

1. What is HTML.?

2. Describe the meaning of the word Hyper.

- ### 3. Give a brief introduction about HTML history.

- HTML (Hypertext Markup Language) is the standard markup language used for creating and structuring web pages.
- It was first proposed by Tim Berners-Lee in 1991 while working at CERN, the European particle physics laboratory.
- The initial version, HTML 1.0, was a simple markup language that allowed for basic text formatting and the inclusion of images.
- Over the years, HTML has evolved through various versions, adding new features and capabilities to support the growing needs of the World Wide Web.
- The latest major version is HTML5, which was released in 2014 and introduced significant improvements, such as better support for multimedia, offline capabilities, and improved semantics.

4. What is an IP address.?

An Internet Protocol (IP) address is a unique numerical identifier for every device or network that connects to the internet

5. Why was IPV 6 introduced.?

IPv6 (Internet Protocol version 6) was introduced to address the depletion of IPv4 addresses and to provide a more robust and scalable internet infrastructure.

The main reasons for the introduction of IPv6 include:

- a. Address space exhaustion
- b. Improved security
- c. Simplified header format
- d. Better support for mobile devices

6. Explain the process of a web server and why do we need it.?

A web server is a software application or a computer system that handles HTTP (Hypertext Transfer Protocol) requests from clients (typically web browsers) and delivers web pages, files, and other resources over the internet or a local network.

The main process of a web server involves the following steps:

- a. Listening for incoming requests:
- b. Accepting the request:
- c. Retrieving the requested resource:
- d. Sending the response
- e. Closing the connection

- Web servers are necessary because they provide a centralized location for hosting and serving web content, enabling clients (web browsers) to access and retrieve the resources they need to display web pages.

7. What is a web browser.?

- A web browser is a software application that enables users to access, retrieve, and display web pages and other resources from the internet.
- It acts as a client in the client-server model of the World Wide Web, sending requests to web servers and receiving and rendering the responses (typically HTML, CSS, JavaScript, images, and other multimedia content) for the user to view and interact with.

8. How HTTP works and what does it really do.?

- HTTP (Hypertext Transfer Protocol) is the foundation protocol for data communication on the World Wide Web.
- It governs how web browsers (clients) and web servers communicate and exchange data.

Here's how HTTP works and what it does:

- a. Request-response model
- b. Request methods
- c. URLs and resource identification
- d. Headers
- e. Response codes
- f. Data transfer
- g. Stateless

9. What are the element function groups that we use in HTML.

In HTML, elements are typically grouped into various functional categories based on their purpose and the type of content they represent.

Here are some common element function groups:

- Structure elements: These define the overall structure and layout of the document, such as `<html>`, `<head>`, `<body>`, `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`.
- Text elements: These are used for formatting and structuring text content, such as `<h1>` to `<h6>` for headings, `<p>` for paragraphs, `<blockquote>`, `<pre>`, ``, ``, ``, and more.
- Multimedia elements: These are used for embedding various types of multimedia content, such as `` for images, `<audio>` and `<video>` for audio and video, `<iframe>` for embedding external content.
- Form elements: These are used for creating interactive forms, such as `<form>`, `<input>`, `<textarea>`, `<select>`, `<button>`, `<label>`, and more.
- List elements: These are used for creating ordered (``) and unordered (``) lists, along with list item elements (``).
- Table elements: These are used for creating tabular data structures, such as `<table>`, `<tr>`, `<th>`, and `<td>`.
- Semantic elements: These provide semantic meaning and structure to the content, such as `<main>`, `<article>`, `<nav>`, `<section>`, `<header>`, `<footer>`, and more.

10. Describe the anatomy of a HTML element

The anatomy of an HTML element typically consists of the following parts:

- Opening tag: `<element-name>` (e.g., `<p>`)
- Content: The actual content or inner HTML that the element contains (e.g., text, other elements, etc.)
- Closing tag: `</element-name>` (e.g., `</p>`)

Some elements are self-closing, meaning they don't have a closing tag and don't have any content within them. These elements are typically represented as `<element-name />` (e.g., `
`, ``, `<input />`).

Additionally, HTML elements can have attributes, which provide additional information or configuration for the element. Attributes are specified within the opening tag, like this: `<element-name attribute-name="value">`.

11. Describe the main strategies that we use for web designing.

The main strategies or approaches used in web design typically include:

- Responsive Web Design (RWD)
- Mobile-First Design
- Fluid/Liquid Layouts
- Grid-Based Layouts
- Component-Based Design
- User Experience (UX) Design
- Accessibility

12. What are meta tags and why do we want meta tags in HTML.?

Meta tags are HTML elements that provide metadata (data about data) about a web page.

