

Question bank of web Technology:

1. Explain OSI model.

The OSI Model (Open Systems Interconnection Model) is a conceptual framework used to describe the functions of a networking system. The OSI model characterizes computing functions into a universal set of rules and requirements in order to support interoperability between different products and software. In the OSI reference model, the communications between a computing system are split into seven different abstraction layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application.

Created at a time when network computing was in its infancy, the OSI was published in 1984 by the International Organization for Standardization (ISO). Though it does not always map directly to specific systems, the OSI Model is still used today as a means to describe Network Architecture.

The 7 Layers of the OSI Model

- Physical Layer
- Data Link Layer
- Network Layer
- Transport Layer
- Session Layer
- Presentation Layer
- Application Layer

2. Explain TCP/IP model.

TCP/IP stands for Transmission Control Protocol/Internet Protocol and is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP is also used as a communications protocol in a private computer network (an intranet or extranet).

The entire IP suite -- a set of rules and procedures -- is commonly referred to as TCP/IP. TCP and IP are the two main protocols, though others are included in the suite. The TCP/IP protocol suite functions as an abstraction layer between internet applications and the routing and switching fabric.

3. What is the difference between TCP/IP and OSI model.

The difference between TCP/IP and OSI Model seems to be minor but by composition, features, functions and purpose, the two are extremely different.

The TCP/IP or the Transmission Control Protocol/ Internet Protocol is a communication protocols suite using which network devices can be connected to the Internet. On the other hand, the Open Systems Interconnection or OSI Model is a conceptual framework, using which the functioning of a network can be described.

In this article, we bring to you a tabulated difference between the TCP/IP model and the OSI model. This will help candidates prepare themselves and upgrade their Computer Knowledge for the upcoming competitive exams.

4. What do you mean by front end and backend web development.

Frontend and backend are two critical aspects of any application. The frontend is what your users see and includes visual elements like buttons, checkboxes, graphics, and text messages. It allows your users to interact with your application. The backend is the data and infrastructure that make your application work. It stores and processes application data for your users.

5. Difference between static and dynamic websites.

Static Web pages: Static Web pages are very simple. It is written in languages such as HTML, JavaScript, CSS, etc. For static web pages when a server receives a request for a web page, then the server sends the response to the client without doing any additional process. And these web pages are seen through a web browser. In static web pages, Pages will remain the same until someone changes it manually.

Dynamic Web Pages: Dynamic Web Pages are written in languages such as CGI, AJAX, ASP, ASP.NET, etc. In dynamic web pages, the Content of pages is different for different visitors. It takes more time to load than the static web page. Dynamic web pages are used where the information is changed frequently, for example, stock prices, weather information, etc.

6. What do you mean by protocols and standards?

A protocol defines a set of rules used by two or more parties to interact between themselves. A standard is a formalized protocol accepted by most of the parties that implement it. Not all protocols are standards (some are proprietary). Not all standards are protocols (some govern other layers than communication).

7. Explain following protocols in details:

7.1. UDP: In computer networking, the UDP stands for User Datagram Protocol. The David P. Reed developed the UDP protocol in 1980. It is defined in RFC 768, and it is a part of the TCP/IP protocol, so it is a standard protocol over the internet. The UDP protocol allows the computer applications to send the messages in the form of datagrams from one machine to another machine over the Internet Protocol (IP) network. The UDP is an alternative communication protocol to the TCP protocol (transmission control protocol).

The following are the features of the UDP protocol:

- Transport layer protocol
- Connectionless
- Ports
- Faster transmission
- Acknowledgment mechanism
- Segments are handled independently.
- Stateless
-

7.2. TCP: Transmission Control Protocol (TCP) is a communications standard that enables application programs and computing devices to exchange messages over a network. It is designed to send packets across the internet and ensure the successful delivery of data and messages over networks.

TCP is one of the basic standards that define the rules of the internet and is included within the standards defined by the Internet Engineering Task Force (IETF). It is one of the most commonly used protocols within digital network communications and ensures end-to-end data delivery.

