

(b) Explain different Form controls in HTML. **(Unit-I, Q.No. 15)**

7. (a) What is Style Sheet ? Explain different types of CSS. **(Unit-II, Q.No. 5, 9)**

OR

(b) Explain different types of Filters used in Multimedia effects. **(Unit-II, Q.No. 14)**

8. (a) Explain various Data types in JavaScript. **(Unit-III, Q.No. 9)**

OR

(b) What is Array ? Write a program to find the sum of a numbers of an Array. **(Unit-III, Q.No. 20)**

9. (a) Explain various Mouse Events with examples. **(Unit-IV, Q.No. 14, 15, 16, 17)**

OR

(b) Write notes on,

(i) OnAbort **(Unit-IV, Q.No. 5)**

(ii) OnSubmit **(Unit-IV, Q.No. 22)**

(iii) OnUnload. **(Unit-IV, Q.No. 23)**

10. (a) Discuss on XML Document Object Model. **(Unit-V, Q.No. 10)**

OR

(b) Explain XML Style Sheets. **(Unit-V, Q.No. 7)**

MAHATMA GANDHI UNIVERSITY

FACULTY OF COMMERCE AND BUSINESS MANAGEMENT

B.Com. (CBCS) III Year V - Semester Examination

May / June - 2019

WEB TECHNOLOGY

Time : 3 Hours] [Max. Marks 60

PART-A (5 × 3 = 15 MARKS)
(Short Answer Type)

ANSWERS

I. **Answer all the questions.**

1. What is Frame HTML and explain with examples ? **(Unit-I, Q.No. 28)**
2. Advantages of CSS. **(Unit-II, Q.No. 5)**
3. Define functions in Javascripts. **(Unit-III, Q.No. 17)**
4. Write about onSubmit event handlers. **(Unit-IV, Q.No. 9)**
5. Write about XML DOM. **(Unit-V, Q.No. 10)**

PART-B (5 × 9 = 45 MARKS)
(Essay Answer Type)

II. Answer all the questions.

6. (a) What are the guidelines to follow while designing a site ? **(Unit-I, Q.No. 7)**

OR

(b) What are hyperlinks? Explain about anchor tag. **(Unit-I, Q.No. 21)**

7. (a) Explain style classes with example program. **(Unit-II, Q.No. 9)**

OR

(b) Explain the dynamic changes to text, style and graphics. **(Unit-II, Q.No. 11, 12, 13)**

8. (a) Explain different types of operators available in Java Script. **(Unit-III, Q.No. 12)**

OR

(b) What is an array ? Explain how to create array and how to add and access elements in array. **(Unit-III, Q.No. 20)**

9. (a) Discuss in brief about onResize and onSelect. **(Unit-IV, Q.No. 20, 21)**

OR

(b) Explain about onError and onFocus. **(Unit-IV, Q.No. 9, 10)**

10. (a) Explain the rules of XML syntax. **(Unit-V, Q.No. 5)**

OR

(b) Explain the XML DOM. **(Unit-V, Q.No. 10)**

MAHATMA GANDHI UNIVERSITY

FACULTY OF COMMERCE AND BUSINESS MANAGEMENT

B.Com. (CBCS) III Year V - Semester Examination

November / December - 2018

WEB TECHNOLOGY

Time : 3 Hours] [Max. Marks 60

PART-A (5 × 3 = 15 MARKS)

(Short Answer Type)

ANSWERS

I. **Answer all the questions.**

1. Write about different formatting tags. **(Unit-I, Q.No. 11)**
2. Write a short note on DHTML. **(Unit-II, SQA-1)**
3. Explain about DOML. **(Unit-III, SQA-8)**
4. Write about different types of event handlers. **(Unit-IV, SQA-1)**
5. Define XML styles sheets. **(Unit-V, Q.No. 7)**

PART-B (5 × 9 = 45 MARKS)

(Essay Answer Type)

II. Answer all the questions.

6. (a) Define forms. Explain various form control in HTML. **(Unit-I, Q.No. 13, 15)**

OR

(b) Explain how a basic table is created using in HTML. **(Unit-I, Q.No. 23)**

7. (a) What is CSS ? Explain different types of CSS. **(Unit-II, Q.No. 5, 9)**

OR

(b) Explain about different types of filters and transitions in DHTML. **(Unit-II, Q.No. 14)**

8. (a) Define data types. Explain various data types available in Java Script. **(Unit-III, Q.No. 9)**

OR

(b) What is Array ? Explain how to create array and how to add and access elements. **(Unit-III, Q.No. 20)**

9. (a) Discuss in brief about onSubmit and onUnload. **(Unit-IV, Q.No. 22, 13)**

OR

(b) Write a program onMove and onReset. **(Unit-IV, Q.No. 15, 19)**

10. (a) Write about hyperlinks in XML. **(Unit-V, Q.No. 9)**

OR

(b) Explain in detail about XML Query. **(Unit-V, Q.No. 11)**

FACULTIES OF COMMERCE

B.Com. II Year - IV Semester (CBCS) Examination

Model Paper - I

WEB TECHNOLOGIES

(Common Paper for Computers / Computer Applications Courses)

Time : 1½ Hours] **[Max. Marks : 50**

PART - A (5 × 2 = 10 Marks)

Note: Answer any FIVE of the following questions

ANSWERS

1. Hyperlinks in HTML. **(Unit-I, SQA-8)**
2. HTML is a Markup Language - Explain. **(Unit-I, SQA-1)**
3. Write some Advantages of DHTML. **(Unit-II, SQA-3)**
4. List out the advantages of Java Script. **(Unit-III, SQA-1)**
5. What is the HTML DOM. **(Unit-III, SQA-8)**
6. Illustrate the differences between onMouseMove and onMouseOver event handler. **(Unit-IV, SQA-8)**
7. What is an event ? **(Unit-IV, SQA-6)**
8. State the uses of XQuery. **(Unit-V, SQA-7)**

PART - B (5 × 8 = 40 Marks)

Note: Answer ALL the questions

9. (a) Write a program to demonstrate rowspan and colspan attributes in table shown below.

