

RJS First Grade College

Koramangala, Bengaluru – 560034

Department of Computer Science and Applications

Subject - WEB PROGRAMMING

UNIT - I FUNDAMENTALS OF THE WEB

Unit – I

12 Hours

Fundamentals of Web: Internet, WWW, Web Browsers, and Web Servers, URLs, MIME, HTTP, Security, The Web Programmers Toolbox. XHTML: Origins and evolution of HTML and XHTML, Basic syntax, Standard XHTML document structure, Basic text markup, Images, Hypertext Links, Lists, Tables.

Two marks questions.

1) Define Internet.

An internet is a group of networks connected together. The internet was first known as the Arpanet. The Internet is system of linked computer networks that facilitate data communication services such as remote login, file transfer and electronic mail.

2) What is www?

www stands for world wide web. It is a hypertext based system providing a uniform user friendly interface for finding and accessing internet resources.

3) What is domain name?

A domain name defines a realm of administrative autonomy, authority, or control in the Internet. It is hostnames that identify internet protocol resources such as web sites.

4) What is web browser?

A web browser is a software application which enables a user to display and interact with text, images, videos, music, games and other information.

5) Name any 4 web browsers.

Internet Explorer, Mozilla Firefox, Google chrome, opera, safari.

6) What is web server?

Web Server is a software program that is working as a daemon serving web documents. It runs on computer and is responsible for serving web pages mostly HTML documents via the Http protocol to clients mostly web browsers.

7) Expand URL.

UNIFORM RESOURCE LOCATOR.

8) Expand MIME and HTTP.

MIME: Multipurpose Internet Mail Extensions. HTTP: The Hypertext Transfer protocol.

9) Define break tag.

Break tag is an empty tag and has no end tag. It is use to insert line breaks.

10) Define image tag.

The image tag is used an image in an XHTML document.

11) What is hypertext link?

Hypertext links are the words or an image that take the user from one web page to another or to another part of the same web page when they are clicked on .(anchor tag <a>).

12) What is table?

A table is a matrix of row and columns in which each intersection of a row and a column is known as cell. A table is specified as the content of the block tag <table>

Five Marks Questions

1. Define Internet, Write its applications.(5marks)

Internet is a system connecting computers around the world using TCP/IP, which stands for Transmission Control Protocol/Internet Protocol, a set of standards for transmitting and receiving digital data. The Internet consists primarily of the collection of billions of interconnected web pages that are transferred using HTTP (Hypertext Transfer Protocol), and is collectively known as the World Wide Web.

Applications:

Search engine: It can be used to search anything and everything. Most popular search engines are Google and yahoo searches.

Shopping: Internet has made shopping very easy and flexible. You can buy or sell online.

Communication: It helps people to communicate either with the use of social networking websites or through e-mails and chats.

Job search: Nowadays, many people search for their jobs online as it is quicker and there is a larger variety of job vacancies present.

Research: Research papers are present online which helps in the researcher doing a literature review. **Video**

Conferencing: It enables direct face-to-face communication across networks via web cameras, microphones, and other communication tools.

E-commerce (electronic **commerce** or EC) is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, primarily the Internet. Largest e-commerce companies in India are Flip kart, Snap deal, and Amazon India.

On-line payments

The rising boom of online payments in India has given way to many new entrants in the industry such as Paytm, Mobikwik, oxigen etc who are majorly wallet driven payment companies.

2. What is MIME?(7 marks)

MIME stands for (*Multipurpose Internet Mail Extensions*). It is widely used internet standard for coding binary files to send them as e-mail attachments over the internet. MIME allows an E-mail message to contain a non-ASCII file such as a video image or a sound and it provides a mechanism to transfer a non text characters to text characters.

The MIME specification includes the following elements:

i. Message header fields. Five message header fields are defined. These fields provide information about the body of the message.

The five header fields defined in MIME are as follows:

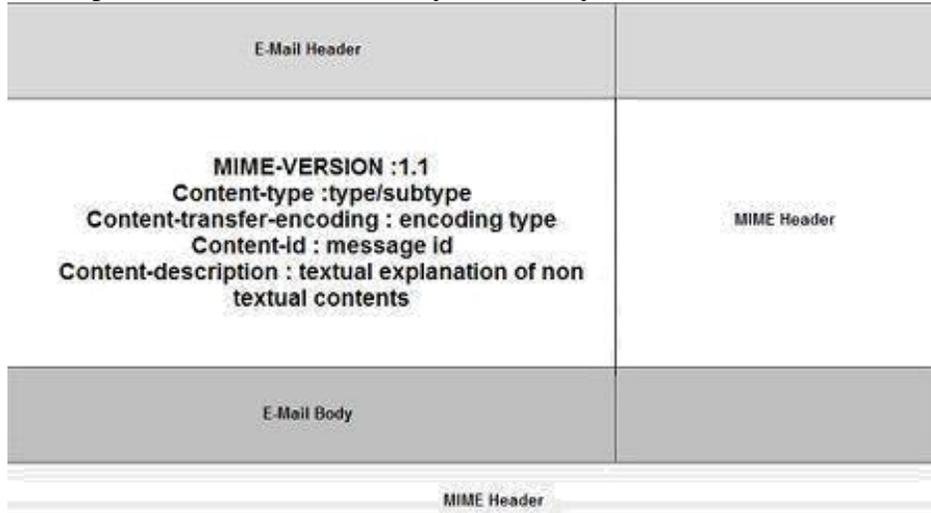
MIME-version. It indicates the MIME version being used. Example: It is represented as: MIME-version:1.1.