They are placed within the <head> section of an HTML document and are used for various purposes:

- a. Search Engine Optimization (SEO)
- b. Content Rendering
- c. Viewport Settings
- d. Social Media Integration
- e. Refresh and Redirection
- f. Content Security Policies

- Meta tags provide valuable information and instructions to browsers, search engines, and other web services, helping to optimize the display, indexing, and sharing of web pages.

13. What is a domain name.?

A domain name is a unique, human-readable address that identifies a website or web server on the internet.

It acts as a virtual address, translating the complex numerical IP addresses (e.g., 192.168.1.1) into a more memorable and user-friendly format (e.g., [www.example.com](#)).

Domain names typically consist of two main parts:

- a. Top-Level Domain (TLD)
- b. Second-Level Domain (SLD)

14. What is DNS.?

DNS (Domain Name System) is a distributed, hierarchical naming system that translates human-readable domain names into numerical IP addresses that computers can understand and use for locating and communicating with servers on the internet.

The DNS system consists of a global network of DNS servers that maintain and update

15. What is a URL.? What parts does it contain.?

A URL (Uniform Resource Locator) is a standardized way of addressing and locating resources on the internet.

It typically contains the following parts:

- Protocol:
- Domain Name or IP Address
- Port (optional)
- Path
- Query String (optional)
- Fragment (optional)
- A typical URL structure looks like this:

16. What are HTML attributes.?

HTML attributes are additional properties or values that can be added to HTML elements to provide further information, configuration, or behavior.

Here's an example of an HTML element with attributes:

```
<a href="https://www.example.com" target="_blank" title="Example Website">Example  
Link</a>
```

17. What is the usage of HTML preamble?

- The HTML preamble, also known as the `<!DOCTYPE>` declaration, is an instruction at the beginning of an HTML document that informs the web browser about the version of HTML being used.
- It helps the browser to render the web page correctly and consistently across different browsers and platforms.



CSS

1. What is a CSS rule?

A CSS rule is a set of instructions that defines how a particular element or group of elements should be styled on a web page. A CSS rule consists of two main parts: a selector and a declaration block.

Here's an example of a CSS rule:

```
h1 {
    color: blue;
    font-size: 24px;
```

```
font-weight: bold;  
}
```

2. What is CSS reset.? and why should we use it.?

- CSS reset is a set of CSS rules that aim to normalize or reset the default styling applied by different browsers to HTML elements.
- Different browsers have their own built-in styles for various elements, which can lead to inconsistencies in rendering across browsers.

We should use a CSS reset for the following reasons:

- a. Cross-browser consistency:
- b. Predictable styling:
- c. Simplification of CSS:
- d. Improved accessibility:

3. What is the meaning of the word Cascade.? describe it.

In CSS, the term "cascade" refers to the process of determining which styles should be applied to an element when there are multiple conflicting rules.

The cascade is a fundamental concept in CSS that governs how styles are prioritized and applied.

The cascade works based on the following principles:

- a. Source order
- b. Specificity
- c. Importance
- d. Inheritance

4. What are the two types of elements in HTML.?

In HTML, there are two main types of elements:

a. Block-level elements

- eg: <div>, <h1> to <h6>, <p>, , , <form>, and

b. Inline-level elements

- eg:, <a>, , , , <input>, and <button>.

5. Describe about types of measurement units in CSS.

In CSS, there are various types of measurement units that can be used to specify values for properties like font sizes, margins, paddings, widths, heights, and more.

Here are some commonly used measurement units:

a. Absolute units:

b. Relative units:

c. Other units:

6. Describe the difference between absolute and relative positions

Absolute Positioning	Relative Positioning
<ul style="list-style-type: none">● Removes the element from the normal document flow● Positions the element relative to its nearest positioned ancestor● Can overlap with other elements● Uses top, right, bottom, left properties to set position● Useful for creating modals, dropdowns, overlays	<ul style="list-style-type: none">● Keeps the element in the normal document flow● Positions the element relative to its normal position● Does not overlap with other elements● Uses top, right, bottom, left properties to offset position● Useful for minor adjustments to an element's position

7. What is the default position of a element.?

- The default position of an HTML element is static.
- When an element is set to position: static;
- Elements with the default static position are not affected by the top, right, bottom, and left properties.

8. How box model help in web designing and what is box model, what are the areas of box model. Explain.

The CSS box model is a fundamental concept in web design that defines how elements are sized and laid out on a web page.

It describes the rectangular box that surrounds every HTML element, consisting of four main areas:

- Content area
- Padding area
- Border area
- Margin area

Understanding the box model is crucial for controlling the layout and spacing of elements on a web page.