7.3. IP: IP address stands for “Internet Protocol address.” The Internet Protocol is a set of rules for communication over the internet, such as sending mail, streaming video, or connecting to a website. An IP address identifies a network or device on the internet.

The internet protocols manage the process of assigning each unique device its own IP address. (Internet protocols do other things as well, such as routing internet traffic.) This way, it’s easy to see which devices on the internet are sending, requesting, and receiving what information.

IP addresses are like telephone numbers, and they serve the same purpose. When you contact someone, your phone number identifies who you are, and it assures the person who answers the phone that you are who you say you are. IP addresses do the exact same thing when you’re online — that’s why every single device that is connected to the internet has an IP address.

7.4. FTP: File transfer protocol (FTP) is an Internet tool provided by TCP/IP. The first feature of FTP is developed by Abhay Bhushan in 1971. It helps to transfer files from one computer to another by providing access to directories or folders on remote computers and allows software, data, text file to be transferred between different kinds of computers. The end-user in the connection is known as localhost and the server which provides data is known as the remote host.

- The goals of FTP are:
- It encourages the direct use of remote computers.
- It shields users from system variations (operating system, directory structures, file structures, etc.)
- It promotes sharing of files and other types of data.

7.5. HTTP: HTTP (Hypertext Transfer protocol) is a fundamental protocol that defines how data is formatted and transmitted on the web. It facilitates the communication between web servers and browsers, allowing the transfer of text, image, videos and other resources. However, HTTP is not secure, making it susceptible to data interception.

7.6. HTTPS: HTTPS (Hypertext Transfer Protocol Secure) is an extension of HTTP that adds a layer of encryption through SSL (Secure Sockets Layer) or TLS (Transport Layer Security). HTTPS ensures secure data transmission by encrypting the information exchange between the user's browser and the web server enhancing confidentiality and protecting against potential cyber threats. It's essential for safeguarding sensitive details like passwords, credit card details and personal information during online transaction and browsing.

8. Write Short note on Following:

8.1. DNS: DNS (Domain Name System) is a crucial internal protocol that translates human readable domain name (ex.: example.com) into IP address (ex.: 192.168.1.1) used by computer to communicate. It acts like a phonebook facilitating efficient browsing by simplifying the way we access websites and services on the internet.

8.2. URL: URL (Uniform Resource Locator) is a web address that specifies the location of a resource on the internet. It comprises several components, including the protocol (ex. https), domain name (ex. example.com), and a path, allowing users to access web pages, files or services. URLs play a fundamental role in internet navigation and enable seamless retrieval of information from the World Wide Web.

8.3. WWW: The World Wide Web (WWW), commonly known as the Web, is an information system where documents and other web resources are identified by Uniform Resource Locators, which may be interlinked by hypertext, and are accessible over the Internet. World Wide Web was invented in 1989 by Tim Berners Lee and has since become a fundamental component of the internet. The web allows users to access and share information, multimedia content and services through web browsers.

8.4. Internet: The Internet is a worldwide network that links computers. People may exchange information and converse through the Internet from any location with an Internet connection. The Internet is a vast network that connects billions of computers and other electronic devices all around the world. You can get nearly any information, interact with anyone in the globe, and do a lot more using the Internet.

8.5. Client and Server: The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-Server Model are Email, World Wide Web, etc.

9. Difference between:

9.1. TCP v/s UDP

Transmission Control Protocol (TCP)

TCP (Transmission Control Protocol) is one of the main protocols of the Internet protocol suite. It lies between the Application and Network Layers which are used in providing reliable delivery services. It is a connection-oriented protocol for communications that helps in the exchange of messages between different devices over a network. The Internet Protocol (IP), which establishes the technique for sending data packets between computers, works with TCP.

User Datagram Protocol (UDP)

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as the UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection before data transfer. The UDP helps to establish low-latency and loss-tolerating connections establish over the network. The UDP enables process-to-process communication.

