HTML Interview Questions

1. Semantic Tags

Q1. What are semantic tags in HTML?
A1. These are tags that clearly tell what their content is about. Example: `<header>` means a header, `<footer>` means a footer.

Q2. Why are semantic tags important?
A2. They make code easier to read, help search engines, and help screen readers for blind users.

Q3. Give examples of semantic and non-semantic tags.
A3. Semantic: `<nav>`, `<article>`, `<section>`
Non-semantic: `<div>`, `<span>`

Q4. Difference between `<section>` and `<article>`?
A4. `<section>` is for grouping related content. `<article>` is for a full, stand-alone piece like a blog post.

2. Attributes

Q5. What are attributes in HTML?
A5. They give extra details to a tag. Example: `<img src="photo.jpg" alt="My Photo">`.

Q6. Difference between global and specific attributes?
A6. Global works with any tag (`id`, `class`). Specific works only with certain tags (`src` for `<img>`).

Q7. What does the `alt` attribute do in `<img>`?
A7. Shows text if the image doesn't load and helps blind users understand the image.

Q8. What is `data-*` used for?
A8. Stores custom information in HTML that can be used in JavaScript. Example: `<div data-id="5"></div>`.

3. HTML Elements

Q9. Difference between block and inline elements?
A9. Block starts on a new line (`<div>`, `<p>`). Inline stays in the same line (`<span>`, `<a>`).

Q10. Difference between `<div>` and `<span>`?
A10. `<div>` is block-level (big container). `<span>` is inline (small text or part of a sentence).

Q11. Difference between HTML tag and HTML element?
A11. Tag: `<p>` or `</p>`. Element: `<p>Hello</p>` (tag + content).

Q12. Difference between empty and non-empty elements?
A12. Empty has no closing tag or content (`<br>`). Non-empty has both (`<p>Text</p>`).

4. Forms and Inputs

Q13. What is the purpose of HTML forms?
A13. To collect user input, like login details or feedback.

Q14. Name different `<input>` types.
A14. `text`, `password`, `email`, `number`, `checkbox`, `radio`, `file`, `submit`.

Q15. Difference between `GET` and `POST`?
A15. `GET` shows data in the URL. `POST` hides it in the background.

Q16. What does `required` do in inputs?
A16. Makes the field must be filled before submitting.

5. Media

Q17. How do you add an image in HTML?
A17. Use `<img src="image.jpg" alt="Text">`.

Q18. How to add audio in HTML?
A18.

```
<audio controls>
  <source src="song.mp3" type="audio/mpeg">
</audio>
```

`controls` shows play buttons.

Q19. How to add video in HTML?
A19.

```
<video controls>
  <source src="video.mp4" type="video/mp4">
</video>
```

Q20. Difference between `<object>`, `<embed>`, and `<iframe>`?
A20.

- `<object>`: Adds files like PDF or image.

- `<embed>`: Adds media like videos.

- `<iframe>`: Shows another web page inside the page.

CSS Interview Questions

1. Selectors

Q1. What are CSS selectors?
A1. They are patterns used to select HTML elements you want to style. Example: `p {color: red;}` changes all `<p>` text to red.

Q2. What is the difference between class and ID selectors?
A2. Class uses a dot (`.myClass`) and can be used on many elements. ID uses a hash (`#myId`) and should be used on only one element.

Q3. What are pseudo-classes in CSS?
A3. They style elements in a special state. Example: `a:hover` changes a link's style when the mouse is over it.

2. Box Model

Q4. What is the CSS box model?
A4. Every element is a box made of 4 parts: content, padding, border, and margin.

Q5. Difference between margin and padding?
A5. Padding is space inside the border (around the content). Margin is space outside the border (between elements).

Q6. How do you set all margins or paddings at once?
A6. Use shorthand: `margin: 10px;` (all sides same) or `margin: 10px 20px;` (top/bottom, left/right).

3. Positioning and Layout

Q7. What are the different CSS position values?
A7. `static` (default), `relative` (moves from normal spot), `absolute` (positioned relative to nearest positioned parent), `fixed` (stays in place), `sticky` (sticks when scrolling).

