Child Selectors

In this lesson, we meet a new type of selector, the child selectors.
Let's begin!

WE'LL COVER THE FOLLOWING ^

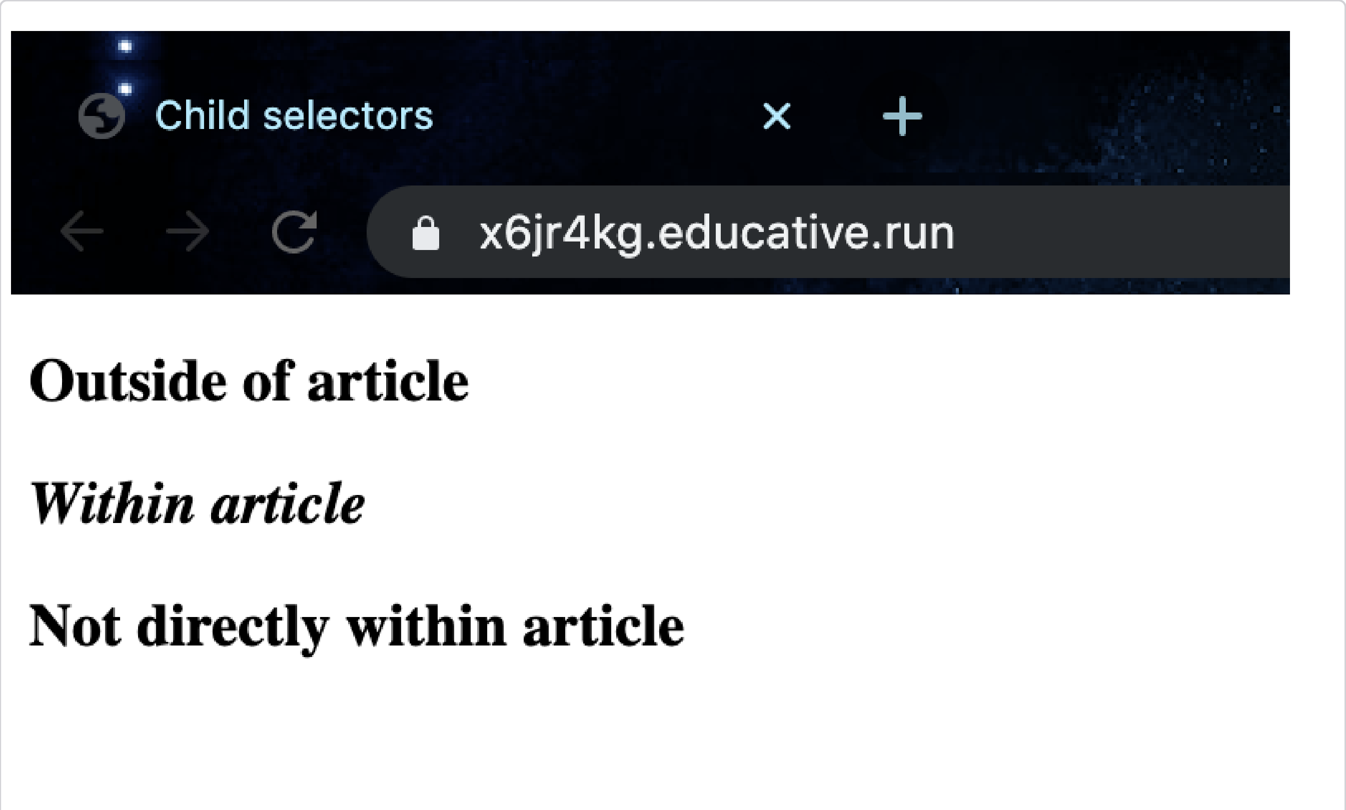
Similar to the descendent selector, CSS lets you style the children of another tag with a child selector. The child selector uses an angle bracket (`>`) to indicate the relationship between the two elements. While the descendent selector applies to all descendants of a tag (children, grandchildren, and so on), the child selector lets you specify which child of which parent you want to deal with.

For example, if you want to select the `<h2>` tags within an `<article>` tag, use the `article > h2` child selector, as demonstrated in Listing 9-8.

Listing 9-8: Styling the children of another tag with a child selector

```
<!DOCTYPE html>
<html>
<head>
  <title>Child selectors</title>
  <style>
    article > h2 {
      font-style: italic;
    }
  </style>
</head>
<body>
  <h2>Outside of article</h2>
  <article>
    <h2>Within article</h2>
    <div>
      <h2>Not directly within article</h2>
    </div>
  </article>
</body>
</html>
```

When you display this page (image below), only the second `<h2>` will be shown in italics, because only that `<h2>` tag matches the `article > h2` rule. The first `<h2>` is outside of `<article>`, and the third `<h2>` is nested in `<article>`, but it is not a direct child.