The box model helps in web design by providing a structured approach to layout and spacing, enabling developers to create visually appealing and consistent designs.

9. How sticky positions work.?

Sticky positioning is commonly used for creating sticky headers, navigation menus, or sidebar elements that remain visible and accessible while scrolling through the page content.

10. What is the difference between sticky and fixed positions?

The main difference between position: sticky; and position: fixed; is their behavior in relation to the viewport and the document flow.

- fixed positioning keeps an element in a fixed position within the viewport at all times, while
- sticky positioning allows an element to initially follow the document flow and then become fixed when a specified scrolling threshold is reached.

11. When should we use relative positions.?

Relative positioning (position: relative;) is often used in the following scenarios:

- Minor adjustments or offsets
- Creating a positioning context
- Providing a reference point
- Preserving document flow:

12. What is em and rem.?

em and rem are relative units in CSS used for specifying font sizes, margins, paddings, and other length-based properties.

Both em and rem are useful for creating responsive and scalable designs, as they allow measurements to adjust proportionally based on the font size.

13. Explain the difference between Flex and Grid layout

Flex and Grid are both CSS layout modules that help developers create complex and responsive layouts

- Flexbox is best suited for one-dimensional layouts (rows or columns) and space distribution.
- Grid is better suited for two-dimensional layouts with explicit row and column placement.

14. How many selectors are there in CSS. Explain their usage one by one with examples.

1.Type Selector (Element Selector): Selects elements based on their tag name.

Example: `h1 { color: blue; }`

2.Class Selector: Selects elements with a specific class attribute. Example: `.my-class { font-weight: bold; }`

3.ID Selector: Selects an element with a specific ID attribute. Example: `#my-id { background-color: yellow; }`

4.Universal Selector: Selects all elements on the page. Example: `* { margin: 0; padding: 0; }`

5.Attribute Selector: Selects elements based on the presence or value of an attribute.

Example: `a[target="_blank"] { color: red; }`

6.Descendant Selector: Selects elements that are descendants (children, grandchildren, etc.) of another element. Example: `div p { font-size: 14px; }`

7.Child Selector: Selects elements that are direct children of another element.

Example: `ul > li { list-style-type: none; }`

8.Adjacent Sibling Selector: Selects elements that are adjacent siblings (immediately following) of another element. Example: `h1 + p { margin-top: 10px; }`

9.General Sibling Selector: Selects elements that are siblings (preceded or followed by) of another element. Example: `h2 ~ p { color: gray; }`

10.Pseudo-class Selector: Selects elements based on a specific state or condition. Example: `a:hover { text-decoration: underline; }`

11.Pseudo-element Selector: Selects and styles a specific part of an element. Example: `p::first-line { font-weight: bold; }`

12.Combinators: These selectors allow you to combine multiple selectors for more specific targeting. Example: `div.my-class p#my-id { /* Styles */ }`

15. Explain how the cascades are resolved step by step.

Step 1: Determine the origin and importance of each rule.

Step 2: Sort the rules by their specificity.

Step 3: Apply the rule with the highest specificity.

Step 4: Inherit values from parent elements (if applicable).

Step 5: Apply the default (user agent) styles.

16. What is a transition.?

In CSS, a transition is a way to control the animation-like behavior when a property value changes from one state to another.

17. What are keyframes in CSS.?

In CSS, keyframes are a way to define the intermediate steps or frames of an animation sequence.

They allow you to specify the styles for different points in the animation, creating a smooth transition between those keyframes.

18. Explain about media queries and write down a media query to change styles of a web site when it is viewed from a 320px to 375px device.

Media queries are a CSS technique that allows you to apply different styles based on the characteristics of the device or viewport, such as screen size, resolution, orientation, or other media features.

This enables the creation of responsive designs that adapt to different devices and screen sizes.

19. Explain about CSS inheritance with a example.

CSS inheritance is a mechanism by which some CSS properties are automatically applied to an element's descendants (child elements, grandchild elements, and so on) unless explicitly overridden.

This behavior allows for more efficient and maintainable styling, as you don't need to specify the same property values for every element in the hierarchy.

20. Explain how the browser perform the rendering process of a web page

This process involves several steps:

- Parsing
- Rendering Tree Construction
- Layout
- Painting
- Compositing
- GPU Acceleration
- Continuous Process

Java Script

1. What is java script.?

JavaScript is a high-level, interpreted programming language primarily used for creating dynamic and interactive web pages.

JavaScript is responsible for adding interactivity, animations, effects, and behavior to web pages, enhancing the user experience and making websites more engaging and responsive.

