

# What is Pseudo Class?

A Pseudo class is used to define a special state of an element, it will give a extra attention to the users. This changes the background color of a button only when the mouse hovers over it.

## Uses of Pseudo Class:

- Style an element when a user moves the mouse over it
- Style visited and unvisited links differently
- Style an element when it gets focus
- Style valid/invalid/required/optional form elements
- Style an element that is the first child of its parent

## Syntax

selector:pseudo-class-name

```
{  
  CSS properties  
}
```

## CSS Pseudo-Class

Hover - when mouse is over the element

active - when element is clicked

focus - when input is focused

visited - visited link color

first-child - first element of the parent

last-child - last element of the parent

nth-child(n)- selects specific child

## Supported Browsers:

- Google Chrome
- Edge
- Firefox
- Opera
- Safari

## Pseudo-element

**::after** - Inserts something after the content of the specified element

**::backdrop** - Inserts something after the content of the specified element

**::before** - Inserts something before the content of the specified element

**::file-selector-button** - Selects any button of type <input type="file">

**::first-letter** - Selects the first letter of every <p> element

**::first-line** - Selects the first line of every <p> element

**::selection** - Styles the user-selected text

Aspect	Pseudo-Classes	Pseudo-Elements
Purpose	Select elements based on their state or user interaction, like :hover or :active.	Style specific parts of an element or create virtual elements like ::before or ::first-line.
Syntax	Uses a single colon (:), e.g.,a:hover	Uses double colons (::), e.g.,p::first-letter
Functionality	Targets an entire element when it meets a certain condition or state.	Targets a specific part of an element or generates content that doesn't exist in the DOM.
Usage	Can be used with multiple selectors and combined with other selectors.	Typically used alone and only one can appear per selector.
Examples	a:hover,input:focus	p::first-line,div::before,span::after

**Positioning** in CSS is the process of controlling how elements are placed and layered on a web page by using the position property and related offset properties like top, right, bottom, and left. Types of Positioning Static: Default position. Elements flow naturally in the document order. Offset properties (top, left, etc.) have no effect. Relative: Element is placed relative to its original position. Offset properties move the element, but its space in the document remains reserved.

## Types of Positioning

- Static: Default position. Elements flow naturally in the document order. Offset properties (top, left, etc.) have no effect.
- Relative: Element is placed relative to its original position. Offset properties move the element, but its space in the document remains reserved.
- Absolute: Element is completely removed from the normal flow and positioned relative to the nearest non-static ancestor. Can overlap other elements and does not reserve space in the document flow.
- Fixed: Element is positioned relative to the browser window (viewport). It stays in the same place even when the page is scrolled. Removed from normal flow.
- Sticky: Combines relative and fixed. The element acts as relative until it crosses a specific threshold when scrolling, then it becomes fixed at that position.

## Z index:

Z-index property controls the stacking order of elements. Elements with a higher z-index appear on top of elements with a lower z-index. Only works on positioned elements: relative, absolute, fixed, or sticky.

### For example:

We have 3 images to display and the z index controls the order of the image

```
Img1{  
  Position: absolute;  
  right: 10px;  
  z-index: 1}  
  
Img2{  
  Position: absolute;  
  right: 10px;  
  z-index: 2}  
  
Img3{  
  Position: absolute;  
  right: 10px;  
  z-index: 3}
```

img3 is displayed first because it has the higher z index value.