Backpack-1

Part-1 | HTML

Semantic Tags

Q1. What are semantic tags in HTML?

Ans: Semantic tags are HTML elements that give meaning to the content. For example, **<header>** is used for the top part of a webpage, and **<footer>** is used for the bottom part. They help browsers and developers understand the structure of the page.

Q2. Why should you use semantic tags?

Ans: Semantic tags make your website easier to read for both people and computers. They help screen readers for blind users and help search engines understand the structure of the webpage, which can improve SEO.

Q3. Name five semantic tags in HTML.

Answer:

- <header> - For the top part of a page.
- <footer> - For the bottom part of a page.
- <article> - For a blog post or news story.
- <section> - For a section of a page.
- <nav> - For a menu or navigation links.

Q4. What is the difference between <div> and <section>?

Ans: <div> is just a container with no meaning, while <section> organizes related content and makes the page structure clearer. Semantic tags like <header> or <article> tell you exactly what the content is, making it easier to understand.

Attributes

Q5. What are HTML attributes?

Answer: HTML attributes provide additional information about HTML elements. They are always specified in the start tag, usually come in name/value pairs like name="value" and define properties like size, color, behavior, or relationships. For example, href is an attribute of the <a> tag.

Q6. What is the purpose of the alt attribute in the <img> tag?

Answer: The alt attribute provides alternative text for an image if it cannot be displayed, improving accessibility and SEO. It is also used by screen readers to describe the image for visually impaired users.

Q7. What is the difference between id and class attributes?

Answer: id is used to give one element a unique name, like a person’s name. class is used to group multiple elements together, like putting people in the same team.

Q8. What is the src attribute used for?

Answer: The src attribute specifies the source URL of an external resource, such as an image in the <img> tag or a script in the <script> tag.

HTML Elements

Q9. What is an HTML element?

Answer: An HTML element is a component of an HTML document that consists of a start tag, content, and an end tag. For example, <p>This is a paragraph.</p> is a paragraph element.

Q10. What is the difference between block-level and inline elements?

Answer: Block-level elements (e.g., <div>, <p>) take up the full width of their container and start on a new line, while inline elements (e.g., <span>, <a>) only take up as much width as necessary and do not start on a new line.

Q11. What are empty elements in HTML?

Answer: These elements don't have closing tags. Examples:

- <br> (line break)
- <img> (image)
- <input> (form field)

Q12. What is the difference between <ul> and <ol>?

Answer: <ul> is used to create an unordered list (bulleted list), while <ol> is used to create an ordered list (numbered list).

Forms and Inputs

Q13. What is the purpose of the <form> tag?

Answer: The <form> tag is used to create a form where users can enter information, like a login form or a contact form.

Q14. What is the difference between GET and POST methods in forms?

Answer: GET sends form data in the website's URL, which is visible to everyone. POST sends the data secretly, which is better for passwords or private information.

Q15. What is the purpose of the <input> tag?

Answer: The <input> tag is used to create different types of fields, like text boxes, checkboxes, or buttons, depending on the type attribute.

Q16. What is the use of the placeholder attribute in an input field?

Answer: The placeholder attribute shows a hint inside the input field, like "Enter your name," to help users know what to type.

Media

Q17. How do you embed an image in HTML?

Answer: Use the <img> tag with the src attribute to specify the image file. For example: <img src="picture.jpg" alt="A beautiful sunset">.

Q18. What is the purpose of the <audio> tag?

Answer: The <audio> tag is used to add sound to a webpage. You can add controls like play, pause, and volume using the controls attribute.

Q19. How do you embed a video in HTML?

Answer: Use the <video> tag with the src attribute to specify the video file. For example: <video src="movie.mp4" controls></video>.

Q20. What is the difference between <iframe> and <embed>?

Answer: <iframe> is used to show another webpage inside your page, like embedding a Google Map. <embed> is used to add things like PDFs or plugins, but <iframe> is more commonly used.