9.2. http v/s https

HTTP

HTTP stands for Hyper-Text Transfer Protocol. It is invented by Tim Berner. Hyper-Text is the type of text which is specially coded with the help of some standard coding language called Hyper-Text Markup Language (HTML). HTTP provides a standard between a web browser and a web server to establish communication. It is a set of rules for transferring data from one computer to another. Data such as text, images, and other multimedia files

are shared on the World Wide Web. Whenever a web user opens their web browser, the user indirectly uses HTTP. It is an application protocol that is used for distributed, collaborative, hypermedia information systems.

HTTPS

HTTPS stands for Hyper Text Transfer Protocol Secure. HTTP Secure (HTTPS), could be a combination of the Hypertext Transfer Protocol with the SSL/TLS convention to supply encrypted communication and secure distinguishing proof of an arranged web server. HTTPS is more secure than HTTP because HTTPS is certified by the SSL (Secure Socket Layer). Whatever website you are visiting on the internet, if its URL is HTTP, then that website is not secure.

10. Difference between: (Old vs New Versions)

10.1. HTML and HTML -5

HTML	HTML - 5
No standardized process to handle structurally incorrect HTML codes.	It supports persistent error handling via improvised error handling process.
It's not mobile friendly.	It's mobile friendly.
No audio and video support in HTML	Audio and video elements can be integrated directly into a web page.
It does not support all major web browsers.	It is supported by all major web browsers.
It does not allow Java-script to run in browser.	It allows Java-script to run in background.

10.2. CSS and CSS-3

CSS	CSS - 3
CSS is the basic version with the basic formatting functionality.	CSS-3 is the latest iteration of the CSS language that extend the functionality of CSS-3.
It does not support responsive design and cannot handle media queries.	It supports responsive design and can handle media queries quite well.
CSS is relatively slower than CSS-3	CSS-3 is the evolution which is much faster than CSS.
It does not support 3D transformations and animations.	You can create 3D transformation, transitions and animations using CSS-3.
CSS can't be split into varied modules.	CSS-3 can be split into modules.
It has old and standard colours.	It offers exciting new ways to play with colours.

10.3. ES and ES-6

ES refers to ECMA script, which is a scripting language specification that Java-script follows. ES-6 or ECMA script 2015, is a specific version of ECMA script that introduced significant enhancements to Java-script, including features like arrow functions, classes and template literal. Essentially, ES-6 is a newer version of the ECMA script standard with additional functionality compared to earlier version like ES-5.

10.4. Var and Let keyword in JavaScript

Var	Let
"Var" is a keyword that is used to declare a variable.	"Let" is also a keyword that is used to declare a variable.
Syntax: var 'name' = 'value' ;	Syntax: let 'name' = 'value' ;
The variables that are defined with var statement have function scope.	The variable that are defined with let statement have block scope.
We can declare a variable again even if it has been defined previously in the same scope.	We cannot declare a variable more than once if we defined that previously in the same scope.
Hoisting is allowed with var.	
Var is an ECMA script 1 feature.	Let is a feature of ES-6.

11. What do you mean by markup Language and Scripting language.

A markup language is a text-encoding system which specifies the structure and formatting of a document and potentially the relationship between its parts. Markup is often used to control the display of the document or to enrich its content to facilitate automated processing.

Scripting languages are a specific kind of computer languages that you can use to give instructions to other software, such as a web browser, server, or standalone application. Many of today's most popular coding languages are scripting languages, such as JavaScript, PHP, Ruby, and Python.

12. Explain the various tag used in HTML to create a table with example.

To create table in HTML, you can use the following steps:

1. Start with the '`<table>`' and '`</table>`' tags. These tags define the beginning and end of the table, respectively.
2. Inside the '`<table>`' and '`</table>`' tags, you add '`<tr>`' and '`</tr>`' tags. These tags define a row in the table.
3. Inside the '`<tr>`' and '`</tr>`' tags, you can add '`<td>`' and '`</td>`' tags. These tags define a cell in the table.

You can add as many rows and Column's to your table as you need. To add a header to a row or column, you use the '`<th>`' and '`</th>`' tags respectively.

