Backpack 01

HTML

1. **Question:** What is a marquee in HTML?

   Answer: Marquee is used for scrolling text on a web page. It scrolls the image or text up, down, left, or right automatically. To apply for a marquee, you have to use </marquee> tags.

2. **Question:** Are the HTML tags and elements the same thing?

   Answer: No, HTML tags are used to define the structure of a web page, while HTML elements are made up of a set of tags that define a specific part of a web page.

3. **Question:** What are void elements in HTML?

   Answer: Void elements in HTML are tags that do not require a closing tag. They are used to insert images, line breaks, and other content that does not require additional information.

4. **Question:** What is the advantage of collapsing white space?
5.

   Answer: Collapsing white space in HTML can help to reduce the size of web pages and make them load faster. It involves removing unnecessary white space between HTML elements.

6. **Question:** What is semantic HTML??

   Answer: Semantic HTML is a coding style. It is the use of HTML markup to reinforce the semantics or meaning of the content.

   For example: In semantic HTML <b> </b> tag is not used for bold statement as well as <i> </i> tag is not used for italic. Instead of these we use <strong></strong> and <em></em> tags.

CSS

1. **Question:** What is the origin of CSS ?

   Answer: Standard Generalized Markup Language marked the beginning of style sheets in the 1980s.

2. **Question:** What are the different variations of CSS ?

   Answer: The variations for CSS are: CSS 1 CSS 2 CSS 2.1 CSS 3 CSS 4

3. **Question:** What are the different variations of CSS ?

   Answer: The variations for CSS are: CSS 1 CSS 2 CSS 2.1 CSS 3 CSS 4

4. **Question:** Explain the concept of specificity in CSS?

   Answer: The concept of specificity in CSS determines which styles will be applied to an element when there are conflicting rules. Specificity is calculated based on the selectors used in the CSS rules. It follows a specific hierarchy where inline styles have the highest specificity, followed by IDs, classes, and finally, element selectors.

   The more specific a selector is, the higher its specificity value. When multiple conflicting rules target the same element, the one with the highest specificity will take precedence.

5. **Question:** What is the box model in CSS?

   Answer: The Box Model in CSS refers to the concept that organizes and structures HTML elements on a web page in the form of rectangular boxes. Every element in a page is comprised of a rectangular box, which includes content, padding, border, and margin. These components contribute collectively to the element's dimensions and positioning.

6. **Question:** Explain the concept of CSS specificity and inheritance?

   Answer: CSS specificity refers to the set of rules that determines which CSS styles should be applied to an element when multiple styles are defined. It is a way of resolving conflicts when different selectors target the same element. CSS inheritance, on the other hand, allows styles to be passed from parent elements to their descendants. Inherited properties are applied to child elements unless overridden by specific styles. The specificity of a selector is determined by the combination of its components such as element type, class, ID, and inline styles. Specificity is calculated based on a scoring system, where certain components have higher weight than others. For example, in the selector h1.title, the class component (.title) has a higher specificity than the element type (h1).

7. **Question:** How do you handle browser-specific CSS prefixes and what tools can help automate this process?

   Answer: Handling browser-specific CSS prefixes can be tedious as different browsers often require different prefixes for certain CSS properties. However, there are tools that can automate this process and help manage browser compatibility. One such is Autoprefixer.

   Autoprefixer is a CSS post-processor that automatically adds vendor prefixes to CSS properties based on the specified browser support configuration. It analyzes CSS code and adds the necessary prefixes, so you don't have to write them manually.

   To use Autoprefixer, you can integrate it into your build process or use it as a standalone tool. It's commonly used as a plugin in build tools like Gulp or webpack or as part of a CSS preprocessor like Sass or Less.

   By using Autoprefixer, you can write clean, standardized CSS code without worrying about browser-specific prefixes as the tool takes care of it for you.

8. **Question:** What is the RGB stream??

   Answer: RGB is a system of representing a certain color in CSS. There are three streams in this nomenclature representing a color, namely the Red, Green and Blue stream. The intensity of the three colors is represented in numbers ranging from 0 to 256. This allows CSS to have a wide range of colors spreading across the entire spectrum of visible colors.

9. **Question:** Explain the scenario you would use translate() instead of absolute positioning?

   Answer: Translate is a value of CSS transform. Changing transform or opacity does not trigger browser reflow or repaint but does trigger compositions; whereas changing the absolute positioning triggers reflow. Transform causes the browser to create a GPU layer for the element but changing absolute positioning properties uses the CPU. Hence translate() is more efficient and will result in shorter paint times for smoother animations. When using translate(), the element still occupies its original space (sort of like position: relative), unlike in changing the absolute positioning.

10. **Question:** What is the difference between the usage of an ID and a Class?

    Answer:
    ID – An ID is unique. A particular ID can be only assigned to a single element. IDs are used when specific styling is being tried to be achieved over a single element. Below is a pictorial example of how to use an ID.

    Class – Just like the word suggests, a class is a collective way of targetting HTML elements for styling. Classes are not unique and multiple elements can have the same class. In fact, multiple classes can also be added to the same element to achieve the desired style and look. Below is an example of the usage of classes.

