Day 07

Pseudo Class and Elements

CSS Learning series

CSS Pseudo-classes

A **pseudo-class** is used to define the **special state of an element,** like when it's **hovered, focused, selected,** or even its **position among siblings.**

A pseudo-class is used to define a special state of an element.

It can be used to:

- 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

Syntax :

```
selector:pseudo-class {
    property: value;
}
```

Pseudo-Class	Description	Example
`:hover`	Styles element when hovered with a mouse.	`a:hover { color: red; }`
`:focus`	When element (input, button) gets focus.	`input:focus { border: blue; }`
`:active`	While element is being clicked.	`button:active { scale: 1.1; }`
`:visited`	Links that have already been visited.	`a:visited { color: purple; }`
`:first-child`	Matches the first child of a parent.	`li:first-child`
`:last-child`	Matches the last child of a parent.	`li:last-child`
`:nth-child(n)`	Matches the nth child of a parent.	`li:nth-child(2)`
`:nth-of-type(n)`	Matches the nth element of a specific type.	`p:nth-of-type(2)`
`:not(selector)`	Excludes elements that match the selector.	`div:not(.box)`
`:checked`	Targets selected radio/checkboxes.	`input:checked { ... }`
`:disabled`	Matches disabled form fields.	`button:disabled { ... }`

↓

CSS Pseudo-elements

A pseudo-element allows you to style specific parts of an element that you normally can't select with regular selectors (like the first line of text or adding content before or after elements).

A CSS pseudo-element is used to style specific parts of an element.

It can be used to:

- Style the first letter or line, of an element
- Insert content before or after an element
- Style the markers of list items
- Style the viewbox behind a dialog box

Syntax :

```
selector::pseudo-element {
  property: value;
}
```

Pseudo-Element	Description	Example
`::before`	Inserts content **before** an element's content.	`h1::before { content: "* "; }`
`::after`	Inserts content **after** an element's content.	`p::after { content: "✅"; }`
`::first-letter`	Styles the **first letter** of a block.	`p::first-letter { font-size: 200%; }`
`::first-line`	Styles the **first line** of a block.	`p::first-line { color: red; }`
`::selection`	Styles the text selected by the user.	`::selection { background: yellow; }`
`::placeholder`	Styles placeholder text in inputs.	`input::placeholder { color: gray; }`
`::marker`	Styles the marker (e.g., bullet) of list items.	`li::marker { color: red; }`
`::file-selector-button`	Styles the upload button in the input.	`input[type="file"]::file-selector-button { ... }`

Pseudo-Class vs Pseudo-Element

Feature	Pseudo-Class (`:`)	Pseudo-Element (`::`)
Targets	**State** of element	**Part** of element
Syntax	`:hover` , `:focus` , etc.	`::before` , `::after` , etc.
Affects	Whole element	Specific content or part
Use Cases	Interaction, structure, condition	Inject content, highlight pieces

Day 01

Introduction to HTML

HTML Learning series

Check Out

Day 01

Introduction of CSS

CSS Learning series

Previous post

@rutuja.codes
Rutuja | Web designer

Thank you for learning with me!

❤️ **Drop a like**

✈️ **Share it with friends**

💬 **Leave a Comment**

🔖 **Save this post**

</> | @rutuja.codes
Rutuja | Web designer