Part-2 | CSS

Selectors

Q1. What is a CSS selector?

Answer: A CSS selector is used to select HTML elements you want to style. For example, p selects all <p> elements, and .class selects all elements with that class.

Q2. What is the difference between a class and an ID selector?

Answer: A class selector (.class) can be used for multiple elements, while an ID selector (#id) is unique and can only be used for one element on a page.

Q3. What is a pseudo-class in CSS?

Answer: A pseudo-class is used to style elements in a specific state. For example, a:hover changes the style of a link when you hover over it with your mouse.

Box Model

Q4. What is the CSS box model?

Answer: The CSS box model is a box that wraps around every HTML element. It includes the content, padding (space inside), border (edge), and margin (space outside).

Q5. What is the difference between padding and margin?

Answer: Padding is the space inside an element, between the content and the border. Margin is the space outside the element, between the border and other elements.

Q6. How do you calculate the total width of an element in the box model?

Answer: Total width = content width + padding + border + margin. For example, if the content is 100px, padding is 10px, border is 5px, and margin is 20px, the total width is 135px.

Positioning and Layout

Q7. What is the difference between position: relative and position: absolute?

Answer: position: relative moves an element relative to its normal position, while position: absolute moves it relative to its nearest positioned ancestor (or the whole page if none exists).

Q8. What does position: fixed do?

Answer: position: fixed keeps an element in the same place on the screen, even when you scroll the page. It’s often used for navigation bars or buttons.

Q9. What is the purpose of display: flex?

Answer: display: flex is used to create a flexible layout where elements can grow, shrink, or align in a row or column. It makes it easier to design responsive layouts.

Responsive Design

Q10. What is a media query in CSS?

Answer: A media query is used to apply different styles based on the screen size or device. For example, you can change the layout for mobile phones using @media (max-width: 600px).

Q11. What is the purpose of viewport in responsive design?

Answer: The viewport meta tag tells the browser how to adjust the page to fit the screen size. It's important for making websites look good on mobile devices.

Q12. What is the difference between em and rem units?

Answer: em is relative to the font size of the parent element, while rem is relative to the font size of the root element (<html>). rem is more predictable for sizing.

Styling

Q13. How do you change the color of text in CSS?

Answer: Use the color property. For example, color: red; makes the text red.

Q14. What is the difference between border and outline?

Answer: A border is part of the box model and takes up space, while an outline is drawn outside the element and doesn't affect layout. Outlines are often used for focus states.

Q15. How do you add a shadow to an element in CSS?

Answer: Use the box-shadow property. For example, box-shadow: 5px 5px 10px gray; adds a gray shadow that's 5px to the right, 5px down, and 10px blurry.

Part 3: JavaScript

DOM Manipulation

Q1. What is the DOM?

Answer: The DOM (Document Object Model) is a tree-like structure that represents the HTML of a webpage. JavaScript can use the DOM to change the content, structure, or style of a webpage.

Q2. How do you select an element by its ID in JavaScript?

Answer: Use document.getElementById("id"). For example, document.getElementById("header") selects the element with the ID "header".

Q3. How do you change the text of an element?

Answer: Use the textContent property. For example, document.getElementById("demo").textContent = "Hello World"; changes the text of the element with ID "demo".

Q4. How do you add a new element to the DOM?

Answer: Use document.createElement() to create a new element and appendChild() to add it to the DOM. For example:

```
let newElement = document.createElement("p");
newElement.textContent = "This is a new paragraph";
document.body.appendChild(newElement);
```

Q5. How do you change the style of an element using JavaScript?

Answer: Use the style property. For example, document.getElementById("demo").style.color = "red"; changes the text color of the element with ID "demo" to red.

Control Flow

Q6. What is an if statement in JavaScript?

Answer: An if statement checks a condition and runs a block of code if the condition is true. For example:

```
if (age > 18) {
  console.log("You are an adult.");
}
```

Q7. What is a for loop?