Here is an example of a simple HTML table.

code

```
<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Mukund Patidar</td>
    <td>18</td>
  </tr>
</table>
```

output

Name	Age
Mukund Patidar	18

13. How many types of lists can be created using HTML.

There are 2 types of lists in HTML unordered list and ordered list. You can use '``' and '``' tag for an unordered list or '``' and '``' tag for an ordered list. Each list item is defined using the '``' tag

Here is an example of an unordered list in HTML.

Code	Output
<pre> Item 1 Item 2 Item 3 </pre>	<ul style="list-style-type: none">• Item 1• Item 2• Item 3

Here is an example of an ordered list in HTML.

Code	Output
<pre> Item 1 Item 2 Item 3 </pre>	<ol style="list-style-type: none">1. Item 12. Item 23. Item 3

14. In how many ways you can style your HTML document (inline, internal, external CSS).

Inline CSS - Inline CSS is applied directly to an HTML element using the 'style' attribute within tag. It's useful for applying unique style to a specific element.

Example

```
<p style = "color : red ; font-size : 16px ;">This is a red, larger text.</p>
```

Internal CSS - Internal CSS is defined within the '`<style>`' tag typically placed in the '`<head>`' section of an HTML document. It's affects the style of elements on that specific page.

Example

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

```

<style>
  h1{ color: blue;
    text-align: center;}
  p{ font-size: 16px;}
</style>
</head>
<body>
  <h1>Welcome to my page</h1>
  <p>This is a paragraph with internal CSS.</p>
</body>
</html>

```

External CSS - External CSS is stored in a separated file with a '.css' extension. This file is then linked to the HTML document using '<link>' tag. External CSS promotes a clean separation of HTML & style.

Example

Styles.css

```

h1{ color: blue;
  text-align: center;}
p{ font-size: 16px;}

```

index.html

```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <h1>Welcome to my page</h1>
  <p>This is a paragraph with external CSS.</p>
</body>
</html>

```

15. What are the selectors and What is Specificity?

Selectors in CSS are patterns that define the element to which a set of style should be applied. Here are various types of selectors:

1. **Type selector** applies style to all instance of specified HTML element.

Example: Style.css

```
p{ color : blue ; }
```

Index.html

```
<p> Blue text </p>
```

2. **Class selector** selects elements with a specific class attribute.

Example: Style.css

```
.highlight{ background-color : yellow ; }
```

Index.html

```
<p class ="highlight"> This paragraph has a special class </p>
```

3. **ID selector** Select a single element of the specific ID Attribute.

Example: Style.css

```
#header { font-size : 20px ; }
```

Index.html

```
<div id ="header"> This is header </div>
```

4. **Descendent selector** selects on element that is a descendant of another specified element.

Example: Style.css

```
article p { font-style : italic ; }
```

Index.html

```
<article>
  <p> This paragraph is italic </p>
</article>
```

5. **Adjacent sibling selector** select element that is directly preceded by a specified element.

Example: Style.css

```
H2 + p { color : red ; }
```

Index.html

```
<h2> Title 2 </h2>
<p> This paragraph is red because it follows an h2 element </p>
```

6. **Attribute selector** selects element based on the presence or value of their attributes.

Example: Style.css

```
input [type ="text"] { border : 1px solid black ; }
```

Index.html

```
<input type ="text" placeholder = "Enter text">
```

7. **Pseudo classes** select element based on their state of position.

Example: Style.css

```
a : hover { color : purple ; }
```

Index.html

```
<a href ="#"> hover over me </a>
```

8. **Pseudo elements** select a part of an elementary, such as the first line or first letter.

Example: Style.css

```
p :: first-line { font-weight : bold ; }
```

Index.html

```
<p> This is the first line </p>
```

16. Explain various types of data types in JS.

There are eight basic data types in JavaScript. They are:

Data Types	Description	Example
String	represents textual data	'hello', "hello world!" etc
Number	an integer or a floating-point number	3, 3.234, 3e-2 etc.
BigInt	an integer with arbitrary precision	900719925124740999n , 1n etc.
Boolean	Any of two values: true or false	true and false
undefined	a data type whose variable is not initialized	let a;
null	denotes a null value	let a = null;
Symbol	data type whose instances are unique and immutable	let value = Symbol('hello');
Object	key-value pairs of collection of data	let student = { };

Here, all data types except Object are primitive data types, whereas Object is non-primitive.

