Position property:

The CSS position property is used to define how an element is positioned in the document. It controls the element’s placement, often in combination with top, right, bottom, and left properties.

Here are the common position values explained:

- static (default): The element follows the normal document flow. The top, bottom, left, and right properties have no effect on it. Elements appear in the order they are written.
- relative: The element is positioned relative to its normal position in the flow. The top, bottom, left, and right properties shift it from that original position, but the space for the element remains as if it were in the normal flow.
- absolute: The element is removed from the normal document flow and positioned relative to its nearest positioned ancestor (non-static). It does not affect or get affected by other elements' layout.
- fixed: The element is positioned relative to the browser viewport and stays fixed in place even when scrolling. It is also removed from the normal document flow.
- sticky: A hybrid between relative and fixed. The element acts relative until the scroll reaches a specified offset, after which it “sticks” and behaves like fixed. It is positioned relative until triggered to fix.

Comparison Table

Position	Description	Affects Layout	Relative to	Offset properties work?
static	Default document flow	Yes	Normal flow	No
relative	Positioned relative to itself	Yes	Original position	Yes
absolute	Removed from flow, positioned	No	Nearest positioned ancestor	Yes
fixed	Removed from flow, viewport fixed	No	Viewport	Yes
sticky	Relative until scroll threshold, then fixed	Yes	Nearest scrolling ancestor	Yes

This table captures the main functional differences to decide which to use based on layout and interaction needs.The differences between the five CSS position values are:

- static: Default position; element follows normal document flow. Offset properties (top, left, etc.) do not apply. It affects the layout normally.
- relative: Positioned relative to its normal position, allowing offset without changing other elements' layout. The space it originally occupies remains.

- absolute: Removed from normal flow and positioned relative to the nearest positioned ancestor. It does not affect the layout of other elements.
- fixed: Similar to absolute but positioned relative to the viewport. It stays fixed when scrolling and doesn't affect the layout.
- sticky: Behaves like relative until scrolling reaches a threshold, then acts like fixed, sticking in place. It affects layout until it sticks.

Z-Index

The CSS `z-index` property controls the stacking order of overlapping elements on a web page. It determines which element appears *in front* or *behind* others when they overlap.

Key Points about z-index

- Works only on elements that have a position value other than `static` (e.g., `relative`, `absolute`, `fixed`, or `sticky`).
- Higher z-index values place elements in front of elements with lower values.
- Negative values are allowed, pushing elements behind.
- The default is `auto`, meaning stacking follows the element order in the HTML.

When to Use z-index

- To control the visual layering when elements overlap (e.g., modals over content).
- To keep background elements behind content using negative z-index.
- To manage dropdowns, tooltips, pop-ups, or decorative layers.

Syntax

```
css

selector {
  position: relative | absolute | fixed | sticky;
  z-index: auto | <integer> | initial | inherit;
}
```