Answer: A for loop repeats a block of code a specific number of times. For example:

```
for (let i = 0; i < 5; i++) {
  console.log(i);
}
```

Q8. What is the difference between == and ===?

Answer: == checks if two values are equal, but it ignores the data type (e.g., 5 == "5" is true). === checks if both the value and data type are the same (e.g., 5 === "5" is false).

Q9. What is a switch statement?

Answer: A switch statement checks a value against multiple cases and runs the code for the matching case. For example:

```
switch (day) {
  case "Monday":
    console.log("Start of the week");
    break;
  case "Friday":
    console.log("Weekend is near");
    break;
}
```

Q10. What is a while loop?

Answer: A while loop repeats a block of code as long as a condition is true. For example:

```
let i = 0;
while (i < 5) {
  console.log(i);
  i++;
}
```

ES6 Features

Q11. What is let and const in ES6?

Answer: let and const are used to declare variables. let allows you to change the value later, while const is for values that cannot be changed.

Q12. What is an arrow function?

Answer: An arrow function is a shorter way to write functions. For example:

```
const add = (a, b) => a + b;
```

Q13. What is template literals in ES6?

Answer: Template literals use backticks (``) to create strings that can include variables. For example:

```
let name = "John";
console.log(`Hello, ${name}!`);
```

Q14. What is destructuring in ES6?

Answer: Destructuring allows you to extract values from arrays or objects into variables. For example:

```
let [a, b] = [1, 2]; // a = 1, b = 2
let {name, age} = {name: "John", age: 30}; // name = "John", age = 30
```

Q15. What is the spread operator?

Answer: The spread operator (...) is used to expand elements of an array or object. For example:

```
let arr1 = [1, 2, 3];
let arr2 = [...arr1, 4, 5]; // arr2 = [1, 2, 3, 4, 5]
```

Q16. What is a class in ES6?

Answer: A class is a blueprint for creating objects with shared properties and methods. For example:

```
class Person {
  constructor(name) {
    this.name = name;
  }
  greet() {
    console.log(`Hello, ${this.name}`);
  }
}
```

Q17. What are default parameters in ES6?

Answer: Default parameters allow you to set default values for function arguments. For example:

```
function greet(name = "Guest") {
  console.log(`Hello, ${name}`);
}
```

Q18. What are Promise and async/await in ES6?

Answer: A Promise is used for asynchronous operations, and async/await makes it easier to work with promises. For example:

```
async function fetchData() {
  let response = await fetch("https://api.example.com/data");
  let data = await response.json();
  console.log(data);
}
```

Q19. What is an API?

Answer: An API (Application Programming Interface) is a way for two programs to communicate. For example, a weather API can provide weather data to your app.

Q20. What is fetch in JavaScript?

Answer: fetch is used to make HTTP requests to APIs. For example:

```
fetch("https://api.example.com/data")
  .then(response => response.json())
  .then(data => console.log(data));
```

Q21. What is the difference between GET and POST requests?

Answer: GET requests are used to retrieve data from a server, while POST requests are used to send data to a server.

Q22. What is JSON?

Answer: JSON (JavaScript Object Notation) is a format for storing and exchanging data. It looks like a JavaScript object but is written as a string.

Q23. How do you handle errors in fetch?

Answer: Use .catch() to handle errors in fetch. For example:

```
fetch("https://api.example.com/data")
  .then(response => response.json())
  .catch(error => console.error("Error:", error));
```

Q24. What is CORS?

Answer: CORS (Cross-Origin Resource Sharing) is a security feature that allows or blocks requests from one website to another. It prevents unauthorized access to resources.

Q25. What is localStorage in JavaScript?

Answer: localStorage is used to store data in the browser that persists even after the browser is closed. For example:

```
localStorage.setItem("name", "John");
console.log(localStorage.getItem("name")); // Output: John
```