- a) **Content-type.** It describes the type and subtype of the data in the body of the message. The content type and content subtype are separated by slash.
- b) **Content-transfer encoding.** It describes how the object within the body has been encoded to US ASCII to make it acceptable for mail transfer. The content transfer encoding field is represented as:
Content-transfer-encoding : <type>
- c) **Content-Id.** It is used to uniquely identify the MIME entities in multiple contexts i.e. it uniquely identifies the whole message in a multiple message environment. This field is represented as: Content-id: id = <content-id>

d) **Content-description.** It is a plaintext description of the object within the body; It specifies whether the body is image, audio or video. This field is represented as: Content-description: <description>

ii. **Content formats.** A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.

iii. **Transfer encoding.** Transfer encoding are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.



3. What is a Web Browser? (5 marks)

A Web browser's basic function is to get files from a server and display them on the screen. It normally displays html files with images, PDF, videos etc in a structured layout. A browser is a collection of blocks or lines code that performs numerous tasks of showing a webpage on the pc screen.

How Web Browsers works?

World Wide Web is based on the client-server model. A user pc works as a client which receives and sends data to the server. If a web page is requested by a user, the browser contacts the server (where the website is stored) and by getting and interpreting the requested files, it shows the web page on the pc screen. Following are the most common web browser available today:

Browser	Vendor
Internet Explorer	Microsoft
Google Chrome	Google
Mozilla Firefox	Mozilla
Netscape Navigator	Netscape Communications Corp.
Opera	Opera Software
Safari	Apple

4. Write Short Notes on Web Servers(5marks)

A Web server is software that responds to client (I.e. web browser) requests. Every web site requires a web server to process client requests and 'serve up' the pages. Server machine is also referred to as the web server.

When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.

If the requested web page is not found, web servers will the send an **HTTP response: Error 404 Notfound.**

If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Following table describes the most leading web servers available today:

1. **Apache HTTP Server:** This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is open source software and can be installed on almost all operating systems.
2. **The Internet Information Server (IIS)** is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms.
3. **Sun Java System Web Server:** This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source.

5. Explain URL with an example (5marks)

Uniform Resource Locator (URL) refers to a web address which uniquely identifies a document over the internet.

For example, www.wikipedia.com/internet technology/index.html is an URL to the index.html which is stored on Wikipedia web server under internet technology directory.

URL Types

There are two forms of URL as listed below:

- a. **Absolute URL**
- b. **Relative URL**

ABSOLUTE URL

Absolute URL is a complete address of a resource on the web. This completed address comprises of protocol used, server name, path name and file name.

For example [http:// www.wikipedia.com / internet technology /index.htm](http://www.wikipedia.com/internet technology/index.htm). Where: **http** is the protocol.

wikipedia.com is the server name.

index.htm is the file name.

The protocol part tells the web browser how to handle the file. Similarly we have some other protocols also that can be used to create URL are:

- a) FTP
- b) https

- c) Gopher
- d) mailto
- e) news

RELATIVE URL

Relative URL is a partial address of a webpage. Unlike absolute URL, the protocol and server part are omitted from relative URL.

6. Explain Domain Name System Architecture(5marks)

The Domain name system comprises of Domain Names, Domain Name Space, and Name Server that have been described below:

Domain Names

Domain Name is a symbolic string associated with an IP address. There are several domain names available; some of them are generic such as com, edu, gov, net etc, while some country level domain names such as au, in, za, us etc.

Domain Name	Meaning	Domain Name	Meaning
Com	Commercial business	Au	Australia
Edu	Education	In	India
Gov	U.S. government agency	cl	Chile
Int	International entity	fr	France
Mil	U.S. military	us	United States
Net	Networking organization		
Org	Nonprofit organization		

Domain Name Space

The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels (from 0 to 127), with a root at the top. Each domain can be partitioned into sub domains and these can be further partitioned and so on

Name Server

Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses.

The three categories of Name Servers that manages the entire Domain Name System:

Root Server - Root Server is the top level server which consists of the entire DNS tree.

Primary Server - Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.

Secondary Server - The secondary server does not have authority to create or update a zone file.

□

7. What is a standard XHTML Document Structure?(7 marks)

A standard XHTML Document is as follows:

A DOCTYPE declaration at the start of the XHTML document should be written. Write all XHTML tags and attributes in lower case only.

Close all XHTML tags properly.

DOCTYPE Declaration

All XHTML documents must have a DOCTYPE declaration at the start. There are three types of DOCTYPE declarations, which are discussed in detail in XHTML Doctypes chapter.

Here is an example of using DOCTYPE –

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Case Sensitivity

XHTML is case sensitive markup language. All the XHTML tags and attributes need to be written in lower case only.

```
<!-- This is invalid in XHTML --> <A Href="/xhtml/xhtml_tutorial.html">XHTML Tutorial</A>
```

```
<!-- Correct XHTML way of writing this is as follows --> <a href="/xhtml/xhtml_tutorial.html">XHTML Tutorial</a>
```

Closing the Tags

Each and every XHTML tag should have an equivalent closing tag, even empty elements should also have closing tags. Here is an example showing valid and invalid ways of using tags –

```
<!-- This is invalid in XHTML --> <p>This paragraph is not written according to XHTML syntax. <!-- This is valid in XHTML --> <p>This paragraph is not written according to XHTML syntax.</p>
```

EXAMPLE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/TR/xhtml1" xml:lang="en" lang="en">
<head>
<title>Every document must have a title</title>
</head>
<body> ...your content goes here... </body>
</html>
```

8. What are the different types of Headings? Give an example. (10 marks)

HTML uses six levels of heading tags <h1> to <h6>; the higher the heading level number, the greater its importance, so <h1> defines the most important heading, whereas the <h6> defines the least important heading in the document.

EXAMPLE

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Example of HTML headings tag</title>
</head>
<body>
<h1>Heading level 1</h1>
<h2>Heading level 2</h2>
<h3>Heading level 3</h3>
<h4>Heading level 4</h4>
<h5>Heading level 5</h5>
<h6>Heading level 6</h6>
</body>
```

</html>

9. What is a HTML Table? Write about Cell spacing and Cell padding.(10 marks)

Tables in HTML are defined with the <table> tag. A table is divided into rows with the <tr> tag, which stands for table row, and each row is divided into data cells with the <td> tag, which stands for table data.