17. How to declare array and function in JS, explain with example.

Declaration of an Array

There are basically two ways to declare an array i.e. Array Literal and Array Constructor.

1. Creating an Array using Array Literal

Creating an array using array literal involves using square brackets [] to define and initialize the array. This method is concise and widely preferred for its simplicity.

Syntax:

```
let arrayName = [value1, value2, ...];
```

example

Code	Output
// Creating an Empty Array let names = []; console.log(names);	[] ['HTML', 'CSS', 'Javascript', 'React']

```
// Creating an Array and Initializing with Values
let courses = ["HTML", "CSS", "Javascript", "React"];
console.log(courses);
```

2. Creating an Array using Array Constructor (JavaScript new Keyword)

The “Array Constructor” refers to a method of creating arrays by invoking the Array constructor function. This approach allows for dynamic initialization and can be used to create arrays with a specified length or elements.

Syntax:

```
let arrayName = new Array();
```

Example:

Code	Output
<pre>// Declaration of an empty array // using Array constructor let names = new Array(); console.log(names); // Creating and Initializing an array with values let courses = new Array("HTML", "CSS", "Javascript", "React"); console.log(courses); // Initializing Array while declaring let arr = new Array(3); arr[0] = 10; arr[1] = 20; arr[2] = 30; console.log(arr);</pre>	<pre>[] ['HTML', 'CSS', 'Javascript', 'React'] [10, 20, 30]</pre>

Declaration of a Function

JavaScript function is a set of statements that take inputs, do some specific computation, and produce output.

A JavaScript function is executed when “something” invokes it (calls it).

Example : A basic javascript function, here we create a function that divides the 1st element by the second element.

Syntax :-

Code	Output
<pre>function myFunction(g1, g2) { return g1 / g2; } const value = myFunction(8, 2); // Calling the function console.log(value);</pre>	<pre>4</pre>

18. Explain various types of data types in PHP.

Data Types define the type of data a variable can store. PHP allows eight different types of data types. All of them are discussed below. There are pre-defined, user-defined, and special data types.

The predefined data types are:

- Boolean
- Integer
- Double
- String

The user-defined (compound) data types are:

- Array
- Objects

The special data types are:

- NULL
- resource

The first five are called simple data types and the last three are compound data types

1. **Integer:** Integers hold only whole numbers including positive and negative numbers, i.e., numbers without fractional part or decimal point.
2. **Double:** Can hold numbers containing fractional or decimal parts including positive and negative numbers or a number in exponential form.
3. **String:** Hold letters or any alphabets, even numbers are included. These are written within double quotes during declaration. The strings can also be written within single quotes, but they will be treated differently while printing variables.
4. **Boolean:** Boolean data types are used in conditional testing. Hold only two values, either TRUE (1) or FALSE (0). Successful events will return *true* and unsuccessful events return *false*.
5. **Array:** Array is a compound data type that can store multiple values of the same data type.
6. **Objects:** Objects are defined as instances of user-defined classes that can hold both values and functions and information for data processing specific to the class.
7. **NULL:** These are special types of variables that can hold only one value i.e., NULL.
8. **Resources:** Resources in PHP are not an exact data type. These are basically used to store references to some function call or to external PHP resources.