Q8. Difference between relative and absolute positioning?
A8. `relative` moves the element from its normal place. `absolute` positions it based on the nearest positioned parent, not its normal place.

Q9. What is Flexbox used for?
A9. Flexbox is used to arrange elements in a row or column easily, and to align them nicely.

4. Responsive Design

Q10. What is responsive design?
A10. It makes websites look good on all devices (mobile, tablet, desktop).

Q11. How do you make a website responsive?
A11. Use flexible layouts, media queries, and percentage-based widths.

Q12. What is a media query in CSS?
A12. It applies styles only when certain conditions are true. Example:

```
@media (max-width: 600px) {
  body { background: yellow; }
}
```

5. Styling

Q13. How do you change the text color and background color in CSS?
A13.

```
p { color: blue; background-color: yellow; }
```

Q14. How to make text bold, italic, or underlined?
A14.

- Bold: `font-weight: bold;`
- Italic: `font-style: italic;`
- Underline: `text-decoration: underline;`

Q15. How to add a border around an element?
A15.

```
div { border: 2px solid black; }
```

JavaScript Interview Questions

1. DOM Manipulation

Q1. What is the DOM?
A1. DOM (Document Object Model) is a way for JavaScript to see and change HTML and CSS of a webpage.

Q2. How do you select an element by ID in JavaScript?
A2. Use `document.getElementById("idName")`.

Q3. How do you change the text inside an HTML element?
A3.

```
document.getElementById("myId").innerText = "New text";
```

Q4. How do you change the style of an element?
A4.

```
document.getElementById("myId").style.color = "red";
```

Q5. How do you create and add a new element to the page?
A5.

```
let p = document.createElement("p");
p.innerText = "Hello!";
document.body.appendChild(p);
```

2. Control Flow

Q6. What is control flow in JavaScript?
A6. It's the order in which code runs, including loops, if/else, and function calls.

Q7. How do you write an if-else statement?
A7.

```
if (x > 10) {
  console.log("Big");
} else {
  console.log("Small");
}
```

Q8. What is a `for` loop?
A8. A loop that repeats code a set number of times.

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

Q9. What's the difference between `break` and `continue`?
A9. `break` stops the loop completely; `continue` skips the current loop step and moves to the next one.

Q10. How do you write a `switch` statement?
A10.

```
switch (color) {
  case "red": console.log("Stop"); break;
  case "green": console.log("Go"); break;
  default: console.log("Unknown");
}
```

3. ES6 Features

Q11. What is `let` and `const`?
A11. `let` is for variables that can change. `const` is for variables that can't change.

Q12. What are arrow functions?
A12. A shorter way to write functions.

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

Q13. What is template literal syntax?
A13. Using backticks `` ` `` to make strings with variables.

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

Q14. What is object destructuring?
A14. A way to get values from objects easily.

```
let {name, age} = person;
```

Q15. What is array destructuring?
A15. Getting values from arrays easily.

```
let [a, b] = [1, 2];
```

Q16. What is the spread operator (`...`)?
A16. It copies or combines arrays/objects.

```
let arr2 = [...arr1];
```

Q17. What is the difference between default and named exports?
A17. Default export has one main thing per file, named export can have many.

```
export default myFunc;
export {a, b};
```

Q18. What are promises in JavaScript?
A18. A way to handle async tasks.

```
new Promise((resolve, reject) => {...});
```

4. APIs

Q19. What is an API?
A19. An API is a way for different software to talk to each other.

Q20. What is the Fetch API?
A20. A built-in function in JS to get data from a server.

```
fetch("url").then(res => res.json());
```

Q21. How do you handle API errors with fetch?
A21. Use `.catch()` after `.then()` to catch errors.

Q22. What is the difference between synchronous and asynchronous code?
A22. Synchronous runs line by line. Asynchronous can run in the background without stopping other code.

Q23. What is JSON?
A23. A text format for storing and sending data. Example:

```
{"name":"John", "age":25}
```

Q24. How do you parse JSON in JavaScript?
A24.

```
let obj = JSON.parse('{"name":"John"}');
```

Q25. How do you turn an object into JSON in JavaScript?
A25.

```
let str = JSON.stringify({name: "John"});
```