A <td> tag can contain text, links, images, lists, forms, other tables, etc.

Table Cell padding and Cell spacing

The cell padding and cell spacing attributes are used to adjust white space inside a table.

Cell padding adjust the white space between table cell border and its content. Cell spacing adjust the white space between table cells.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Example of HTML Table Cellpadding andCellspacing</title> </head>
```

```
<body>
```

```
<table border="1" cellpadding="10" cellspacing="5">
```

```
<thead>
```

```
<tr>
```

```
<th>No.</th>
```

```
<th>Name</th>
```

```
<th>Email</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
<td>1</td>
<td> Ram </td>
<td>ram@mail.com</td>
</tr>
<tr>
<td>2</td>
<td>Ali</td>
<td>ali@mail.com</td>
</tr>
</tbody>
</table>
</body>
</html>
```

10. What is Hyper Text Transfer Protocol (HTTP)? (8marks)

HTTP is a communication protocol. It defines mechanism for communication between browser and the web server. It is also called request and response protocol because the communication between browser and server takes place in request and response pairs.

HTTP Request

HTTP request comprises of lines which contains:

Request line Header Fields Message body

The first line i.e. the **Request line** specifies the request method i.e. **Get** or **Post**.

The second line specifies the header which indicates the domain name of the server from where index.htm is retrieved.

GET Method:

- a) Appends form-data into the URL in name/value pairs

- b) The length of a URL is limited (about 3000 characters)
- c) Never use GET to send sensitive data! (will be visible in the URL)
- d) Useful for form submissions where a user want to bookmark the result

POST Method:

- a) Appends form-data inside the body of the HTTP request (data is not shown is in URL)
- b) Has no size limitations
- c) Form submissions with POST cannot be bookmarked

Value	Description
get	Default. Appends the form-data to the URL in name/value pairs: URL?name=value&name=value
post	Sends the form-data as an HTTP post transaction

HTTP Response

Like HTTP request, HTTP response also has certain structure. HTTP response contains: Status line Headers Message body

In answer to a request, the server typically issues an HTTP response, the first line of which is often referred to as the status line. In that line the server echoes the HTTP version and gives a response status code (which is a three-digit integer) and a short message known as a reason phrase. Here's an example HTTP response:

HTTP/1.0 200 OK

Status Code	Explanation
200 - OK	The request succeeded.
204 - No Content	The document contains no data.
404 - Not Found	The requested resource does not exist on the server

11. What is a Protocol? Explain various types of Protocols(5 marks)

There are rules governing how data is transferred over networks, how they are compressed, how they are presented on the screen and so on. These set of rules are called protocols.

Various Types of Protocols are:

HTTP: Hypertext Transfer Protocol, used for transmitting and displaying information in the form of web pages on browsers.

FTP: File Transfer Protocol, used for file transfer (uploading and downloading) over the Internet.

POP: The most common protocol for receiving mail is Post Office Protocol (POP). Email clients such as Outlook Express require an address for a POP3 server before they can read mail. **SMTP:** Simple Mail Transfer Protocol used for email

Both SMTP and POP3 use TCP for managing the transmission and delivery of mail across the Internet.

Ethernet: Used for data transmission over a LAN.

IP: Internet Protocol is the primary network protocol used on the Internet, developed in the 1970s. IP is often used together with the Transport Control Protocol (TCP)

12. What is a List? Explain various types of Lists in HTML(10 marks)

Lists are the preferred way to display items one after the other, Lists have a tag to start and end the list itself, as well as a tag for each item in the list.

There are three types of lists, ordered, unordered and definition lists.

Unordered Lists

An unordered list is a bulleted list, similar to the menu on the right. An Unordered List is represented by - ``
... ``

A number of list items (li elements) will go within the ul tags. Add the text for each item in between some ``
and `` tags.

Bullet Type `<ul type="disc | circle | square">` by default a browser will show a round bullet. This can be changed by using the type attribute of the ul tag, which will change the bullet type for the entire list.

Item Type `<li type="?">` You can set the type of bullet for an item in the middle of the list by setting the type attribute of a li tag.

Example for Unordered List

```
<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Example of HTML Unordered List</title>

</head>

<body>
```

```
<h1>HTML Unordered List</h1>
```

```
<ul>
```

```
<li>Chocolate Cake</li>
```

```
<li>Black Forest Cake</li>
```

```
<li>Pineapple Cake</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

Ordered Lists

An ordered list is used to create an indexed list, such as a numbers or alphabetical list. A number of list items will then go between the ordered list tags.

Ordered List Item - `` an item `` each item must use the `` tags the same as with an unordered list. But this time the browser will number each item automatically, instead of showing bullets.

List Type `<ol type="A | a | I | i | 1">`Set the type of list index by using the `type=""` attribute.

The default style is numeric, and you can also choose from upper or lowercase, alphabetic or roman numerals.

List Starting Position `<ol start="">`Set the starting number (or letter) if you don't want the list to start at 1 or A.

Item Value `<li value="">`You can set the value of an item in the middle of the list manually, if you do not want it to follow the previous letter or number.

