

CSS COMBINATORS

Combinators are symbols used to define relationships between different elements in a selector. They allow you to target specific elements based on their position or relationship to other elements in the HTML structure.

There are four different combinators in CSS:

- Descendant combinator (space)
- Child combinator (>)
- Next sibling combinator (+)
- Subsequent-sibling combinator (~)

Descendant Combinator:

The descendant combinator matches all elements that are descendants of a specified element.

- The following elements select all <p> elements inside <div> elements

```
EX:    div p {  
        background-color: Green;  
    }
```

Child Combinators:

The child combinator selects all elements that are the children of a specified element.

- The following example selects all <p> elements that are children of a <div> element:

```
EX: div > p {  
    background-color: orange;  
}
```

Next Sibling Combinators:

The next sibling combinator is used to select an element that is directly after another specific element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

- The following example selects the first <p> element that are placed immediately after <div> elements:

```
EX: div + p {  
    background-color: white;  
}
```

Subsequent-sibling combinator :

The subsequent-sibling combinator selects all elements that are next siblings of a specified element.

The following example selects all <p> elements that are next siblings of <div> elements:

EX:

```
div ~ p {  
    background-color: yellow;  
}
```

CSS Positions

The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky).

There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

Position :Static;

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This <div> element has position: static;

EX:

```
div.static {  
  position: static;  
  border: 3px solid green;  
}
```

Position: Relative;

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

EX:

```
div.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;  
  border: 3px solid red;  
}
```

Position: Absolute;

An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like `fixed`).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

EX:

```
div.relative {  
  position: relative;  
  width: 400px;  
  height: 200px;  
  border: 3px solid #73AD21;  
}
```

```
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```

Position: Sticky;

An element with `position: sticky;` is positioned based on the user's scroll position.

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset

position is met in the viewport - then it "sticks" in place (like position:fixed).

EX:

```
div.sticky {  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}
```