11. **Question:** Name a few CSS style components?

    Answer: A few CSS style components include Property, Value, and Selector.

12. **Question:** How Do You Select All Elements in a Paragraph?

    Answer: You can select all elements in a paragraph by using the p[lang] command.

13. **Question:** What is the ruleset?

    Answer: Rulesets can be used to identify selectors attached to one another. It consists of two different parts - selector and declaration.

14. **Question:** How case-sensitive is CSS??

    Answer: CSS is not case-sensitive, but the URLs of images and font families are case-sensitive. CSS is case-sensitive only for XML declarations with XHTML DOCTYPE on the page.

15. **Question:** What is the difference between physical and logical tags??

    Answer: Logical tags are older compared to physical ones and mainly focus on the content. They hardly find any usage in terms of presentation. Logical tags do not find any application in terms of aesthetics, while the physical ones find their application in presentation too.

JavaScript

1. **Question:** What are the data types in JavaScript?

   Answer: JavaScript has six primitive data types: string, number, boolean, null, undefined, and symbol, along with a complex data type called object.

2. **Question:** What is the difference between null and undefined?

   Answer: null represents the intentional absence of any object value, while undefined indicates the absence of a value or an uninitialized variable.

3. **Question:** What is the DOM in JavaScript?

   Answer: The Document Object Model (DOM) is a programming interface that represents the structure of HTML and XML documents. It allows JavaScript to access and manipulate the content and structure of a webpage.

4. **Question:** What is an anonymous function in JavaScript?

   Answer: An anonymous function is a function without a name. It can be assigned to a variable or passed as an argument to another function. They are often used for onetime or callback functions.

5. **Question:** What is the difference between == and === in JavaScript??

   Answer: The == operator checks for equality after performing type coercion, while the === operator checks for equality without type coercion, ensuring both the value and type match.

6. **Question:** What is hoisting in JavaScript?

   Answer: Hoisting is a JavaScript behavior where variable and function declarations are moved to the top of their containing scope during the compilation phase, allowing them to be used before they are declared.

7. **Question:** What are the different ways to define a function in JavaScript?

   Answer: Functions in JavaScript can be defined using function declarations, function expressions, arrow functions, and methods within objects.

8. **Question:** What are template literals in JavaScript?

   Answer: Template literals, denoted by backticks (`), are a way to create strings in JavaScript that support interpolation of variables and multi-line strings

9. **Question:** What are JavaScript promises?

   Answer: Promises are used for asynchronous programming in JavaScript. They represent the eventual completion (or failure) of an asynchronous operation and allow chaining of operations using .then() and .catch().

10. **Question:** What is the async/await syntax in JavaScript?

    Answer: The async/await syntax is a modern approach to handle asynchronous operations. It allows writing asynchronous code in a more synchronous-like manner, making it easier to read and maintain.

11. **Question:** What is a callback function in JavaScript?

    Answer: A callback function is a function that is passed as an argument to another function and gets executed at a later time or in response to an event. It enables asynchronous and event-driven programming.

12. **Question:** What is the purpose of the try/catch statement in JavaScript?

    Answer: The try/catch statement is used for error handling in JavaScript. It allows catching and handling exceptions that occur during the execution of a block of code.

13. **Question:** What is event bubbling in JavaScript?

    Answer: Event bubbling is a mechanism in which an event triggered on a specific element will also trigger the same event on all of its parent elements. It starts from the innermost element and propagates upwards in the DOM tree.

14. **Question:** What is the difference between null and undefined?

    Answer: null is an explicitly assigned value that represents the absence of an object, while undefined is a value assigned by the JavaScript engine to variables that have been declared but have not been assigned a value

15. **Question:** What is event propagation in JavaScript?

    Answer: Event propagation is the process of an event being triggered on an element and then propagating to its parent elements or capturing down from its parent elements. It allows handling events at different levels of the DOM hierarchy

16. **Question:** What is the purpose of the addEventListener() method in JavaScript??

    Answer: The addEventListener() method is used to attach an event listener to an element. It allows you to listen for specific events and execute a function when the event is triggered.

17. **Question:** What is the purpose of the addEventListener() method in JavaScript??

    Answer: The addEventListener() method is used to attach an event listener to an element. It allows you to listen for specific events and execute a function when the event is triggered.

18. **Question:** What is the purpose of the parentNode property in JavaScript?

    Answer: The parentNode property is used to access the parent node of an element in the DOM. It allows traversal and manipulation of the DOM tree by accessing the immediate parent of an element.

19. **Question:** What is the purpose of the Math.ceil() function in JavaScript?

    Answer: The Math.ceil() function is used to round a number up to the nearest integer. It increases the number to the next higher integer, regardless of the decimal part.

20. **Question:** What is the purpose of the isFinite() function in JavaScript?

    Answer: The isFinite() function is used to check if a value is a finite number. It returns true if the value is a finite number, and false otherwise. It also returns false for NaN, Infinity, and -Infinity