Example for Ordered List

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Example of HTML Ordered List</title>
```

```
</head>
```

```
<body>
```

```
<h1>HTML Ordered List</h1>
```

```
<ol>
```

```
<li>Mix ingredients</li>
```

```
<li>Bake in oven for an hour</li>
```

```
<li>Allow to stand for ten minutes</li>
```

```
</ol>
```

```
</body>
```

```
</html>
```

Definition Lists

This type of list is used to define and describe terms, much like a dictionary. A browser will usually bold the term, and indent the definition.

Define a Definition List - `<dl>` `</dl>` Set the start and end of a definition list.

All entries go within the dl tags. Each entry will usually consist of one dt and one dd element.

Definition Title - `<dt>` `</dt>` The title of a term being defined.

Definition Description - `<dd>` `</dd>` the definition of a term. Note: you can have multiple definitions for a single term.

Example for Definition List

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Example of HTML Definition List</title>
```

```
</head>
```

```
<body>
```

```
<h1>HTML Definition List</h1>
```

<dl>

<dt>Bread</dt>

<dd>A baked food made of flour.</dd>

<dt>Coffee</dt>

<dd>A drink made from roasted coffee beans.</dd>

</dl>

</body>

</html>

RISEFGC

UNIT – II

Unit – II

12 Hours

HTML and XHTML: Forms, Frames in HTML and XHTML, Syntactic differences between HTML and XHTML. CSS: Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The Box model, Background images, The and

HTML AND XHTML

TWO MARKS

1) What is CSS?

CSS outline the style of an HTML webpage, it is a language by which we can set the behavior of an HTML webpage. It describes how the HTML content will be shown on screen.

CSS controls the layout of several HTML web pages. CSS is referred to as the Cascading Style Sheet.

2) Name all the modules which are used in the current version of CSS. Answer: There are several modules in CSS as stated below:

- Selectors
- Box Model
- Backgrounds and Borders
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface.

3) Explain different types of CSS.

There are three types of CSS as mentioned below.

- **External** – These are written in separate files.
- **Internal** – These are cited at the top of the web page code document.
- **Inline** – These are written right next to the text.

4) Why is the external style sheet useful?

Answer: External style sheet is very useful as we write all the styling codes in a single file and it can be used anywhere by just referencing the link of that external style sheet file.

So if we do any changes in that external file, then the changes can also be observed on the webpage. So we can say that it is very useful and it makes your work easy while working on larger files.

5) What are the uses of an embedded style sheet?

Answer: Embedded style sheet gives us the privilege to define styles in one place in an HTML document.

We can generate multiple classes using an embedded style sheet to use on multiple tag types of a web page and also there is no extra downloading required for importing the information

6) How to use CSS selector?

Answer: By using the CSS selector, we can choose the content which we want to style so that we can say that it is a bridge between the style sheet and the HTML files.

The syntax for CSS selector is “select” HTML elements created on their id, class, type, etc.

7) Explain the concept of Tweening.

Answer: Tweening is the process in which we create intermediate frames between two images to get the appearance of the first image which develops into the second image.

It is mainly used for creating animation.

7) Explain the term Responsive web design.

Answer: It is a method in which we design and develop a web page according to the user activities and conditions which are based on various components like the size of the screen, portability of the web page on the different devices, etc.

Hence it is done by using different flexible layouts and grids.

8) What is CSS specificity?

Answer: CSS specificity is a score or rank that decides which style declaration has to be used to an element. (*)

this

Universal selector has low specificity while ID selectors have high specificity.

There are four categories in CSS which authorized the specificity level of the selector.

- Inline style
- IDs
- Classes, Attributes, and pseudo-classes.
- Elements and pseudo-elements.

9) How will you add border images to an HTML element?

Answer: We can set the image to be used as the border-image alongside an element by using the property of CSS “border-image”.

Example:

```
#bordering {  
border: 15px solid transparent; padding: 20px;  
border-image: url(border.png) 30 round;  
}
```

10) What is the use of the Box Model in CSS?

Answer: In CSS, the box model is a box that binds all the HTML elements and it includes features like margins, border, padding, and the actual content.

By using a box model we will get the authority to add the borders all around the elements and we can also define the space between the elements.

11) What is a CSS pseudo-class?

Answer: It is a class that is used to define a special state of an HTML element.

This class can be used by styling an element when a user snooped over it and also it can style an HTML element when it gets the focus.

```
selector:pseudo-class {property:value;
}
```

12) Explain the concept of pseudo-elements in CSS.

Answer: It is a feature of CSS which is used to style the given parts of an element. For Example, we can style the first letter or line of an HTML element. selector::pseudo-element {
property:value;
}

13) Write all the position states used in CSS.

Answer: In CSS, there are four position states as stated below:

- Static(default)
- Relative
- Fixed
- Absolute

14) What are navigation bars in CSS?

Answer: By using navigation bars we can make an ordinary HTML page into a user-specific and more dynamic webpage.

Basically, it is a list of links, hence use of and elements makes the perfect sense. ul {

```
list-style-type: none;margin: 0;
```

```
padding: 0;
}
```

15) What are the differences between relative and absolute in CSS?

Answer: The main difference between relative and absolute is that “relative” is used for the same tag in CSS and it means that if we write the left:10px then the padding will shift to 10px in the left while absolute is totally relative to the non-static parent.

It means if we write left:10px then the result will be 10px far from the left edge of the parent element.

16) How Is Xhtml Better Than Html? Answer :

The following are the reasons for why XHTML is better than HTML:

- XHTML uses style sheets instead of font, color and alignment tags of HTML.
- XHTML allows to style sheets and scripts embedding in CDATA section.
- XML of XHTML eases the integration of new elements as subsets of SGML.
- Portable – Non-pc platforms can be accessed with XHTML.

