

HTML Interview Questions and Answers

1. What is a `<div>` tag in HTML?

The `<div>` tag is a **block-level element** that occupies the full width of the container and always starts on a new line. It is mainly used to create sections or divisions in a webpage and to apply CSS styles or JavaScript functionalities to them individually.

```
<div>Content here</div>
```

2. What are void tags in HTML?

Void tags (also called **self-closing or unpaired tags**) are tags that do not require a closing tag. Examples include:

- `
` (line break)
- `<hr>` (horizontal rule)
- `` (image)

These tags are complete in themselves.

3. Explain the `thead`, `tbody`, and `tfoot` tags.

These tags are parts of a table structure:

- `<thead>`: Defines the **header** section of the table.
- `<tbody>`: Contains the **main data** of the table.
- `<tfoot>`: Contains the **footer** section.

They help in organizing the table and improving **code readability** and **browser rendering**.

4. What is the difference between `id` and `class`?

In HTML, the `id` attribute is used to uniquely identify a single element. An `id` must be unique within a page, and one element can have only one `id`. It is commonly used for styling or targeting elements in JavaScript, and is referenced in CSS using a `#` symbol.

The `class` attribute, on the other hand, is used to group multiple elements under the same name. It can be reused across many elements and is referenced in CSS using a `.` (dot).

IDs have higher specificity than classes in CSS.

- `id`: Unique selector. Only one element should have a specific `id`. Used in CSS with `#`.

```
<div id="header"></div>
```

- `class`: Can be reused on multiple elements. Used in CSS with `.`.

```
<div class="section"></div>
```

Note: `id` has higher specificity than `class`.

5. What is the purpose of the `<label>` tag?

The `<label>` tag is used to associate a **description with a form input**. It improves accessibility and usability by linking text to input elements.

```
<label for="username">Username:</label>
<input type="text" id="username">
```

6. What are self-closing tags?

Self-closing tags are also called **void tags or unpaired tags**. They do not have a closing tag. Examples:

- `` (image)
 - `
` (line break)
 - `<hr>` (horizontal rule)
 - `<input>` (form input)
-

7. What is the difference between `` and `` ?

Both tags make the text bold, but:

- `` : Purely **visual**, no semantic meaning.
- `` : Adds **semantic importance** and is understood better by screen readers and search engines.

Note: In HTML5, `` is preferred.

8. Explain `<fieldset>` and `<legend>` tags.

- `<fieldset>` : Groups related elements in a form with a visible border. It is always used inside the form tag.
- `<legend>` : Provides a **caption or title** for the fieldset. It is always inside the `<fielsset></fielsset>` tag

```
<form>
  <fieldset>
    <legend>Personal Info</legend>
    <input type="text" placeholder="Name">
  </fieldset>
</form>
```

9. What is the `<meta>` tag used for?

The `<meta>` tag provides **metadata** about the HTML document like:

- Character set
- Page description
- Keywords
- Author

Example:

```
<meta charset="UTF-8">
<meta name="description" content="Free Web Tutorials">
```

10. Difference between `` and `<i>` tags?

- Both `` and `<i>` tags make the text appear italic, but they have different purposes.
- The `<i>` tag is used only for visual styling with no special meaning.
- The `` tag stands for emphasis and adds semantic meaning, helping screen readers and search engines understand that the text is important.
- `` : Emphasizes the text **semantically**, useful for screen readers.

Example:

```
<em>Important word</em>  
<i>Just styled text</i>
```

11. What is the purpose of the `alt` attribute in ``?

The `alt` attribute provides **alternate text** for an image when it can't be loaded (due to missing file or slow network). It improves **accessibility** and **SEO**.

