

# Backpack Questions - HTML, CSS, JS

## Part 1: HTML (20 Questions)

### Semantic Tags

1. **What are semantic HTML tags?**

Semantic HTML tags provide meaning to the content they wrap. Examples include `<header>`, `<footer>`, `<article>`, and `<section>`, making it easier for search engines and developers to understand the page structure.

2. **What is the `<article>` tag used for?**

The `<article>` tag is used for self-contained content that can stand alone, such as a blog post or news article.

3. **What is the difference between `<section>` and `<div>`?**

`<section>` is a semantic tag that groups related content, while `<div>` is a non-semantic tag used purely for layout purposes.

4. **Why should you use semantic tags in HTML?**

Semantic tags enhance accessibility, SEO, and make the code more understandable by defining the purpose of the content.

### Attributes

5. **What is an attribute in HTML?**

An attribute is additional information provided inside the tag, modifying its behavior. For example, `class` or `id`.

6. **What is the purpose of the `alt` attribute in images?**

The `alt` attribute provides alternative text for an image, improving accessibility and SEO.

7. **How do you create a hyperlink in HTML?**

Use the `href` attribute in an `<a>` tag, e.g., `<a href="...">`

`href="https://www.example.com">Link</a>.`

**8. How do you use the `target` attribute in an anchor tag?**

The `target` attribute specifies where to open the linked document. For example, `target="_blank"` opens it in a new tab.

## HTML Elements

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

Block-level elements (e.g., `<div>`, `<p>`) take up the full width, while inline elements (e.g., `<span>`, `<a>`) only take up as much width as their content.

**10. What are void elements in HTML?**

Void elements are elements that do not have closing tags. Examples include `<img>`, `<br>`, and `<input>`.

**11. What is the purpose of the `<meta>` tag?**

The `<meta>` tag provides metadata about the HTML document, such as charset, author, and description, which helps in SEO and page rendering.

**12. How do you create an ordered list in HTML?**

Use the `<ol>` tag for ordered lists and `<li>` for each list item, e.g., `<ol><li>Item 1</li><li>Item 2</li></ol>`.

## Forms and Inputs

**13. How do you create a form in HTML?**

Use the `<form>` tag and include input elements like `<input>`, `<textarea>`, `<select>`, and `<button>`. For example, `<form action="/submit" method="POST">`.

**14. What is the purpose of the `method` attribute in forms?**

The `method` attribute specifies how the form data is sent to the server (GET or POST).

**15. How do you create a checkbox input in HTML?**

Use the `<input type="checkbox">` tag for checkboxes, e.g., `<input type="checkbox" name="subscribe" value="yes">`.

**16. How do you associate a label with a form element?**

Use the `for` attribute in the `<label>` tag to associate it with an `id` of an input, e.g.,  
`<label for="name">Name</label><input id="name" type="text">`.

## Media

**17. How do you embed an image in HTML?**

Use the `<img>` tag with `src` and `alt` attributes, e.g., ``.

**18. How do you embed a video in HTML?**

Use the `<video>` tag with the `src` and `controls` attributes, e.g., `<video src="video.mp4" controls></video>`.

**19. What is the purpose of the `<audio>` tag?**

The `<audio>` tag embeds sound content on a webpage, with controls like play, pause, and volume.

**20. How do you add a YouTube video to your HTML page?**

You can embed a YouTube video using the `<iframe>` tag, e.g., `<iframe src="https://www.youtube.com/embed/videoId"></iframe>`.

---

## Part 2: CSS (15 Questions)

### Selectors

**1. What are CSS selectors?**

CSS selectors are used to select the HTML elements you want to style, such as element (`p`), class (`.className`), and ID (`#id`).

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

A class selector applies to multiple elements, while an ID selector applies to only one unique element.

**3. How do you select all `<p>` elements inside a `<div>`?**

Use the descendant selector: `div p {}` to select all `<p>` elements within a `<div>`.

## Box Model

4. **What is the CSS box model?**

The box model describes the layout of elements, including the content, padding, border, and margin.

5. **How do padding and margin differ in the box model?**

Padding is the space between the content and the element's border, while margin is the space outside the border between the element and other elements.

6. **How do you set the width and height of an element including its padding and border?**

Use `box-sizing: border-box;` to include padding and border in the element's total width and height.

## Positioning and Layout

7. **What is the `position` property in CSS?**

The `position` property defines how an element is positioned on the page. Values include `static`, `relative`, `absolute`, and `fixed`.

8. **What is the difference between `absolute` and `relative` positioning?**

`Absolute` positions an element relative to its nearest positioned ancestor, while `relative` positions an element relative to its normal position.

9. **What is Flexbox in CSS?**

Flexbox is a layout model that makes it easier to align and distribute space among items in a container using properties like `display: flex`, `justify-content`, and `align-items`.