17) What is simple selector?

Answer: A simple selector can be either be a an element type selector or universal selector followed by attributeselector, ID selectors or pseudo-classes . the simple selector matches if all of its components match.

18) What is class selector?

Answer: A class selector is a name preceded by a full stop(.). It matches components that meet a class condition. The CSS syntax to declare a class selector is to prefix the condition with a dot.

19) What is universal selector?

Answer: The universal selector is denoted with an asterisk(*) as the local name. It represent any single element inthe document tree in any namespace if no default namespace has been specified for selectors.

20) What is Font property?

Answer:The font properties are the properties that include font-style, font-variant, font-weight, font-size, font shorthand's,and font-family.

21) What is List property?

Answer:A list property includes list-style-type , list-style-postion,list-style-image and list-style

22) What is form?

Answer:The **form** tag is used to delimit the start and stop of a form element and serves as a container for form controls(fields).

23) What is frame tag?

Answer:The <frame> tag defines one particular window (frame) within a <frameset>.

Each <frame> in a <frameset> can have different attributes, such as border, scrolling, the ability to resize, etc.

5 marks,7marks, 8 marks, 10marks

1) What are the differences between Html and xhtml?

Html	xhtml
HTML or HyperText Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser.	XHTML (Extensible HyperText Markup Language) is a family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML), the language in which web pages are written.

Filename extension .html, .htm	.html, .htm xhtml, .xhtml, .xml, .html, .htm
Internet media type text/html	application/xhtml+xml
Developed by W3C & WHATWG	World Wide Web Consortium
Extended from sgml	XML, SGML, HTML
Flexible framework requiring lenient HTML specific parser.	Restrictive subset of XML and needs to be parsed with standard XML parsers.

Some key Difference between HTML and XHTML are explained below

1. In HTML, we can have the empty or open tags means it is not required to end the tag e.g. `<p>`. In XHTML, the tags should be closed or self closed, if opened `<p> </p>` or `
`
2. In HTML, all the written text that should be displayed in browser can put under HTML bodyelement. In XHTML, the content can be put in blocks.
3. In HTML, While defining the attributes it is not necessary to mention quotes. For e.g. `<option selected>`. In XHTML, while defining the attributes it is mandatory to mentionquotes. For e.g. `<option selected="Selected">`.
4. In HTML, the values of attributes are not so important. For e.g. `<input type="radiobutton" selected>`. In XHTML, the values of attributes are important. For e.g. `<input type="radiobutton" selected="selected">`.
5. In HTML, there are no strict rules on writng the structure of elements for e.g. `<p> Hello world</p>`. In XHTML, there are strict rules on writing structureof elements For e.g. `<p>Hello world</p>`
6. In HTML, the tags and attributes can be described in lower case or upper case. In XHTML, the tags and attributes can be described in lower case only.
7. In HTML, one root element is not mandatory. In XHTML, the documents should have oneroot element.
8. In HTML, XML declaration is not necessary. In XHTML, it is based on the set of rules ofXML.

Unit: III JAVA SCRIPT

Unit:II

Java Script: Overview of JavaScript; Object orientation and JavaScript; General syntactic characteristics; Primitives, Operations, and expressions; Screen output and keyboard input; Control statements; Object creation and Modification; Arrays; Functions; Constructor; Pattern matching using expressions; Errors in scripts; Examples. tags, Conflict resolution.

2 Marks questions

1. Who invented java scripts?

JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**.and It is invented by BrendanEich. JavaScript was first known as **LiveScript**

2. What is JavaScript?

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.

3. What is the relation between Java Script and ECMAScript?

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- 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.

4. Which tag is used to add javascript?

JavaScript can be implemented using JavaScript statements that are placed within the

`<script>... </script>` HTML tags in a web page.

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>
```

5. What are the attributes of script tag?

The attributes of script tag are two:

1.type attribte 2.src attribute

1. **Type:** This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".
2. **src attribute :** This attribute is used to link the external javascript to Html page.

6.How to declare a variable in java script?

Variables are declared with the **var** keyword as follows.

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

5marks questions

1. Advantages and Disadvantages of JavaScript.

Advantages of JavaScript

The merits of using JavaScript are:

- Less server interaction:** You can validate user input before sending the page off to the server. 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.

Disadvantages of JavaScript:

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features:

- 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.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

2. What are the different ways of placing the JavaScript code in XHTML?

There 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:

❑ **Between the XHTML document's head tags:**

Script in <head>...</head> section.

❑ **Between the XHTML document's body tags:** Script in <body>...</body> section.

❑ **In Both Head and Body tag**

Script in <body>...</body> and <head>...</head> sections.

❑ **In an External file:**

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>
Click here for the result
<input type="button" onclick="sayHello()" value="Say Hello" />
```

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>
```

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>
```

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>
<script type="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. What are the JavaScript Data types?

JavaScript Datatypes are classified into two types

1. Primitive data types 2. Composite data types

1. Primitives data types:

One of the most fundamental characteristics of a programming language is the set of data types it supports. These are the type of values that can be represented and manipulated in a programming language.

JavaScript allows you to work with three primitive data types:

Numbers, e.g., 123, 120.50 etc.

Strings of text, e.g. "This text string" etc.

Boolean, e.g. true or false.

Undefined

Null

Compositive data types:

Arrays ,functions and objects are example of compositive datatype.

4. What is an Operator?

Let us take a simple expression **4 + 5 is equal to 9**. Here 4 and 5 are called **operands** and '+' is called the **operator**. JavaScript supports the following types of operators.