Example:

```

```

12. Inline, Block and Inline-block Elements

◊ **Inline Elements**

- Do **not start on a new line**
- Take **only as much width as needed**
- Cannot set `width` and `height`
- Examples: ``, `<a>`, ``, ``

◊ **Block Elements**

- Always **start on a new line**
- Take **full available width**
- You can apply `width`, `height`, `margin`, `padding`
- Examples: `<div>`, `<p>`, `<section>`, `<form>`

◊ **Inline-Block Elements**

- Behaves like **inline** (stays in the same line)

- But allows `width` and `height` like block
- Used when you want elements side-by-side but styled like blocks (e.g. cards, buttons)
- Set using:

```
display: inline-block;
```

13. What is the difference between `span` and `div` tag?

- The `` tag is an inline element, which means it only takes as much width as needed and stays on the same line.
- In contrast, the `<div>` tag is a block-level element, which takes the full width and always starts from a new line.
- We use `<div>` to style or group large sections of a webpage, and `` when we need to style small parts inside a line, like a word or phrase.

14. What is semantic tag?

- Semantic tags in HTML are meaningful tags that clearly describe the purpose of the content inside them. These tags help both browsers and developers understand the structure of the webpage. Most semantic tags are block-level elements. Examples include:

```
`<nav>` - for navigation links  
`<header>` - for page or section headings  
`<main>` - for the main content  
`<aside>` - for side content like ads  
`<section>`, `<article>`, and `<footer>` - for content sections, articles, and the page footer respectively
```

15. What is the purpose of the viewport `<meta>` tag?

- The viewport meta tag tells the browser how to control the page's dimensions and scaling on different devices. It helps make web pages responsive.

Example:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- It ensures the content fits properly on mobile, tablet, and desktop screens.

16. Difference between ``, ``, and `<dl>` tags in HTML

- `` (Unordered List)**

It is used to show items without any particular order.

- Default bullet type is `disc`.
- `type` attribute values: `disc`, `circle`, `square` and `diamond`.
- Example:

```
<ul type="square">  
  <li>HTML</li>  
  <li>CSS</li>  
</ul>
```

- **** (Ordered List)

It is used to display items in a specific order.

- Default numbering is 1.
- type attribute values: 1, A, a, I, i.
- Example:

```
<ol type="A">
  <li>Install VS Code</li>
  <li>Write HTML</li>
</ol>
```

- **<dl>** (Description List)

It is used to define terms and their definitions.

- <dt> : Data Term (acts as a heading)
- <dd> : Data Description (definition of the term)
- Example:

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

17. What is the difference between **<link>** and **<a>** tags in HTML?

- The **<link>** tag is used inside the **<head>** section of an HTML document to connect external resources to the webpage. It is commonly used to link external CSS files, favicons, fonts, etc.

Example:

```
<link rel="stylesheet" href="styles.css">
```

- The `<a>` tag stands for anchor tag and is used to create **clickable hyperlinks** that navigate users to another page, section, file, or URL. It is used inside the `<body>` tag.

Example:

```
<a href="https://example.com">Visit Example</a>
```

- Key Difference:**

- `<link>` is for connecting **external files/resources** (invisible).
- `<a>` is for creating **visible clickable links** on the webpage.

18. What is the difference between `<section>` and `<div>` tags in HTML?

The `<div>` tag is a non-semantic container used to group elements and apply CSS styling or JavaScript functionality. It doesn't provide any meaning about the content inside.

The `<section>` tag is a semantic HTML5 element used to define a standalone section of content that has a heading and represents a specific purpose on the page. It gives meaning to the browser and helps with accessibility and SEO.

- Use `<div>` when you just need a container with no semantic meaning.
- Use `<section>` when the content inside forms a standalone, meaningful block (like a blog post, services, or about section).

Example:

```
<!-- Non-semantic container -->
<div class="card">
  <h2>Title</h2>
  <p>Some content</p>
</div>

<!-- Semantic section -->
<section>
  <h2>Our Services</h2>
  <p>We offer web development and design services.</p>
</section>
```

19. What is the difference between `<article>` and `<section>` tags in HTML?

Both `<article>` and `<section>` are **semantic tags** in HTML, meaning they give meaning to the content inside them.

- `<article>` is used for **independent, self-contained content** that can make sense on its own — like a blog post, news article, forum post, or product card. It should be reusable and understandable without needing the rest of the page.
- `<section>` is used to **group related content together** under a common heading. It helps structure the page into logical parts like "About Us", "Contact", "Features", etc.