19. Why JS is called object based-programming language and dynamically typed language.

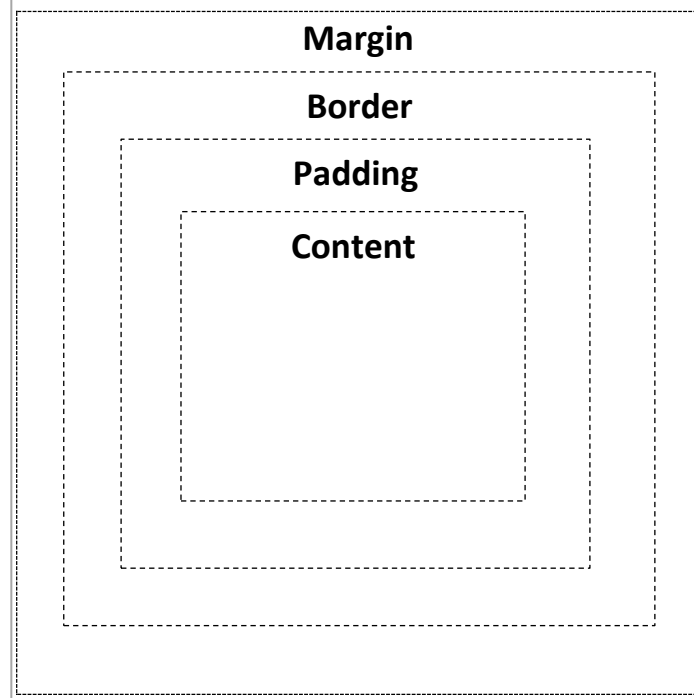
To be more precise, JavaScript is a prototype-based Object-Oriented Language, which means it doesn't have classes, rather it defines behaviours using a constructor function and then reuses it using the prototype. JavaScript is called a dynamic language because it doesn't just have a few dynamic aspects, pretty much everything is dynamic. All variables are dynamic (both in type and existence), and even the code is dynamic. You can create new variables at runtime, and the type of variables is determined at runtime.

20. Explain Box Model of CSS.

The CSS box model is a fundamental concept that describes the layout of elements in a web page. It consists of four main components: content, padding, border and margin. These components surround each HTML element, creating a box-like structure.

1. **Content** – The innermost part of the box is the content area. It holds the actual content of the element such as text, images or other media.
2. **Padding** – Padding is the space between the content and the border. It provides internal spacing and helps control the distance between the content and the border. You can set padding using properties like 'padding-top', 'padding-bottom', 'padding-right' and 'padding-left' or shorthand as 'padding'.
3. **Border** – The border surrounds the padding and content. It can be styled, coloured and sized using properties like 'border-width', 'border-style', 'border-colour' and their shorthand 'border'. The border separates the padding from the margin.
4. **Margin** – The outermost part of the box is the margin. It creates space outside the border. Influencing the distance between the element's border and its surrounding elements like padding. You can set margin using properties like 'margin-top', 'margin-bottom', 'margin-right' and 'margin-left' or shorthand as 'margin'.

illustration:



21. Differentiate between XML and HTML.

XML	HTML
The full form is extensible Markup Language	The full form is Hypertext Markup Language
The main purpose is to focus on the transport of data and saving the data	Focuses on the appearance of data. Enhances the appearance of text

XML is dynamic because it is used in the transport of data	HTML is static because its main function is in the display of data
It is case-sensitive. The upper and lower case needs to be kept in mind while coding	It is not case-sensitive. Upper and lower case are of not much importance in HTML
You can define tags as per your requirement, but closing tags are mandatory	It has its own pre-defined tags, and it is not necessary to have closing tags
XML can preserve white spaces	White spaces are not preserved in HTML
extensible Markup Language is content-driven, and not many formatting features are available	Hypertext Markup Language, on the other hand, is presentation driven. How the text appears is of utmost importance
Any error in the code shall not give the final outcome	Small errors in the coding can be ignored and the outcome can be achieved
The size of the document may be large	No lengthy documents. Only the syntax needs to be added for best-formatted output

22. What do you mean by validation of XML?

XML validation is the process of checking a document written in XML (extensible Markup Language) to confirm that it is both well-formed and also "valid" in that it follows a defined structure. A well-formed document follows the basic syntactic rules of XML, which are the same for all XML documents.

23. How namespaces are used in an XML?

An XML namespace is a collection of names that can be used as element or attribute names in an XML document. The namespace qualifies element names uniquely on the Web in order to avoid conflicts between elements with the same name.

24. What are the cookies? Writes its significance.

Cookies are small files of information that a web server generates and sends to a web browser. Web browsers store the cookies they receive for a predetermined period of time, or for the length of a user's session on a website. They attach the relevant cookies to any future requests the user makes of the web server.