Name	Marks		
	FM	FIT	BOM
Rahul	68	68	75
Rakesh	60	69	60
Ravi	68	66	60

(Unit-I, Q.No. 27)

OR

(b) List and explain web designing principles. **(Unit-I, Q.No. 7)**

10. (a) What is CSS? Explain the different types of CSS with an examples. **(Unit-II, Q.No. 5, 9)**

OR

(b) Explain briefly dynamic changes to text, and graphics. **(Unit-II, Q.No. 13)**

11. (a) What are the data types supported by javascript? **(Unit-III, Q.No. 9)**

OR

(b) How java script supports conditional statements? Explain with example programs. **(Unit-III, Q.No. 14)**

12. (a) Explain about On unloadevent and On submit event? **(Unit-IV, Q.No. 23, 22)**

OR

(b) Explain about onmouse move and onmouse down event. **(Unit-IV, Q.No. 15, 14)**

13. (a) Distinguish between XML and HTML. **(Unit-V, Q.No. 4)**

OR

(b) Explain types of hyperlinks in xml? and Give a basic syntax ? **(Unit-V, Q.No. 9)**

FACULTIES OF COMMERCE

B.Com. II Year - IV Semester (CBCS) Examination

Model Paper - II

WEB TECHNOLOGIES

(Common Paper for Computers / Computer Applications Courses)

Time : 1½ Hours] **[Max. Marks : 50**

PART - A (5 × 2 = 10 Marks)

Note: Answer any FIVE of the following questions

ANSWERS

1. How can you format the text in HTML ? **(Unit-I, SQA-5)**
2. How can you insert an image in to your web document ? **(Unit-I, SQA-15)**
3. What is Event Handling. **(Unit-II, SQA-9)**
4. Explain the do...while Loop in Java. **(Unit-III, SQA-11)**
5. List and Explain Bitwise Operators in Java. **(Unit-III, SQA-5)**
6. Explain Onloadevent. **(Unit-IV, SQA-3)**
7. What is XSL ? **(Unit-V, SQA-3)**
8. What is Table ? **(Unit-I, SQA-11)**

PART - B (5 × 8 = 40 Marks)

Note: Answer ALL the questions

9. (a) What is meant by List ? How we can create the list in html ? **(Unit-I, Q.No. 22)**

OR

(b) What is meant by Frames ? Explain disadvantages of frames? **(Unit-I, Q.No. 28)**

10. (a) What are the differences between HTML and DHTML? **(Unit-II, Q.No. 4)**

OR

(b) Explain filters and transactions for multimedia effects. **(Unit-II, Q.No. 14)**

11. (a) Explain in detail how Javascript Supports function. **(Unit-III, Q.No. 17)**

OR

(b) Define JavaScript. What are the advantages and limitations of java script? **(Unit-III, Q.No. 1, 2)**

12. (a) Define Event. Explain the various attributes supported by the JavaScript. **(Unit-IV, Q.No. 3)**

OR

(b) Explain briefly about key board events. **(Unit-IV, Q.No. 11, 12)**

13. (a) Explain about DOM in XML ? **(Unit-V, Q.No. 10)**

OR

(b) Explain in detail about X-Query (XML Query). **(Unit-V, Q.No. 11)**

FACULTIES OF COMMERCE

B.Com. II Year - IV Semester (CBCS) Examination

Model Paper - III

WEB TECHNOLOGIES

(Common Paper for Computers / Computer Applications Courses)

Time : 1½ Hours] **[Max. Marks : 50**

PART - A (5 × 2 = 10 Marks)

Note: Answer any FIVE of the following questions

ANSWERS

1. What is Table ? **(Unit-I, SQA-11)**
2. Define Form. **(Unit-I, SQA-6)**
3. Write some Advantages of DHTML. **(Unit-II, SQA-3)**
4. What is Event Handling. **(Unit-II, SQA-9)**
5. What is client side scripting. **(Unit-III, SQA-2)**
6. Define Date-Object. **(Unit-III, SQA-9)**
7. Define Event handling Mechanism? **(Unit-IV, SQA-1)**
8. Define XML. **(Unit-V, SQA-1)**

PART - B (5 × 8 = 40 Marks)

Note: Answer ALL the questions

9. (a) Explain different types of HTML tags. **(Unit-I, Q.No. 5)**

OR

(b) What is mean by Nested tables? How can you create ? **(Unit-I, Q.No. 24)**

10. (a) What is CSS? What are the advantages of CSS? **(Unit-II, Q.No. 5)**

OR

(b) Explain filters and transactions for multimedia effects. **(Unit-II, Q.No. 14)**

11. (a) Define expression. List the various types of expressions in JavaScript. **(Unit-III, Q.No. 13)**

OR

(b) What are the data types supported by javascript? **(Unit-III, Q.No. 9)**

12. (a) Explain about Mouse related events with examples. **(Unit-IV, Q.No. 14, 15, 16, 17)**

OR

(b) Explain in detail Keyboard related events with examples. **(Unit-IV, Q.No. 11, 12)**

13. (a) Explain about DOM in XML ? **(Unit-V, Q.No. 10)**

OR

(b) Explain in detail about XML style sheets? **(Unit-V, Q.No. 7)**