

# Pseudo-Classes

CSS pseudo-classes are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-classes is as follows –

- selector:pseudo-class {property: value}

CSS classes can also be used with pseudo-classes –

- selector.class:pseudo-class {property: value}

While defining pseudo-classes in a <style>...</style> block, following points should be noted –

- a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.
- a:active MUST come after a:hover in the CSS definition in order to be effective.
- Pseudo-class names are not case-sensitive.
- Pseudo-class are different from CSS classes but they can be combined.

- The **:visited** CSS pseudo-class represents links that the user has already visited. For privacy reasons, the styles that can be modified using this selector are very limited.

```
a:visited {  
  color: green;  
}
```

- The **:link** CSS pseudo-class represents an element that has not yet been visited. It matches every unvisited <a>, <area>, or <link> element that has an href attribute.

```
a:link {  
  color: red;  
}
```

- The **:focus** CSS pseudo-class represents an element (such as a form input) that has received focus. It is generally triggered when the user clicks or taps on an element or selects it with the keyboard's "tab" key.

```
input:focus {  
  color: red;  
}
```

- The **:active** CSS pseudo-class represents an element (such as a button) that is being activated by the user. When using a mouse, "activation" typically starts when the user presses down the primary mouse button.

```
a:active {  
  color: red;  
}
```

- The **:hover** CSS pseudo-class matches when the user interacts with an element with a pointing device, but does not necessarily activate it. It is generally triggered when the user hovers over an element with the cursor (mouse pointer).

```
a:hover {  
  color: orange;  
}
```

- The **:first-child** CSS pseudo-class represents the first element among a group of sibling elements.

```
p:first-child {  
  color: lime;  
}
```

- The **:first-of-type** CSS pseudo-class represents the first element of its type among a group of sibling elements.

```
p:first-of-type {  
  color: red;  
}
```

- The **:last-child** CSS pseudo-class represents the last element among a group of sibling elements.

```
p:last-child {  
  color: lime;  
}
```

- The **:last-of-type** CSS pseudo-class represents the last element of its type among a group of sibling elements.

```
p:last-of-type {  
  color: lime;  
}
```

- The **:nth-child()** CSS pseudo-class matches elements based on their position in a group of siblings.

```
:nth-child(4n) {
```

```
color: lime;
}
```

- The **:nth-last-child()** CSS pseudo-class matches elements based on their position among a group of siblings, counting from the end.

```
:nth-last-child(4n) {
  color: lime;
}
```

- The **:nth-last-of-type()** CSS pseudo-class matches elements of a given type, based on their position among a group of siblings, counting from the end.

```
p:nth-last-of-type(4n) {
  color: lime;
}
```

- The **:only-child** CSS pseudo-class represents an element without any siblings. This is the same as **:first-child:last-child** or **:nth-child(1):nth-last-child(1)**, but with a lower specificity.

```
p:only-child {
  background-color: lime;
}
```

- The **:only-of-type** CSS pseudo-class represents an element that has no siblings of the same type.

```
p:only-of-type {
  background-color: lime;
}
```

# Validation-

- The **:checked** CSS pseudo-class selector represents any **radio** (`<input type="radio">`), **checkbox** (`<input type="checkbox">`), or **option** (`<option>` in a `<select>`) element that is checked or toggled to an on state.

```
:checked {
  margin-left: 25px;
  border: 1px solid blue;
}
```

- The **:default** CSS pseudo-class selects form elements that are the default in a group of related elements.

It may match the `<button>`, `<input type="checkbox">`, `<input type="radio">`, and `<option>` elements:

- A default option element is the first one with the `selected` attribute, or the first enabled option in DOM order. multiple `<select>`s can have more than one selected option, so all will match `:default`.
- `<input type="checkbox">` and `<input type="radio">` match if they have the `checked` attribute.
- `<button>` matches if it is a `<form>`'s default submission button: the first `<button>` in DOM order that belongs to the form. (This also applies to `<input>` types that submit forms, like `image` or `submit`.)

- The `:disabled` CSS pseudo-class represents any disabled element. An element is disabled if it can't be activated (selected, clicked on, typed into, etc.) or accept focus. The element also has an enabled state, in which it can be activated or accept focus.

```
input:disabled {
  background: #ccc;
}
```

- The `:enabled` CSS pseudo-class represents any enabled element. An element is enabled if it can be activated (selected, clicked on, typed into, etc.) or accept focus. The element also has a disabled state, in which it can't be activated or accept focus.

```
input:enabled {
  color: blue;
}
```

- The `:empty` CSS pseudo-class represents any element that has no children. Children can be either element nodes or text (including whitespace). Comments, processing instructions, and CSS content do not affect whether an element is considered empty.

```
div:empty {
  background: lime;
}
```

- The `:in-range` CSS pseudo-class represents an `<input>` element whose current value is within the range limits specified by the `min` and `max` attributes.

```
input:in-range {
  background-color: rgba(0, 255, 0, 0.25);
}
```

- The `:out-of-range` CSS pseudo-class represents an `<input>` element whose current value is outside the range limits specified by the `min` and `max` attributes.

```
input:out-of-range {
  background-color: rgba(255, 0, 0, 0.25);
}
```

- The `:invalid` CSS pseudo-class represents any `<input>` or other `<form>` element whose contents fail to validate.

```
input:invalid {
```

```
background-color: pink;
}
```

- The **:valid** CSS pseudo-class represents any `<input>` or other `<form>` element whose contents validate successfully. This allows to easily make valid fields adopt an appearance that helps the user confirm that their data is formatted properly.

```
input:valid {
background-color: powderblue;
}
```

- The **:not()** CSS pseudo-class represents elements that do not match a list of selectors. Since it prevents specific items from being selected, it is known as the *negation pseudo-class*.

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

- The **:optional** CSS pseudo-class represents any `<input>`, `<select>`, or `<textarea>` element that does not have the required attribute set on it.

```
input:optional {
border: 1px dashed black;
}
```

- The **:required** CSS pseudo-class represents any `<input>`, `<select>`, or `<textarea>` element that has the required attribute set on it.

```
input:required {
border: 1px dashed red;
}
```

- The **:lang()** CSS pseudo-class matches elements based on the language they are determined to be in.

```
p:lang(en) {
quotes: '\201C' '\201D' '\2018' '\2019';
}
```