Example:

```
<section>
  <h2>Our Services</h2>
  <article>
    <h3>Web Development</h3>
    <p>We build websites using modern technologies.</p>
  </article>
  <article>
    <h3>App Development</h3>
    <p>We create cross-platform mobile apps.</p>
  </article>
</section>
```

20. What is the difference between `<header>`, `<head>`, and `<h1>` to `<h6>` tags in HTML?

This is a common confusion for beginners. Let's clarify all three:

◊ `<head>` Tag:

- Appears **only once** in the HTML document.
- Placed inside the `<html>` tag, before `<body>`.
- Contains **meta-information** (not displayed on the page).
- Used for:
 - Defining page title
 - Adding metadata
 - Linking CSS or JS files

Example:

```
<head>
  <title>My Web Page</title>
  <meta charset="UTF-8" />
  <link rel="stylesheet" href="style.css" />
  <script src="script.js"></script>
</head>
```

◊ `<header>` Tag:

- A **semantic** element that defines a **visible heading section** of a page or section.
- Can appear **multiple times** (once per page, per section, etc.).
- Used for:
 - Logo

- Site or section heading
- Navigation bar

Example:

```
<header>
  <h1>Welcome to My Blog</h1>
  <nav>
    <a href="/">Home</a>
    <a href="/about">About</a>
  </nav>
</header>
```

◇ `<h1>` to `<h6>` Tags:

- Represent **HTML headings**.
- `<h1>` is the **most important (largest)** heading.
- `<h6>` is the **least important (smallest)** heading.
- They define **content structure**, not just styling.
- Used **inside** `<body>`, often placed inside `<header>`, `<section>`, etc.

Example:

```
<h1>Main Title</h1>
<h2>Sub Title</h2>
<h3>Section Heading</h3>
<h4>Sub-section</h4>
<h5>Minor heading</h5>
<h6>Tiny note</h6>
```

✓ Summary Comparison Table:

Tag	Purpose	Placement	Visibility	Count	Common Use
-----	---------	-----------	------------	-------	------------

Tag	Purpose	Placement	Visibility	Count	Common Use
<code><head></code>	Metadata (title, meta, CSS/Javascript links)	Inside <code><html></code>	<input checked="" type="checkbox"/> Hidden	Only once	SEO, linking files, page settings
<code><header></code>	Page/section header (logo, nav, heading)	Inside <code><body></code>	<input checked="" type="checkbox"/> Visible	Multiple	Top part of page or section
<code><h1> – <h6></code>	Content headings (h1 = main, h6 = smallest)	Inside <code><body></code>	<input checked="" type="checkbox"/> Visible	Multiple	Structuring content hierarchy

21. What is the difference between `<nav>` and `<footer>` tag in HTML5?

- The `<nav>` tag is used to define a **navigation section** in a webpage. It usually contains links to important sections like **Home, About, Services, Contact**, etc. These links help users navigate through different parts of the website.
- The `<footer>` tag is used to define the **bottom section** of a webpage. It typically contains content like **copyright information, privacy policies, contact details, or social media links**.

Both are semantic tags, meaning they give meaning to the structure of the HTML page and help search engines and screen readers understand the layout better.

22. What is the difference between `<script>` and `<noscript>` tags in HTML?

Understanding these two tags is important for handling both JavaScript-enabled and disabled browsers.

<script> Tag:

- Used to **embed or reference JavaScript code** in HTML.
- Can include **inline JavaScript** or use an external `.js` file via the `src` attribute.
- Runs **only if JavaScript is enabled** in the browser.

Examples:

```
<!-- Inline JavaScript -->
<script>
  console.log("This is inline JavaScript");
</script>

<!-- External JavaScript -->
<script src="main.js"></script>
```

<noscript> Tag:

- Acts as a **fallback** for browsers that do **not support JavaScript** or have it **disabled**.
- Any content inside this tag is shown **only if JavaScript is turned off**.
- Can be used to alert users or provide basic alternative functionality.

