

CSS Pseudo-classes Review

User Action Pseudo-classes

- **Pseudo-classes Definition:** These are special CSS keywords that allow you to select an element based on its specific state or position.
- **User Action Pseudo-classes:** These are special keywords that allow you to change the appearance of elements based on user interactions, improving the overall user experience.
- **:active Pseudo-class:** This pseudo-class lets you select the active state of an element, like clicking on a button.
- **:hover Pseudo-class:** This pseudo-class defines the hover state of an element.
- **:focus Pseudo-class:** This pseudo-class applies styles when an element gains focus, typically through keyboard navigation or when a user clicks into a form input.
- **:focus-within Pseudo-class:** This pseudo-class is used to apply styles to an element when it or any of its descendants have focus.

Input Pseudo-classes

- **Input Pseudo-classes:** These pseudo-classes are used to target HTML input elements based on the state they are in before and after user interaction.
- **:enabled Pseudo-class:** This pseudo-class is used to target form buttons or other elements that are currently enabled.
- **:disabled Pseudo-class:** This pseudo-class lets you style an interactive element in disabled mode.
- **:checked Pseudo-class:** This pseudo-class is used to indicate to the user that it is checked.
- **:valid Pseudo-class:** This pseudo-class targets the input fields that meet the validation criteria.
- **:invalid Pseudo-class:** This pseudo-class targets the input fields that do not meet the validation criteria.

- **:in-range and :out-of-range Pseudo-classes:** These pseudo-classes apply styles to elements based on whether their values are within or outside specified range constraints.
- **:required Pseudo-class:** This pseudo-class targets input elements that have the required attribute. It signals to the user that they must fill out the field to submit the form.
- **:optional Pseudo-class:** This pseudo-class applies styles input elements that are not required and can be left empty.
- **:autofill Pseudo-class:** This pseudo-class applies styles to input fields that the browser automatically fills with saved data.

Location Pseudo-classes

- **Location Pseudo-classes:** These pseudo-classes are used for styling links and elements that are targeted within the current document.
- **:any-link Pseudo-class:** This pseudo-class is a combination of the :link and :visited pseudo-classes. So, it matches any anchor element with an href attribute, regardless of whether it's visited or not.
- **:link Pseudo-class:** This pseudo-class allows you to target all unvisited links on a webpage. You can use it to style links differently before the user clicks on them.
- **:local-link Pseudo-class:** This pseudo-class targets links that point to the same document. It can be useful when you want to differentiate internal links from external ones.
- **:visited Pseudo-class:** This pseudo-class targets a link the user has visited.
- **:target Pseudo-class:** This pseudo-class is used to apply styles to an element that is the target of a URL fragment.
- **:target-within Pseudo-class:** This pseudo-class applies styles to an element when it or one of its descendants is the target of a URL fragment.

Tree-structural Pseudo-classes

- **Tree-structural Pseudo-classes:** These pseudo-classes allow you to target and style elements based on their position within the document tree.

- **:root Pseudo-class:** This pseudo-class is usually the root html element. It helps you target the highest level in the document so you can apply a common style to the entire document.
- **:empty Pseudo-class:** Empty elements, that is, elements with no children other than white space, are also included in the document tree. That's why there's an :empty pseudo-class to target empty elements.
- **:nth-child(n) Pseudo-class:** This pseudo-class allows you to select elements based on their position within a parent.
- **:nth-last-child(n) Pseudo-class:** This pseudo-class enables you to select elements by counting from the end.
- **:first-child Pseudo-class:** This pseudo-class selects the first element in a parent element or the document.
- **:last-child Pseudo-class:** This pseudo-class selects the last element in a parent element or the document.
- **:only-child Pseudo-class:** This pseudo-class selects the only element in a parent element or the document.
- **:first-of-type Pseudo-class:** This pseudo-class selects the first occurrence of a specific element type within its parent.
- **:last-of-type Pseudo-class:** This pseudo-class selects the last occurrence of a specific element type within its parent.
- **:nth-of-type(n) Pseudo-class:** This pseudo-class allows you to select a specific element within its parent based on its position among siblings of the same type.
- **:only-of-type Pseudo-class:** This pseudo-class selects an element if it's the only one of its type within its parent.

Functional Pseudo-classes

- **Functional Pseudo-classes:** Functional pseudo-classes allow you to select elements based on more complex conditions or relationships. Unlike regular pseudo-classes which target elements based on a state (for example, :hover, :focus), functional pseudo-classes accept arguments.
- **:is() Pseudo-class:** This pseudo-class takes a list of selectors (ex. ol, ul) and selects an element that matches one of the selectors in the list.

▼ Example Code

```
<p class="example">This text will change color.</p>
<p>This text will not change color.</p>
<p>This text will not change color.</p>
<p class="this-works-too">This text will change color.</p>
```

▼ Example Code

```
p:is(.example, .this-works-too) {
  color: red;
}
```

- **:where()** Pseudo-class: This pseudo-class takes a list of selectors (ex. `ol`, `ul`) and selects an element that matches one of the selectors in the list. The difference between `:is` and `:where` is that the latter will have a specificity of 0.

▼ Example Code

```
:where(h1, h2, h3) {
  margin: 0;
  padding: 0;
}
```

- **:has()** Pseudo-class: This pseudo-class is often dubbed the "parent" selector because it allows you to style elements that contain child elements specified in the selector list.

▼ Example Code

```
article:has(h2) {
  border: 2px solid hotpink;
}
```

- **:not()** Pseudo-class: This pseudo-class is used to select elements that do not match the provided selector.

▼ Example Code

```
p:not(.example) {
  color: blue;
}
```

Pseudo-elements

- **::before Pseudo-element:** This pseudo-element uses the content property to insert cosmetic content like icons just before the element.
- **::after Pseudo-element:** This pseudo-element uses the content property to insert cosmetic content like icons just after the element.
- **::first-letter Pseudo-element:** This pseudo-element targets the first letter of an element's content, allowing you to style it.
- **::marker Pseudo-element:** This pseudo-element lets you select the marker (bullet or numbering) of list items for styling.