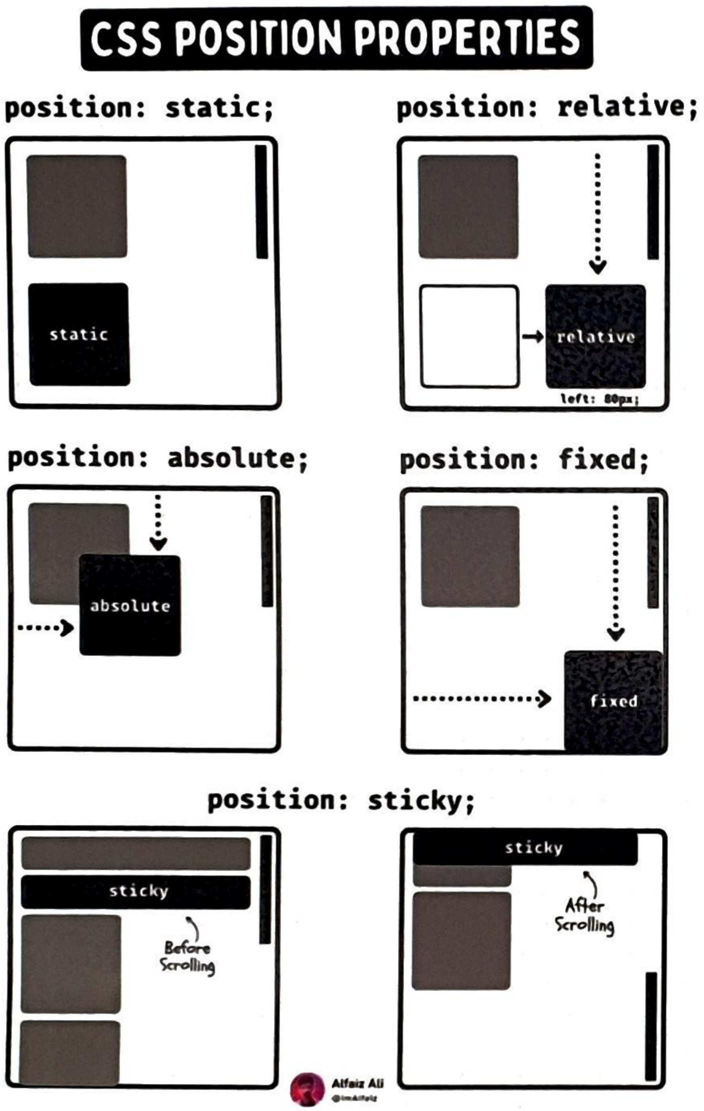
POSITION properties



{ position: static; }

HTML elements are positioned static by default. Static-positioned elements are unaffected by the top, bottom, left, and right properties.

{ position: relative; }

The element is positioned according to the document's normal flow, and then offset relative to itself based on the values of top, right, bottom, and left.

{ position: absolute; }

The absolute element is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed). if an absolute positioned element has no positioned ancestors, it uses the document body and moves along with page scrolling.

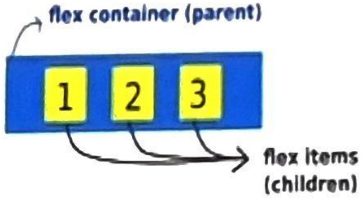
{ position: fixed; }

The element is removed from the normal document flow, and no space is created for the element in the page layout. Its final position is determined by the values of the top, right, bottom, and left.

{ position: sticky; }

The sticky element is positioned based on the user's scroll position. Depending on the scroll position, a sticky element toggles between relative and fixed.

Flexbox Cheat Sheet



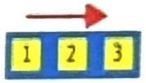
Parent properties

display: enables flex context for all direct children.

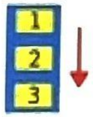
```
.container{
  display: flex;  // or inline-flex
}
```

flex-direction: sets the main-axis.

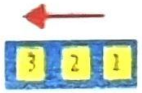
row



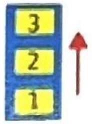
column



row-reverse

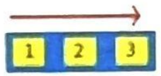


column-reverse

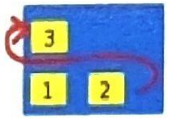


flex-wrap: allows the items to wrap as needed.

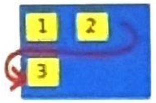
no-wrap



wrap-reverse

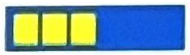


wrap

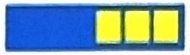


justify-content: defines alignment along the main axis.

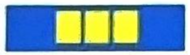
flex-start



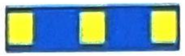
flex-end



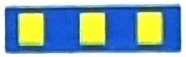
center



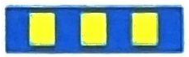
space-between



space-around

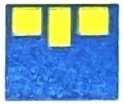


space-evenly

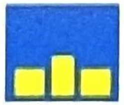


align-items: defines alignment along the cross axis.

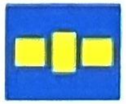
flex-start



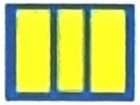
flex-end



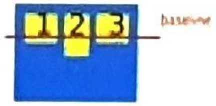
center



stretch

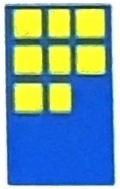


baseline

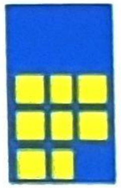


align-content: aligns multiple lines, like justify-content does with individual items.

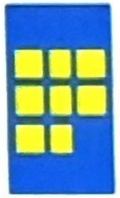
flex-start



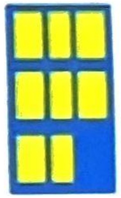
flex-end



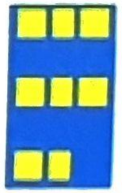
center



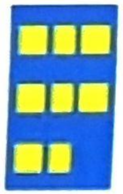
stretch



space-between

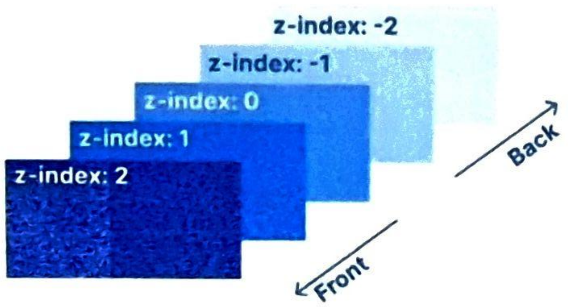


space-around



Z-INDEX proporties

Z-Index



AI Prompt Example

"Adjust the CSS code below to ensure the navigation bar always stays in front."

D-libro

The **`z-index`** property in CSS determines the **stack order of positioned elements**. An element with a higher `z-index` will appear in front of an element with a lower `z-index`.

Key points to remember:

- It only works on **positioned elements** (elements with `position` set to `absolute`, `relative`, `fixed`, or `sticky`).
- Elements with a higher `z-index` value are stacked on top of elements with a lower value.
- If `z-index` is not specified, elements are typically stacked in the order they appear in the HTML.