Lesson:

Advanced Selector in CSS (Pseudo Element and Pseudo Class)

PW SKILLS

Topics Covered

- Introduction to Pseudo Selectors
- Define Pseudo Classes
- List of some of the common Pseudo-class selectors with example
- Define Pseudo element
- List of some of the common Pseudo-element selectors with example

Introduction to Pseudo Selectors

Pseudo-selectors are a type of selector in CSS (Cascading Style Sheets) that allow you to target and style specific elements based on their state or position in the HTML document, without using additional classes or IDs. They are denoted with a colon (":") followed by a keyword and are used to apply styles to elements that meet specific conditions or criteria.

Pseudo-selectors provide a powerful way to target and style elements in specific ways without modifying the HTML structure or adding additional classes or IDs. They are widely used in CSS to create dynamic and interactive web pages with custom styles and behaviours

Pseudo-selectors can be of two types: pseudo-classes and pseudo-elements.

Define Pseudo Classes

Pseudo-classes are predefined keywords used to select an element based on its state or target a specific child. they start with a single colon (:). They can be used as part of a selector, and they are very useful to style active or visited links for example, change the style on hover, focus, target the first child, or odd rows which is very handy in many cases.

For example, it can be used to:

- Element styling when a user mouses over it
- Styling of visited and visited links differently
- Styling of elements when focused.

List of some of the common Pseudo-class selectors with some example

Pseudo class name	CSS	Description
hover selector	button:hover	Select elements that are hovered by the mouse.
focus selector	button:focus	Select elements that are focused by either tabbing or clicking on the element.

required selector	input:required	Select elements that are required.
checked selector	input:checked	Select checkboxes/radio buttons that are checked
disabled selector	input:disabled	Select inputs that are disabled
first Child selector	a:first-child	Select elements that are the first child inside a container
last Child selector	a:last-child	Select elements that are the last child inside a container
nth Child selector	a:nth-child(2n)	Select elements that are the nth child inside a container based on the formula.
empty selector	p:empty	Select element that do not contain any child elements or text content
active selector	a:active	selects an element when it is being clicked or pressed down by the mouse pointer.
visited selector	a:visited	Selects a link that has been visited
link selector	a:link	selects an unvisited link.
:not()	root	Selects the document's root element
enabled	input: enabled	selects a form element that is enabled.
first-of-type	P:first-of-type	Selects every element that is the first element of its parent
in-range	input:in-range	Selects element with a value within a specified range
Required selector	input:required	Selement elements with a required attribute specified
valid selector	input: valid	Select all elements with a valid value

target selector	a:target	Select the current active element
visited selector	a:visited	Selects all visited links
invalid:selector	input:invalid	Selects all elements with an invalid value
last-of-type selector	p:last-of-type	Selects elements that is last element of its parent
not selector	:not(p)	Select every element that is not a <p> element
Optional selector	input:optional	Select <input> elements with no “required” attribute
nth-last-child selector	p:nth-last-child(2)	Select every <p> element that is the second child of its parent, counting from the last child
nth-last-of-type selector	p: nth-last-type(2)	Selects every <p> element that is the second child of its parent, counting from the last child
nth-of-type selector	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
only-of-type selector	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
only-child selector	p:only-child	Selects every <p> element that is the only child of its parent

Example of Pseudo class selectors

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example of Semantic HTML Tags</title>
    <style>
      /* hover */
      h1:hover {
        color: red;
```

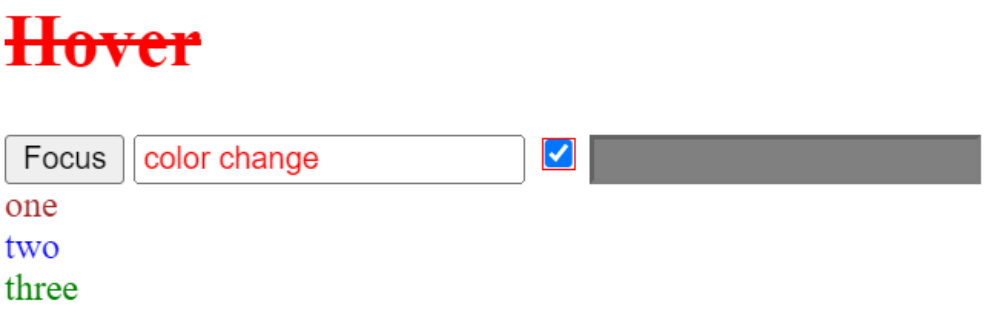
```
text-decoration: line-through;
      }
      /* focus*/
      button:focus {
        color: red;
      }

      /* required */
      input:required {
        color: red;
      }
      /* checked */
      input:checked {
        border: none;
        outline: 1px solid red;
      }
      /* disable */
      input:disabled {
        background-color: grey;
      }
      /*child selector*/
      div:first-child {
        color: brown;
      }

      div:last-child {
        color: green;
      }
      div:nth-child(2n) {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Hover</h1>
    <button>Focus</button>
```

```
<input type="text" required="true" />
    <input type="checkbox" />
    <input type="text" disabled="true" />
    <div>
      <div>one</div>
      <div>two</div>
      <div>three</div>
    </div>
  </body>
</html>
```

Browser output -



Define Pseudo element

Pseudo-element selectors are used to style a specific part of an element. They start with a double colon (::).

Examples like ::before,::after are probably the most used pseudo-elements. They are used to add content before or after an element.

List of some of the common Pseudo-element selectors with example

Pseudo Element name	CSS	Description
after	tag::after	Insert content after the content of an element
before	tag::before	Insert content before the content of an element
first-letter	tag::first-letter	Selects the forest letter of each element
first-line	tag::first-line	Select the first line of each element
selection	tag::selection	Matches the portion of an element that is selected by a user

Example of Pseudo element Selectors

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example of Semantic HTML Tags</title>
    <!— <link rel="stylesheet" href="style.css" /> —>
    <style>
      /* before and after selector - insert content to
before and after */
      h1::after {
        content: "After!";
        color: red;
      }
      h1::before {
        content: "Before!";
        color: green;
      }

      /* first-letter selector - select the first letter
and applied the css properties  */
      h2::first-letter {
        color: red;
        font-size: 40px;
      }

      /* first-line - apply the css properties to the first
line */
      p::first-line {
        background-color: yellow;
        text-decoration: underline;
      }
      /* selection - apply the css properties to the select
element */
```

```
 p::selection {
        background-color: yellow;
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>Pseudo before and after selector</h1>
    <h2>First letter selector</h2>
    <p>
      Lorem, ipsum dolor sit amet consectetur adipisicing
elit. Sequi
      consequuntur error ratione molestiae dignissimos
earum quisquam
      aspernatur, repudiandae dicta sapiente.
    </p>
  </body>
</html>
```

Browser output –