

--- DISPLAY PROPERTY ---

The display property in CSS defines how an element is displayed on the web page. It plays a crucial role in layout and determines whether an element behaves as a block, inline, or other types.

1. block

- The element is displayed as a block-level element.
- Starts on a new line and takes up the full width available.
- Examples: <div>, <p>, <section>.

```
div {  
  
    display: block;  
}
```

2. inline

- The element is displayed as an inline element.
- Does not start on a new line and only takes up as much width as necessary.
- Examples: , <a>, .

```
span {  
    display: inline;  
}
```

3. inline-block

- Behaves like an inline element but can have block-level properties (e.g., width and height).
- Does not start on a new line.

```
button {  
    display: inline-block;  
}
```

4. flex

- The element becomes a flex container, enabling Flexbox layout for its children.
- Allows for alignment, spacing, and ordering within the container.

```
.container {  
  display: flex;  
}
```

5. inline-flex

- Behaves like flex, but the container itself is inline.

```
.inline-container {  
  display: inline-flex;  
}
```

6. grid

- The element becomes a grid container, enabling CSS Grid layout for its children.

```
.grid-container {  
  display: grid;  
}
```

7. inline-grid

- Behaves like grid, but the container itself is inline.

```
.inline-grid-container {  
  
  display: inline-grid;  
}
```

8. table

- Behaves like a <table> element, adopting table layout rules.

```
.table {  
  display: table;  
}
```

9. **table-row, table-cell, table-caption, etc.**

- These mimic the behavior of `<tr>`, `<td>`, `<caption>`, etc., in a table.

```
.table-row {  
  display: table-row;  
}
```

10. **none**

- The element is not displayed and does not occupy any space in the layout.

```
.hidden {  
  display: none;  
}
```

- **contents**

- The element's children are displayed, but the element itself does not generate a box.
- Useful for wrapping elements without impacting layout.

```
.wrapper {  
  display: contents;  
}
```

- **list-item**

- Makes an element behave like a list item, including a bullet or number if in a `` or ``.

```
li {  
  display: list-item;  
}
```

Block-Level Elements

- Block-level elements:

- Start on a new line by default.
- Take up the full width available, stretching from left to right.
- Can contain other block or inline elements.

- Structural Elements:
 - `<div>`
 - `<section>`
 - `<article>`
 - `<aside>`
 - `<header>`
 - `<footer>`
 - `<main>`
 - `<nav>`

- Text Elements:
 - `<p>` (paragraph)
 - `<h1>` to `<h6>` (headings)
 - `<pre>` (preformatted text)
 - `<blockquote>` (quoted block)

- Form Elements:
 - `<form>`
 - `<fieldset>`
 - `<legend>`

- Media and Layout:
 - `<figure>`
 - `<figcaption>`
 - `<address>`
 - `<hr>` (horizontal rule)

Inline-Level Elements

- Inline-level elements:
 - Do not start on a new line.
 - Only take up as much width as necessary.
 - Can only contain other inline elements or text (not block elements).
- Text Formatting:
 - `` (generic inline container)
 - `` (bold)
 - `` (strong emphasis)
 - `<i>` (italic)
 - `` (emphasis)
 - `<mark>` (highlighted text)
 - `<small>` (smaller text)
 - `<sub>` (subscript)
 - `<sup>` (superscript)
 - `<u>` (underlined text)
 - `<s>` (strikethrough)
- Links and Interactions:
 - `<a>` (anchor/link)
 - `<button>` (inline buttons in specific contexts)
- Media and Forms:
 - `` (image)
 - `<input>` (input field)
 - `<label>` (form label)
 - `<select>` (dropdown)
 - `<textarea>` (multiline text input)

Default display Property

Block-Level Elements (Default display: block)

- `<div>`, `<p>`, `<h1>` to `<h6>`, `<section>`, `<article>`, `<nav>`, `<header>`, `<footer>`, `<blockquote>`, `<figure>`, `<main>`, `<form>`, `<fieldset>`, `<legend>`, `<pre>`, `<address>`, `<hr>`

Inline-Level Elements (Default display: inline)

- ``, `<a>`, ``, `<i>`, ``, ``, `<mark>`, `<small>`, `<sub>`, `<sup>`, `<u>`, `<s>`, `<label>`, ``, `<input>`, `<button>`, `<textarea>`, `<select>`