Example:

```
<noscript>
  <p>Please enable JavaScript to use this site.</p>
</noscript>
```

Summary Table:

Tag	Purpose	Executes When	Visibility
<code><script></code>	Runs JavaScript code	When JavaScript is enabled	<input checked="" type="checkbox"/> Visible (JS output)
<code><noscript></code>	Shows fallback content	When JavaScript is disabled	<input checked="" type="checkbox"/> Visible (in fallback)

Key Difference:

- `<script>` is for executing JavaScript.
- `<noscript>` is for displaying content **only when JavaScript is not available.**

23. What is the use of the `<title>` tag in HTML?

The `<title>` tag serves two important purposes in HTML:

◊ 1. Inside `<head>` Tag (Most Common Use):

- The `<title>` tag is placed **inside the `<head>` section** of the HTML document.
- It sets the **title of the web page**, which:
 - Appears on the **browser tab**.
 - Is shown as the title when the page is **bookmarked**.
 - Helps with **SEO (Search Engine Optimization)** by telling search engines what the page is about.

Example:

```
<head>
  <title>My Portfolio Website</title>
</head>
```

This will show **My Portfolio Website** as the name on the browser tab.

2. Inside Elements like `` or `<a>` (Tooltip Behavior):

- When used inside tags like ``, `<a>`, etc., the **title attribute** (not the `<title>` tag) works differently.
- It shows a **tooltip** when the user hovers over the element.

Example:

```

```

Here, **"Company Logo"** will be shown as a tooltip when the user hovers over the image.

 Note: The `<title>` tag (used in `<head>`) and the **title attribute** (used in tags like ``) are different.

Summary Table:

Use Case	Tag / Attribute	Purpose	Where Used
Browser tab title	<code><title></code>	Displays page title on tab	Inside <code><head></code> tag
Tooltip on hover	<code>title</code> attribute	Shows extra info on hover	Inside tags like <code></code> or <code><a></code>

Conclusion:

- 
- Use `<title>` inside `<head>` for the page title.
 - Use **title** as an **attribute** for tooltips on elements.

24. What is the difference between `<meta charset="UTF-8">` and `<meta name="viewport" content="width=device-width, initial-scale=1.0">` in HTML?

- `<meta charset="UTF-8">`

It defines the character encoding used in the HTML document.

`UTF-8` is the most common character set, which supports almost all characters from all languages.

It ensures the browser displays special symbols and characters correctly.

- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

It controls the layout on mobile browsers.

It sets the width of the webpage to match the screen-width of the device and ensures the content scales properly.

This tag is **essential for responsive web design**.

`charset` = character encoding

`viewport` = screen scaling and width control on different devices

25. What is the use of the `<base>` tag in HTML?

The `<base>` tag is used to set a base URL or default target for all relative links (`<a>`, ``, etc.) in the HTML document.

- It must be placed **inside the `<head>`** tag.
- Only **one `<base>` tag** is allowed in a document.

Syntax:

```
<base href="https://www.example.com/" target="_blank">
```

If you use

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

then all relative links like

```
<a href="contact.html">  
    will point to  
    https://www.example.com/contact.html  
    by default.  
* It helps in managing URLs more efficiently in large projects.
```

26. What is the difference between `defer` and `async` attributes in the `<script>` tag?

When we use the `defer` attribute in the `<script>` tag, the browser loads the JavaScript file in the background while parsing the HTML. Once the HTML is completely parsed, the script runs **in the order they appear** in the document.

On the other hand, when we use the `async` attribute, the script is also fetched in parallel, but it gets executed **as soon as it's downloaded**, which means the output **order is not guaranteed** and can break dependencies if multiple scripts rely on each other.

☒ Key Differences:

Feature	<code>defer</code>	<code>async</code>
Loading	Parallel with HTML	Parallel with HTML
Execution	After HTML is parsed	As soon as script is ready
Execution Order	Maintains order	Does not maintain order
Use Case	Best for scripts that need the DOM	Best for independent, small scripts