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

Let's have a look at all the operators one by one.

Arithmetic Operators

JavaScript supports the following arithmetic operators: Assume variable A holds 10 and variable B holds 20, then:

S. No.	Operator and Description
1	+ (Addition) Adds two operands Ex: A + B will give 30
2	- (Subtraction) Subtracts the second operand from the first Ex: A - B will give -10
3	* (Multiplication) Multiply both operands Ex: A * B will give 200
4	/ (Division) Divide the numerator by the denominator Ex: B / A will give 2
5	% (Modulus) Outputs the remainder of an integer division Ex: B % A will give 0
6	++ (Increment) Increases an integer value by one Ex: A++ will give 11
7	-- (Decrement) Decreases an integer value by one Ex: A-- will give 9

Example

The following code shows how to use arithmetic operators in JavaScript.

```
<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>
<p>Set the variables to different values and then try...</p>
</body>
</html>
```

Comparison Operators

JavaScript supports the following comparison operators: Assume variable A holds 10 and variable B holds 20, then:

S.No	Operator and Description
1	== (Equal) Checks if the value of two operands are equal or not, if yes, then the condition becomes true. Ex: (A == B) is not true.
2	!= (Not Equal) Checks if the value of two operands are equal or not, if the values are not equal, then the condition becomes true. Ex: (A != B) is true.
3	> (Greater than) Checks if the value of the left operand is greater than the value of the right operand, if yes, then the condition becomes true. Ex: (A > B) is not true.
4	< (Less than) Checks if the value of the left operand is less than the value of the right operand, if yes, then the condition becomes true. Ex: (A < B) is true.
5	>= (Greater than or Equal to) Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true. Ex: (A >= B) is not true.
6	<= (Less than or Equal to) Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomes true.

Ex: (A <= B) is true.

Example

The following code shows how to use comparison operators in JavaScript.

```
<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>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
```

UNIT IV

Unit:IV

Java Script and HTML Documents: The JavaScript execution environment; The Document Object Model; Element access in JavaScript; Events and event handling; Handling events from the Body elements, Button elements, Text box and Password elements; The DOM 2 event model; The navigator object; DOM tree traversal and modification.

1. Define CSS?

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, as well as a variety of other effects.

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

2. Advantages of CSS?

- **CSS saves time** - You can write CSS once and then reuse the 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 the 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 cellphones or for printing.
- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

3. Who creates and Maintains the CSS?

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called **specifications**. When a specification has been discussed and officially ratified by the W3C members, it becomes a recommendation.

These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

NOTE: The World Wide Web Consortium or W3C is a group that makes recommendations about how the Internet works and how it should evolve

4. What are the different versions of CSS?

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

5. What is the Syntax of the CSS?

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 the value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows:

```
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 the given value *1px solid #C00* is the value of that property. You can define selectors in various simple ways based on your comfort.

5Marks Questions

6. What are the different types of CSS selectors.

This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings:

```
h1 {  
    color: #36CFFF;  
}
```

The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type:

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

This rule renders the content of every element in our document in black.

The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, the style rule will apply to `` element only when it lies inside the `` tag.

```
ul em {  
  color: #000000;  
}
```

The 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 a given element. Consider the following example:

```
<p class="center bold">  
This para will be styled by the classes center and bold.
```

```
</p>
```

The 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 within tags having *id* attribute set to *black*.

The Child Selectors

```
body > p {  
    color: #000000;  
}
```

You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example:

This rule will render all the paragraphs in black if they are a direct child of the `<body>` element. Other paragraphs put inside other elements like `<div>` or `<td>` would not have any effect of this rule.

The Attribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the

```
input[type="text"]{
  color: #000000;
}
```

input elements having a type attribute with a value of *text*:

The advantage to this method is that the `<input type="submit" />` element is unaffected, and the color applied only to the desired text fields.

There are following rules applied to attribute selector.

- **p[lang]** - Selects all paragraph elements with a *lang* attribute.
- **p[lang="fr"]** - Selects all paragraph elements whose *lang* attribute has a value of exactly "fr".
- **p[lang~="fr"]** - Selects all paragraph elements whose *lang* attribute contains the word "fr".
- **p[lang|="en"]** - Selects all paragraph elements whose *lang* attribute contains values that are exactly "en", or begin with "en-".

Multiple Style Rules

```
h1 {
  color: #36C;
  font-weight: normal;
  letter-spacing: .4em;
  margin-bottom: 1em;
  text-transform: lowercase;
}
```

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example: Here all the property and value pairs are separated by a **semicolon (;)**. You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

For a while, don't bother about the properties mentioned in the above block. These properties will be explained in the coming chapters and you can find the complete detail about properties in CSS References.

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example:

```
h1, h2, h3 {
color: #36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
```

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various *class* selectors together as shown below:

```
#content, #footer, #supplement {
position: absolute;
left: 510px;
width: 200px;
}
```

7. What are the ways that we can embed the CSS in HTML?

There are four ways to associate styles with your HTML document. Most commonly used methods are inline CSS and External CSS.

Embedded CSS-The <style>Element

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside the <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document. Here is the generic syntax:

```
<head>
<style type="text/css" media="...">
Style Rules
.....
</style>
</head>
```

Attributes

Attributes associated with <style> elements are:

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type(MIME type). This is a required attribute.
media	screen tty tv projection handheld print braille	Specifies the device, the document will be displayed on. Default value is <i>all</i> . This is an optional attribute.

Example

Following is an example of embed CSS based on the above syntax:

```
<head>
<style type="text/css" media="all">
h1{
color: #36C;
}
</style>
</head>
```

Inline CSS - The *style* Attribute

You can use *style* attribute of any HTML element to define style rules. These rules will be applied to that element only. Here is the generic syntax:

```
<element style="...style rules ...">
```

Attributes

Attribute	Value	Description
style	stylerules	The value of <i>style</i> attribute is a combination of style declarations separated by semicolon (;).

Example

Following is the example of inline CSS based on the above

syntax:

```
<h1 style ="color:#36C;"> This is inline CSS </h1>
```

It will produce the following result:

This is inline CSS

External CSS - The <link> Element

The <link> element can be used to include an external stylesheet file in your HTML document.

An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

Here is the generic syntax of including external CSS file:

```
<head>
<link type="text/css" href="..." media="..." />
</head>
```

Attributes

Attributes associated with <style> elements are:

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type (MIME type). This attribute is required.
href	URL	Specifies the style sheet file having Style rules. This attribute is required.
media	screen tv projection handheld print braille aural all	Specifies the device the document will be displayed on. Default value is <i>all</i> . This is an optional attribute.

Example

Consider a simple style sheet file with a name *mystyle.css* having the following rules:

```
h1, h2, h3 {
color: #36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
```

Now you can include this file *mystyle.css* in any HTML document as follows:

```
<head>
<link type="text/css" href="mystyle.css" media="all" />
</head>
```

Imported CSS-@import Rule

@import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.

```
<head>
<@import "URL";
</head>
```

```
<@import url("URL");
</head>
```

Example

Here URL is the URL of the style sheet file having style rules. You can use another syntax as well:<head>

Following is the example showing you how to import a style sheet file into an HTML document:

```
<head>
@import "mystyle.css";
</head>
```

5. What are CSS Rules?

We have discussed four ways to include style sheet rules in an HTML document. Here is the rule to override any Style Sheet Rule.

- Any inline style sheet takes the highest priority. So, it will override any rule defined in <style>...</style> tags or the rules defined in any external style sheet file.
- Any rule defined in <style>...</style> tags will override the rules defined in any external style sheet file.
- Any rule defined in the external style sheet file takes the lowest priority, and the rules defined in this file will be applied only when the above two rules are not applicable.

Handling Old Browsers

There are still many old browsers who do not support CSS. So, we should take care while writing our Embedded CSS in an HTML document. The following snippet shows how to use comment tags to hide CSS

from older browsers:

```
<style type="text/css">
<!--
body, td {
    color: blue;
}
-->
```

CSS Comments

Many times, you may need to put additional comments in your style sheet blocks. So, it is very easy to comment any part in the style sheet. You can simply put your comments inside `/*.....this is a comment in style sheet*/`.

You can use `/* */` to comment multi-line blocks in similar way you do in C and C++ programming languages.

Example

```
/* This is an external style sheet file */
h1, h2, h3 {
color: #36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
/* end of style rules. */
```

6. What are the Units of Measures in CSS?

Before we start the actual exercise, we would like to give a brief idea about the CSS Measurement Units. CSS supports a number of measurements including absolute units such as inches, centimeters, points, and so on, as well as relative measures such as percentages and em units. You need these values while specifying various measurements in your Style rules e.g. **`border="1px solid red"`**.

We have listed out all the CSS Measurement Units along with proper Examples:

Unit	Description	Example
%	Defines a measurement as a percentage relative to another value, typically an enclosing element.	p {font-size: 16pt; line-height: 125%;}
cm	Defines a measurement in centimeters.	div {margin-bottom: 2cm;}
em	A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt.	p {letter-spacing: 7em;}
ex	This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x.	p {font-size: 24pt; line-height: 3ex;}
in	Defines a measurement in inches.	p {word-spacing: .15in;}
mm	Defines a measurement in millimeters.	p {word-spacing: 15mm;}
pc	Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch.	p {font-size: 20pc;}
pt	Defines a measurement in points. A point is defined as 1/72nd of an inch.	body {font-size: 18pt;}
px	Defines a measurement in screen pixels.	p {padding: 25px;}

Q6. What the different formats of in CSS?

CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or for the background of the element. They can also be used to affect the color of borders and other decorative effects.

You can specify your color values in various formats. Following table lists all the possible formats:

Format	Syntax	Example
Hex Code	#RRGGBB	p{color:#FF0000;}
Short Hex Code	#RGB	p{color:#6A7;}
RGB %	rgb(rrr%,ggg%,bbb%)	p{color:rgb(50%,50%,50%);}
RGB Absolute	rgb(rrr,ggg,bbb)	p{color:rgb(0,0,255);}
keyword	aqua, black, etc.	p{color:teal;}

These formats are explained in more detail in the following sections:

CSS Colors-Hex Codes

A hexadecimal is a 6 digit representation of a color. The first two digits (RR) represent a red value, the next two are a green value (GG), and the last are the blue value (BB).

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Jasc Paintshop Pro, or even using Advanced Paint Brush.

Each hexadecimal code will be preceded by a pound or hash sign '#'. Following are the examples to use Hexadecimal notation.

COLOR	COLOR in HEX
	#000000
	#FF0000
	#00FF00
	#0000FF
	#FFFF00
	#00FFFF
	#FF00FF
	#C0C0C0
	#FFFFFF

CSS Colors-Short Hex Codes

This is a shorter form of the six-digit notation. In this format, each digit is replicated to arrive at an equivalent six-digit value. For example: #6A7 becomes #66AA77.

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, Jasc Paintshop Pro or even using Advanced Paint Brush.

Each hexadecimal code will be preceded by a pound or hash sign #. Following are the examples to use the Hexadecimal notation.

UNIT: V

Unit: V

Dynamic Documents with JavaScript: Introduction to dynamic documents; Positioning elements; Moving elements; Element visibility; Changing colors and fonts; Dynamic content; Stacking elements; Locating the mouse cursor; Reacting to a mouse click; Slow movement of elements; Dragging and dropping elements. XML: Introduction; Syntax; Document structure; Document Type definitions;

2marks Questions

1. Define XML and its purpose?

XML stands for **Extensible Markup Language**. It 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.

2. Write the characteristics of XML in brief?

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.

3. What are the uses of XML?

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.

4. What is Markup Language?

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!

5. Why XML is called as Programming Language?

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.

In this chapter, we will discuss the simple syntax rules to write an XML document. Following is a complete XML document:

```
<?xml version="1.0"?>
<contact-info>
<name>Tanmay Patil</name>
<company>TutorialsPoint</company>
<phone>(011) 123-4567</phone>
</contact-info>
```

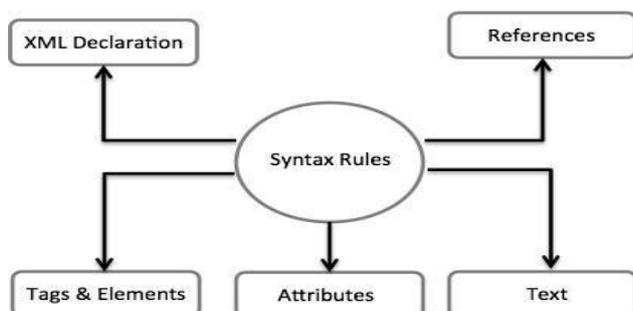
You can notice, there are two kinds of information in the above example:

- Markup, like <contact-info>
- The text, or the character data

5 Marks Questions

6. Write the syntax rules of XML With Diagram.

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 bellow diagram in detail.

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 the document contains XML declaration, then it strictly needs to be the first statement of the XML document.
- The XML declaration strictly needs to be the first statement in the XML document.
- An HTTP protocol can override the value of *encoding* that you put in the XML declaration.

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> ... </element>
```

Element Syntax: Each XML-element needs to be closed either with start or with end elements as shown below:

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>**.

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:

```
<a b="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:

```
<a b=x> ...</a>
```

In the above syntax, the attribute value is not defined in quotation marks.

XMLReferences

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".

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

7. Define XML Documents? Explain With Example?

An **XML document** is a basic unit of XML information composed of elements and other markup in an orderly package. An XML document can contain a wide variety of data. For example, database of numbers, numbers representing molecular structure or a mathematical equation.

XML Document Example

A simple document is shown in the following example:

```

<?xml version="1.0"?>
<contact-info>
  <name>Tanmay Patil</name>
  <company>TutorialsPoint</company>
  <phone>(011) 123-4567</phone>
</contact-info>

```

The following image depicts the parts of XML document.



DocumentPrologSection

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](#).

DocumentElementsSection

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.

8. Write about XML declaration?

XML declaration in detail. **XML declaration** contains details that prepare an XML processor to parse the XML document. It is optional, but when used, it must appear in the first line of the XML document.

Syntax

Following syntax shows XML declaration:

```

<?xml
  version="version_number"
  encoding="encoding_declaration"
  standalone="standalone_status"
?>

```

Each parameter consists of a parameter name, an equals sign (=), and parameter value inside a quote. Following table shows the above syntax in detail:

Parameter	Parameter value	Parameter description
Version	1.0	Specifies the version of the XML standard used.
Encoding	UTF-8, UTF-16, ISO-10646-UCS-2, ISO-10646-UCS-4, ISO-8859-1 to ISO-8859-9, ISO-2022-JP, Shift_JIS, EUC-JP	It defines the character encoding used in the document. UTF-8 is the default encoding used.
Standalone	yes or no.	It informs the parser whether the document relies on the information from an external source, such as external document type definition (DTD), for its content. The default value is set to <i>no</i> . Setting it to <i>yes</i> tells the processor there are no external declarations required for parsing the document.

Rules

An XML declaration should abide with the following rules:

- If the XML declaration is present in the XML, it must be placed as the first line in the XML document.
- If the XML declaration is included, it must contain version number attribute.
- The parameter names and values are case-sensitive.
- The names are always in lower case.
- The order of placing the parameters is important. The correct order is: *version, encoding and standalone*.
- Either single or double quotes may be used.
- The XML declaration has no closing tag, i.e. `<?xml>`

XML Declaration Examples

Following are few examples of XML declarations:

XML declaration with no parameters:

```
<?xml >
```

XML declaration with version definition:

```
<?xml version="1.0">
```

XML declaration with all parameters defined:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

XML declaration with all parameters defined in single quotes:

```
<?xml version='1.0' encoding='iso-8859-1' standalone='no' ?>
```

9. Write about XML tags?

Let us learn about one of the most important part of XML, the XML *tags*. **XML tags** form the foundation of XML. They define the scope of an element in XML. They can also be used to insert comments, declare settings required for parsing the environment, and to insert special instructions.

We can broadly categorize XML tags as follows:

StartTag

The beginning of every non-empty XML element is marked by a start-tag. Following is an example of start-tag:

```
<address>
```

End Tag

Every element that has a start tag should end with an end-tag. Following is an example of end-tag:

```
</address>
```

Note, that the end tags include a solidus ("/") before the name of an element.

EmptyTag

The text that appears between start-tag and end-tag is called content. An element which has no content is termed as empty. An empty element can be represented in two ways as follows:

A start-tag immediately followed by an end-tag as shown below:

```
<hr></hr>
```

A complete empty-element tag is as shown below:

```
<hr />
```

Empty-element tags may be used for any element which has no content.