2. Explain about the history of Java Script.

- The history of JavaScript dates back to the early days of the World Wide Web in 1995.
- Initially, it was created by Brendan Eich at Netscape Communications Corporation under the name "Mocha," later renamed "LiveScript." However, to capitalize on the popularity of Sun Microsystems' Java language, it was finally renamed "JavaScript" before its official release in 1997 with Netscape Navigator 2.0.
- Today, JavaScript is not limited to just web browsers

3. What is ECMAScript.?

ECMAScript (or ES) is the standardized specification and scripting-language specification upon which JavaScript is based.

It is maintained by the ECMA International organization, a standards association responsible for defining and publishing standards for information and communication technologies.

ECMAScript defines the syntax, types, statements, keywords, reserved words, operators, objects, and other language elements of JavaScript.

4. What are the methods of creating valid identifiers in JS.?

In JavaScript, there are several methods for creating valid identifiers (names for variables, functions, objects, etc.). Here are the main rules:

Here are some examples of valid identifiers in JavaScript:

```
let myVariable;  
const _privateVariable;  
var $elementId;  
function calculateArea() { /* ... */ }  
let userName = "John Doe";  
const PI = 3.14159;
```


5. What is the difference between var, let and const.?

var	let	const
<ul style="list-style-type: none">• Function or global scope• Can be re-declared and updated• Hoisted to the top of the scope, but not initialized	<ul style="list-style-type: none">• Block scope• Cannot be re-declared in the same scope• Can be updated• Not hoisted, so cannot be accessed before declaration	<ul style="list-style-type: none">• Block scope• Cannot be re-declared or re-assigned• Must be initialized at declaration• Used for declaring constants/immutable values

6. Why JS is known as a dynamic type language.?

JavaScript is known as a "dynamic type" or "loosely typed" language because it does not require explicit type declarations for variables.

Instead, variable types are determined automatically at runtime based on the values assigned to them.

This dynamic typing behavior is in contrast to statically typed languages like Java, C++, or C#, where variables must be declared with their specific data types before use.

7. Why JS was a loosely types language.?

JavaScript was designed as a loosely typed language (also known as dynamically typed) for several reasons:

- a. Simplicity and flexibility
- b. Prototyping and experimentation
- c. Compatibility with the web
- d. Historical reasons

8. What is variable hoisting.?

Variable hoisting is a mechanism in JavaScript where variable and function declarations are moved to the top of their respective scopes (either global or function scope) during the compilation phase, before the code is executed. However, only the declarations are hoisted, not the initializations or assignments.

9. What are the two ways of creating an array in JS.?

- a. Array literal syntax
- b. Array constructor

10. When we should use the indexOf method in JS arrays and how does it works?

The `indexOf()` method is used in JavaScript arrays to find the index of the first occurrence of a specific value within the array.

It is useful when you need to check if an element exists in the array and get its position (index).

Here's how the `indexOf()` method works:

- `const fruits = ['apple', 'banana', 'orange', 'apple'];`
- `const index = fruits.indexOf('apple');` // Returns 0 (the index of the first 'apple')
- `const notFound = fruits.indexOf('pear');` // Returns -1 (value not found)

11. What are the methods that we can use to manipulate data in JS arrays?

- `push()`: Adds one or more elements to the end of the array and returns the new length of the array.
- `pop()`: Removes the last element from the array and returns it.
- `shift()`: Removes the first element from the array and returns it.
- `unshift()`: Adds one or more elements to the beginning of the array and returns the new length of the array.
- `splice()`: Changes the contents of an array by removing or replacing existing elements and/or adding new elements.
- `slice()`: Returns a shallow copy of a portion of an array into a new array object.
- `concat()`: Merges two or more arrays and returns a new array.
- `join()`: Joins all elements of an array into a string, separated by a specified separator (comma by default).
- `reverse()`: Reverses the order of the elements in the array.
- `sort()`: Sorts the elements of an array in place and returns the sorted array.
- `map()`: Creates a new array with the results of calling a provided function on every element in the calling array.
- `filter()`: Creates a new array with all elements that pass the test implemented by the provided function.
- `reduce()`: Applies a function against an accumulator and each element in the array to reduce it to a single value.

12. How to add a value to the last index of an array?

To add a value to the last index of an array in JavaScript, you can use the `push()` method. The `push()` method appends one or more elements to the end of an array and returns the new length of the array.

13. What are string related methods that you know?

- `length`: Returns the length of the string.

