1. **HTML &CSS Basics**
2. **What is HTML, and why is it important for website design?**

**Ans** - **HTML** (HyperText Markup Language) is the foundational language used to create and structure content on the web. It defines the elements of a webpage—such as headings, paragraphs, images, links, and forms—using a system of tags and attributes.

1. **How do you structure a basic HTML document?**

**Ans** –

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Your Page Title</title>

</head>

<body>

<h1>Welcome to My Website</h1>

<p>This is a simple HTML document.</p>

</body>

</html>

**Breakdown of Each Section**

1. **<!DOCTYPE html>**  
   Declares the document type and version of HTML being used (HTML5 in this case). This helps browsers render the page correctly.
2. **<html lang="en">**  
   The root element that wraps all the content of the document. The lang attribute specifies the language of the document, which is important for accessibility and search engine optimization.
3. **<head>**  
   Contains meta-information about the document, such as:
   * **<meta charset="UTF-8">**: Specifies the character encoding for the document, ensuring proper display of text.
   * **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Makes the page responsive on different devices.
   * **<title>Your Page Title</title>**: Sets the title of the webpage, which appears in the browser tab.
4. **<body>**  
   Contains all the visible content of the webpage:
   * **<h1>**: Defines the main heading of the page.
   * **<p>**: Defines a paragraph of text.
5. **What are semantic HTML elements, and why should they be used?**

**Ans** - Semantic HTML elements are tags that convey meaningful information about the content they enclose, both to browsers and developers.

* **<header>**: Represents introductory content or navigational links.
* **<nav>**: Defines a block of navigation links.
* **<main>**: Specifies the main content of a document.
* **<section>**: Represents a standalone section of content.
* **<article>**: Defines independent, self-contained content.
* **<aside>**: Represents content tangentially related to the content around it.
* **<footer>**: Defines the footer for a document or section.

1. **Explain the difference between inline, block, and inline-block elements in HTML.**

**Ans** –

**Block Elements**

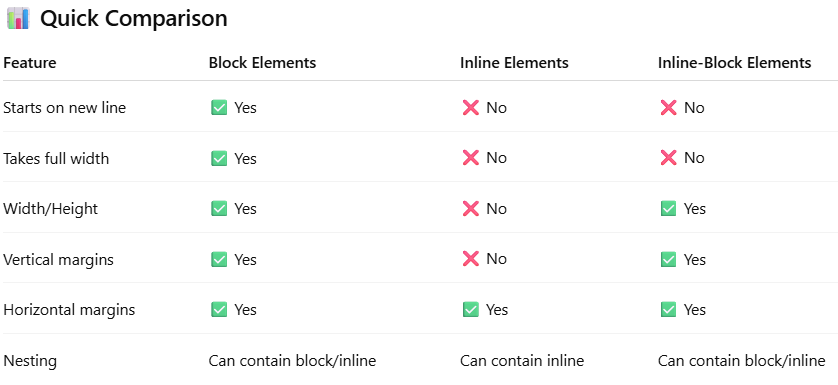
* **Display Behavior**: Block elements take up the full width of their parent container, stretching from left to right. They always start on a new line, causing subsequent elements to appear below them.[Pluralsight+10Reddit+10Html Reader+10](https://www.reddit.com/r/Frontend/comments/1166hyv?utm_source=chatgpt.com)
* **Width & Height**: You can set both width and height properties.[DEV Community](https://dev.to/vayolapradeep/difference-between-css-display-inline-block-inline-block-3cpk?utm_source=chatgpt.com)
* **Examples**: <div>, <p>, <header>, <footer>, <section>, <article>, <h1>–<h6>, <ul>, <ol>, <li>, <form>, <table>, <nav>.
* **Use Case**: Ideal for structural elements like containers, sections, and paragraphs.

**Inline Elements**

* **Display Behavior**: Inline elements only occupy as much width as their content requires and do not start on a new line. They flow within the content, allowing other inline elements to be placed beside them.[MDN Web Docs+11DEV Community+11Html Reader+11](https://dev.to/vayolapradeep/difference-between-css-display-inline-block-inline-block-3cpk?utm_source=chatgpt.com)
* **Width & Height**: You cannot set width or height properties.[DEV Community+1Pluralsight+1](https://dev.to/vayolapradeep/difference-between-css-display-inline-block-inline-block-3cpk?utm_source=chatgpt.com)
* **Examples**: <span>, <a>, <strong>, <em>, <img>, <code>, <button>.[CodePen](https://codepen.io/CodeStitch/pen/nVpmwa?utm_source=chatgpt.com" \t "_blank)
* **Use Case**: Suitable for styling small portions of content within block elements, such as links, emphasized text, or inline images.

**Inline-Block Elements**

* **Display Behavior**: Inline-block elements are similar to inline elements in that they do not start on a new line. However, unlike inline elements, they allow you to set width and height properties.[FreeCodeCamp+4Html Reader+4Poynter Andrew+4](https://htmlreader.com/html/html_blocks.html?utm_source=chatgpt.com)
* **Width & Height**: Both width and height properties can be set, providing more control over the element's size.
* **Examples**: Any element can be converted to inline-block by setting display: inline-block; in CSS. Commonly used for creating horizontal navigation menus, image galleries, and grid layouts.[W3Schools](https://www.w3schools.com/css/css_inline-block.asp?utm_source=chatgpt.com)
* **Use Case**: Useful when you want elements to flow inline but also need to control their dimensions and spacing



1. **How do you link a CSS stylesheet to an HTML document?**

**Ans -**  To link an external CSS file to an HTML document, use the <link> tag within the <head> section: This approach allows you to separate content (HTML) from presentation (CSS), promoting cleaner code and easier maintenance.

<head>

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

</head>

1. **Describe the CSS box model. What are the properties associated with it?**

**Ans -**  The CSS box model defines the rectangular boxes generated for elements in the document tree. It consists of:

* **Content**: The actual content of the box (e.g., text, images).
* **Padding**: Clears an area around the content, inside the border.
* **Border**: A border surrounding the padding (if specified).
* **Margin**: Clears an area outside the border, separating the element from others.

These properties determine the layout and spacing of elements on a page.

1. **How would you center a div horizontally and vertically on a page?**

**Ans -**  You can center a <div> both horizontally and vertically using Flexbox:

.container {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

1. **What’s the difference between margin and padding?**

* **Ans -**  **Margin**: The space outside an element's border, creating distance between it and other elements.
* **Padding**: The space inside an element's border, between the border and the content.

In essence, margin separates elements, while padding creates space within an element.

1. **Explain the difference between class and id selectors in CSS. When would you use each?**

**Ans -**  **Class (.)**: Used to select multiple elements sharing the same style. Ideal for styling groups of elements.

**ID (#)**: Used to select a single, unique element. Suitable for elements that appear once per page.

1. **What is the purpose of media queries, and how do they contribute to responsive design?**

**Ans -**  Media queries allow CSS to apply styles based on device characteristics like screen width, height, orientation, and resolution. They enable responsive design by adapting layouts to different screen sizes, ensuring optimal viewing experiences across devices.

@media (max-width: 768px) {

.container {

flex-direction: column;

}

}

**2.4 How do flexbox and CSS grid help with page layout? When would you use each?**

**Ans -**