25. Write short note on web socket.

A WebSocket is a communication protocol that provides full-duplex communication channels over a single TCP connection. It enables real-time, event-driven communication between a client and a server. Unlike traditional HTTP, which follows a request-response model, WebSocket's allow bi-directional communication.

26. How to link: (Syntax n examples)

26.1. **HTML with CSS:** An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the <head> section of each HTML page:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph. </p>
</body>
</html>
```

The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is what the "styles.css" file looks like:

```
body {
  background-color: powderblue;
```

```

}
h1 {
  color: blue;
}
p {
  color: red;
}

```

26.2. **HTML with JS:** To link java-script to HTML we have to use the Html `<script></script>` tag. There are two ways to link java-script to Html:

1. Embedding the java-script code inside the Html document, which is called inline java-script.

Syntax-

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
  </head>
  <body>
    <script>
      //Script logic
    </script>
  </body>
</html>

```

2. Linking external java-script files using src attribute of the `<script>` tag, this way of linking java-script is called external java-script.

Syntax-

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    //Referencing external javascript file
    <script src="script.js"></script>
  </head>
  <body>
  </body>
</html>

```

26.3. **HTML with XML:** The simple way to insert XML code into an HTML file is to use the `<xml>` tag. The XML tag informs, the browser that the contents are to be parsed and interpreted using the XML parser. Like most other HTML tags, the `<xml>` tag has attributes. The most important attribute is the ID, which provides for the unique naming of the code. The contents of the XML tag come from one of two sources: inline XML code or an imported XML file.

Example:

```

<html>
<xml id= message>
<message>
<to> Visitors </to>
<from> Author </from>
<subject> XML code island </subject>
<body> XML is embedded in html </body>
</message>
</xml>
</html>

```

26.4. **One HTML page to Another HTML Page:** Link is simply defined as text, which is clickable, so it helps to move from one page to another whenever this text has been clicked. You give the link to any element like page, image, or website to move from one page to another. HTML link uses `<a>` tag with href attribute which is having path were actually want to jump.

Syntax:

```

<a href="URL"> Text Content </a>

```

27. Short note on:

- 27.1. **HTML:** HTML stands for Hyper-Text Markup Language. It is used to design web pages using a markup language. HTML is a combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.
- 27.2. **CSS:** Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independently of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colours, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.
While HTML uses tags, CSS uses rulesets. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.
- 27.3. **JS:** JavaScript is a *lightweight, cross-platform, single-threaded, and interpreted compiled* programming language. It is also known as the scripting language for webpages. It is well-known for the development of web pages, and many non-browser environments also use it.
JavaScript is a weakly typed language (dynamically typed). JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript is both an imperative and declarative type of language. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.
- 27.4. **NodeJS:** Node JS is an open-source and cross-platform runtime environment for executing JavaScript code outside a browser. NodeJS is not a framework and it's not a programming language. Node.js is used to build back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Pay-pal, Uber, Netflix, Walmart, and so on.
- 27.5. **PHP:** The term PHP is an acronym for *PHP: Hypertext Preprocessor*. PHP is a server-side scripting language designed specifically for web development. It is open-source which means it is free to download and use. It is very simple to learn and use. The files have the extension ".php".
Rasmus Lerdorf inspired the first version of PHP and participated in the later versions. It is an interpreted language and it does not require a compiler.
- 27.6. **XML:** Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML focus on simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

28. Name the various From-end web development Technologies and tool and difference among them.

29. Explain the various tag used to create a form with example, in HTML.

30. Explain the various input tag used to create a form with example, in HTML.

31. Write the syntax to change the background colour of a tag. (All three ways with example)

32. How to declare array and function in PHP, explain with example.

33. How to declare variables in JS and PHP, explain with example.

34. Explain various pseudo-class in CSS with example.

35. Explain various operators in JS and PHP with examples.

36. Explain various control statements in JS and PHP with examples.

37. What do you mean by loops in a programming language give some examples of loops in JS and PHP.

38. Explain Following in detail:

- 38.1. Elements and attributes in XML
38.2. DTD
38.3. XML Schema
38.4. DOM (Document Object Model)
38.5. XML Parser