## Responsive Design

10. **What are media queries in CSS?**

Media queries allow you to apply CSS rules based on device characteristics like screen width, using `@media` rules, e.g., `@media (max-width: 600px) {}`.

11. **What is the difference between `min-width` and `max-width` in media queries?**

`min-width` applies styles when the viewport width is greater than the specified value, while `max-width` applies styles when the width is less than the specified value.

12. **How do you create a responsive grid layout?**

Use CSS Grid or Flexbox along with media queries to adjust the layout for different screen sizes.

## Styling

13. **How do you change the background color of an element?**

Use the `background-color` property, e.g., `body { background-color: lightblue; }`.

14. **What is the `color` property in CSS?**

The `color` property sets the color of the text, e.g., `p { color: red; }`.

15. **How do you change the font size of an element?**

Use the `font-size` property, e.g., `h1 { font-size: 32px; }`.

---

## Part 3: JavaScript (25 Questions)

### DOM Manipulation

1. **What is the DOM in JavaScript?**

The DOM (Document Object Model) is a representation of the HTML structure as objects, allowing JavaScript to interact with and manipulate the content and layout.

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

Use `document.getElementById('id')` to select an element by its ID.

3. **How do you add a class to an HTML element using JavaScript?**

Use `element.classList.add('class-name')` to add a class to an element.

4. **How do you remove an element from the DOM in JavaScript?**

Use `element.remove()` to remove an element from the DOM.

5. **How do you change the text content of an element in JavaScript?**

Use `element.textContent = 'New text'` to change the text of an element.

## Control Flow

6. **What is a `for` loop in JavaScript?**

A `for` loop is used to execute a block of code a certain number of times. Example: `for (let i = 0; i < 5; i++) { console.log(i); }`.

7. **What is an `if-else` statement in JavaScript?**

An `if-else` statement executes a block of code if a condition is true, otherwise it runs the code in the `else` block.

8. **What is the purpose of the `switch` statement in JavaScript?**

A `switch` statement allows you to execute different blocks of code based on the value of a variable.

9. **What is a `while` loop in JavaScript?**

A `while` loop repeatedly executes a block of code as long as a specified condition is true.

10. **How do you exit a loop in JavaScript?**

Use the `break` statement to exit a loop prematurely.

## ES6 Features

11. **What are arrow functions in JavaScript?**

Arrow functions are a shorthand syntax for writing functions in JavaScript. Example: `const add = (a, b) => a + b;`

12. **What is destructuring in JavaScript?**

Destructuring allows you to extract values from arrays or objects into variables. Example: `const [a, b] = [1, 2];`

13. **What is the difference between `let` and `const` in JavaScript?**

`let` allows you to reassign variables, while `const` creates read-only constants that cannot be reassigned.

14. **What is template literals in JavaScript?**

Template literals are string literals that allow embedded expressions, using backticks (``) and `${}` for placeholders.

**15. What are default parameters in JavaScript?**

Default parameters allow you to initialize function parameters with default values if no value is passed.

**16. What is the `spread` operator in JavaScript?**

The `spread` operator (`...`) allows an iterable to expand in places where multiple arguments are expected.

**17. What is the `rest` parameter in JavaScript?**

The `rest` parameter (`...args`) allows you to pass an indefinite number of arguments to a function.

**18. What are promises in JavaScript?**

Promises represent asynchronous operations that either resolve or reject. Example: `new Promise((resolve, reject) => {})`.

## APIs

**19. What is an API?**

An API (Application Programming Interface) allows applications to communicate with each other, providing data and services.

**20. What is the Fetch API in JavaScript?**

The Fetch API is used to make network requests to retrieve resources from a server. Example: `fetch('https://api.example.com/data').then(response => response.json())`.

**21. How do you handle errors in Fetch API requests?**

Use `.catch()` to handle errors in a Fetch request, e.g., `fetch(url).catch(error => console.error('Error:', error))`.

**22. What is JSON and how is it used with APIs?**

JSON (JavaScript Object Notation) is a lightweight format for data interchange, commonly used to send and receive data in API requests.

**23. What is the difference between `GET` and `POST` requests in APIs?**

`GET` requests retrieve data, while `POST` requests send data to the server.

**24. How do you send data in a POST request using Fetch API?**

Use the `fetch()` function with the method set to `POST` and include data in the `body`.

Example:

```
fetch(url, {  
  
  method: 'POST',  
  headers: {  
    'Content-Type': 'application/json'  
  },  
  body: JSON.stringify({ key: 'value' })  
});
```

**25. What is CORS and how does it relate to APIs?**

CORS (Cross-Origin Resource Sharing) is a security feature that restricts how resources on a web page can be requested from another domain.