- `charAt()`: Returns the character at a specified index position within the string.
- `charCodeAt()`: Returns the Unicode value of the character at a specified index position.
- `concat()`: Joins two or more strings and returns a new string.
- `includes()`: Checks if a string contains a specified substring and returns true or false.
- `indexOf()`: Returns the position of the first occurrence of a specified value in a string, or -1 if not found.
- `lastIndexOf()`: Returns the position of the last occurrence of a specified value in a string, or -1 if not found.
- `match()`: Searches for a match against a regular expression and returns an array of matched substrings.
- `replace()`: Replaces a specified value or regular expression within a string with a new value.
- `search()`: Searches for a match against a regular expression and returns the position of the first match, or -1 if not found.
- `slice()`: Extracts a section of a string and returns a new string.
- `split()`: Splits a string into an array of substrings based on a specified delimiter.
- `startsWith()`: Checks if a string starts with a specified substring and returns true or false.
- `endsWith()`: Checks if a string ends with a specified substring and returns true or false.
- `substring()`: Returns a subset of a string between specified start and end indexes.
- `toLowerCase()`: Converts the string to lowercase letters.
- `toUpperCase()`: Converts the string to uppercase letters.
- `trim()`: Removes whitespace characters from both ends of a string.

14. What are the data types in JS.?

- Primitive Data Types (eg: number, string, boolean)
- Non-Primitive (Reference) (Data Types - eg: object)

15. What is bigint.?

bigint is a new numeric data type introduced in ECMAScript 2020 (ES11).

It is used to represent integers of arbitrary length, beyond the safe range of the number data type.

16. What is the difference between null and undefined?

1.null represents an intentional non-value or the intentional absence of any object value. It is often used to represent an empty or non-existent reference, such as when a variable is explicitly assigned the value null.

2.undefined, on the other hand, represents a variable that has been declared but has not been assigned a value, or an object property that does not exist.

17. What is a DOM.? Explain.

- DOM stands for Document Object Model.
- It is a programming interface for web documents that represents the structure of a web page as a tree-like hierarchy of nodes.
- The DOM provides a structured and standardized way to access and manipulate the elements, attributes, and content of a web page using programming languages like JavaScript.
- The DOM represents the entire web page as a tree of objects, with the document object at the root.

18. What is the difference between for in and for of.?

- for...in loop: The for...in loop is used to iterate over the enumerable properties of an object. It loops over the property names or keys of the object.
- for...of loop: The for...of loop was introduced in ES6 (ES2015) and is used to iterate over the values of iterable objects, such as arrays, strings, or other objects implementing the iterable protocol.

19. What are literals.? and what is a literal base object.?

There are several types of literals:

- Numeric Literals:
- String Literals:
- Boolean Literals:
- Undefined Literal:
- Object Literals:
- Array Literals:
- Regular Expression Literals:
- Template Literals (ES6):

A literal base object is an object created using the object literal notation `{}`. It is the simplest way to create an object in JavaScript.

20. What are the two types of functions in JS. Explain with examples.

1.Regular Functions

Example:

```
function greet(name) {  
  console.log(`Hello, ${name}!`);  
}
```

```
greet('John'); // Output: Hello, John!
```

2.Arrow Functions

Example:

```
const greet = (name) => {
```

```
console.log(`Hello, ${name}!`);  
}  
  
greet('John'); // Output: Hello, John!
```

21. What is a event object. Explain.?

- In JavaScript, an event object is a object that is automatically passed as an argument to an event handler function when an event occurs.
- It contains properties and methods related to the specific event that has been triggered.
- The event object provides information about the event, such as the type of event, the target element, any data associated with the event, and more.

22. What is jQuery.?

jQuery is a popular open-source JavaScript library that simplifies client-side scripting of HTML and provides a set of functions and methods for DOM manipulation, event handling, animation, and AJAX (Asynchronous JavaScript and XML) interactions.

23. What are the benefits of using jQuery?

- Cross-browser Compatibility
- Simplified DOM Manipulation
- Event Handling
- Animations and Effects
- AJAX Support
- Extensive Plugin Ecosystem
- Compact Size
- Large Community and Documentation

24. What are the traversing methods that you have used in jQuery and when do we need them.?

Here are some commonly used traversing methods in jQuery, along with their use cases:

1. `parent()`
2. `parents()`
3. `children()`
4. `find()`
5. `siblings()`
6. `next()`
7. `prev()`
8. `closest()`
9. `filter()`

25. Why JS is introduced to have first class functions.?

The reasons for introducing first-class functions in JavaScript include:

1. Higher-Order Functions
2. Callbacks and Asynchronous Programming
3. Abstraction and Code Reusability
4. Function Composition
5. Closures
6. Functional Programming