Theory: Pseudo-classes

15 minutes 0 / 5 problems solved

If you made it to this topic, you probably already know what a selector is. Every selector can have a pseudo-class attached to it. Pseudo-classes include elements that allow you to work with the page in a dynamic mode and, for example, respond to user's actions. **Pseudo-classes** control the dynamic state of the page elements. An example of such change can be an element changing color when you hover over it, or coloring a visited link differently. There are two types of pseudo-classes: those that determine the state of elements, and those related to the document tree.

§1. Determining the state of elements

Any type of pseudo-class can be written with the following syntax:

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

Pseudo-classes that determine the state of elements respond to the current state of the selector. Let's not forget that pseudo-classes cannot exist separately and must be attached to the selector whose state they describe.

Pseudo-class `:active` is used when the user activates a page element. It allows the page to react when the element is activated. Activation is the time between the user pressing and releasing the mouse button. The link is considered active after it has been clicked on, for example:

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

Pseudo-class `:focus` is used when an element gets a **focus**. It usually happens to the `input` element of the form when one clicks on the form:

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

Pseudo-class `:hover` is attached to the selector if you want some `HTML` page element to react when the mouse pointer is hovering over it:

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

Another frequently used pseudo-class is `:visited`. We use it to mark links that have already been visited by the user:

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

§2. Working with a DOM tree

Pseudo-classes that allow working with the `DOM` tree target the hierarchy of an `HTML` document. They work with elements depending on their order in the `HTML` code structure. The first class we'll look at is `:first-child`. Let's consider an example:

Current topic:

Pseudo-classes Stage 3

Topic depends on:

- HTML forms Stage 3
- Lists Stage 3
- Basic syntax Stage 2

Table of contents:

```
<ul>
  <li>One</li>
  <li>Two</li>
  <li>Three</li>
  <li>Four</li>
  <li>Five</li>
  <li>Six</li>
  <li>Seven</li>
  <li>Eight</li>

  <li>Nine</li>

  <li>Ten</li>

</ul>
```

```
li:first-child {
   background: green;
}
```

As you probably guessed, the first element of the list will be highlighted with a green background. The `:last-child` works the same way except that it highlights the last `li` element:

```
li:last-child {
   background: red;
}
```

We can also select the *n-th* element using the pseudo-class `:nth-child`:

```
li:nth-child(2) {
    background: yellow;
}
```

The child element that is selected is indicated in the brackets after `:nth-child`, like in the example above.

Besides specifying the element number, we can indicate whether the accessed number in the `:nth-child` is even or odd with the help of key words `odd` and `even`:

```
li:nth-child(odd) {
    color: blue;
}

li:nth-child(even) {
    color: orange;
}
```

We can also point to every *n-th* element, for example, every third, using the following syntax:

```
li:nth-child(3n) {
    border-style: solid;
}
```

§3. The use of pseudo-classes

Now try to save the following `HTML` and `CSS` code in `index.html` and `style.css` files, open them in your browser, and check how pseudo-classes work:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" href="style.css">
    <meta charset="utf-8">
    <title>Pseudo-classes with Cookies</title>
  </head>
  <body>
    <h1>Chocolate Chip Cookies Recipe</h1>

    <ul>


  <li>Whisk together <a href="https://en.wikipedia.org/wiki/Salt">salt</a> and butter.</li>

  <li>Whisk the egg, vanilla and the <a href="https://en.wikipedia.org/wiki/Sugar">sugars</a></li>

  <li>Sift in the flour and baking <a href="https://en.wikipedia.org/wiki/Sodium_carbonate">soda</a>.</li>

  <li>Fold in the chocolate chunks, then chill the dough for at least 30 minutes.</li>

  <li>Preheat oven to 350°F (180°C). Line a baking sheet with parchment paper.</li>

        <li>Scoop the dough with an ice-cream scoop onto a parchment paper-lined baking sheet</li>

        <li>Bake for 12-
15 minutes, or until the edges have started to barely brown.</li>

        <li>Cool completely before serving.</li>

        <li>Enjoy!</li>

      </ul>

    </body>

  </html>
```

```
body {
  text-align: justify;
  margin-left: 25%;
  margin-right: 25%;
  width: 50%;
}

a:visited {
  color: green;

}



li:nth-child(3n) {

  border: solid;

}



h1:hover {

   background-color: gold;

}
```

What we have covered here is not a complete list of pseudo-classes, so you might want to go over the full list by yourself.

§4. Conclusion

In this topic, we considered the concept of pseudo-classes, learned about their two types, and saw some examples of their usage. Now it's high time for putting that knowledge into practice.

Report a typo

59 users liked this theory. **1** didn't like it. **What about you?**

Start practicing

Comments (0) Hints (0) Useful links (0) Show discussion