Outside of article

Within article

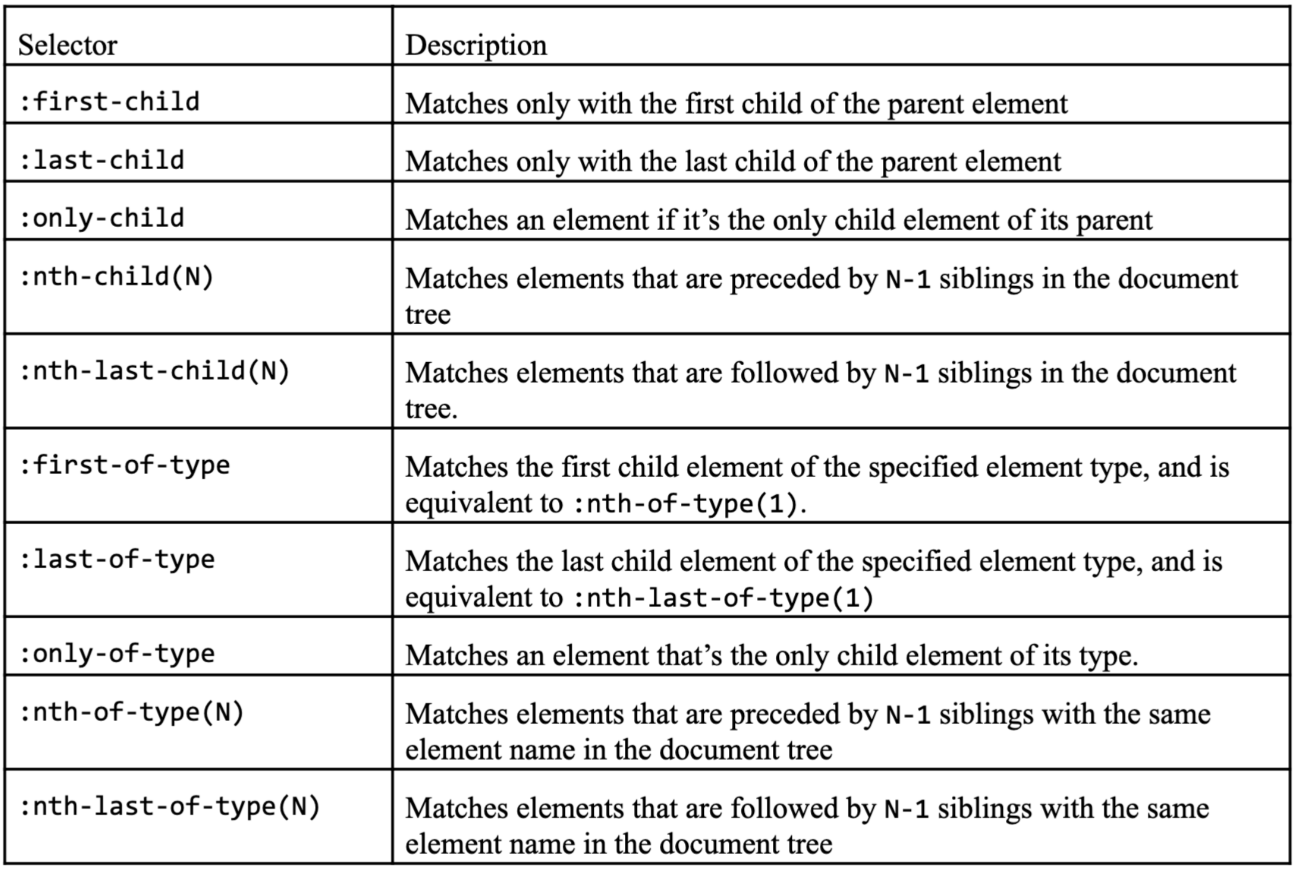
Not directly within article

Using the child selector

Very often, you need to select children of a certain HTML node by their **position** and **type**. You have already learned about pseudo-class selectors, but those were only a part of them.

CSS3 includes a few very specific pseudo-class selectors for selecting child elements, as summarized in the table below:

Pseudo-class selectors for child elements



Selector	Description
`:first-child`	Matches only with the first child of the parent element
`:last-child`	Matches only with the last child of the parent element
`:only-child`	Matches an element if it's the only child element of its parent
`:nth-child(N)`	Matches elements that are preceded by `N-1` siblings in the document tree
`:nth-last-child(N)`	Matches elements that are followed by `N-1` siblings in the document tree.
`:first-of-type`	Matches the first child element of the specified element type, and is equivalent to `:nth-of-type(1)`.
`:last-of-type`	Matches the last child element of the specified element type, and is equivalent to `:nth-last-of-type(1)`
`:only-of-type`	Matches an element that's the only child element of its type.
`:nth-of-type(N)`	Matches elements that are preceded by `N-1` siblings with the same element name in the document tree
`:nth-last-of-type(N)`	Matches elements that are followed by `N-1` siblings with the same element name in the document tree

As you can see, a number of pseudo-class selectors use an argument, `N`, which can be a keyword, a number, or can be given as `an+b`, where a and b are integers, for example (`2n+1`). Use the odd keyword for selecting odd-numbered elements, and `even` for selecting even-numbered elements.

If `N` is a number, it represents the ordinal position of the selected element. For example, 3 represents the third element. If `N` is given as `an+b`, `b` represents the ordinal position of the first element that we want to match, and `a` represents the ordinal number of every element we want to match after that.

So, the expression `3n+2` will match the second element, and every third element after that: the fifth, eighth, eleventh, and so on.
There is a difference between the `nth`- and `nth-last- pseudo-classes`. The `nth- pseudo-classes` count from the top of the document tree down, they select elements that have `N-1` siblings before them; meanwhile, the `nth-last-pseudo-classes` count from the bottom up, they select elements that have `N-1` siblings after them.

Listing 9-9 demonstrates these concepts.

Instead of providing a static web page, it is **dynamic**; *there's a text box where you can type in the child selector to apply*.

It uses JavaScript to *dynamically* add a style rule to the internal style sheet to mark the selected children with a `red` color and `italicized` font.

The items on which the demonstration is carried out, are nested into a `<div>` tag and are one of these types: `<p>`, `<span>`, or `<strong>`. The items within `<p>` tags are marked with a "(`p`)" suffix to help you identify how the selectors work.

Listing 9-9: Pseudo-class child selectors in action

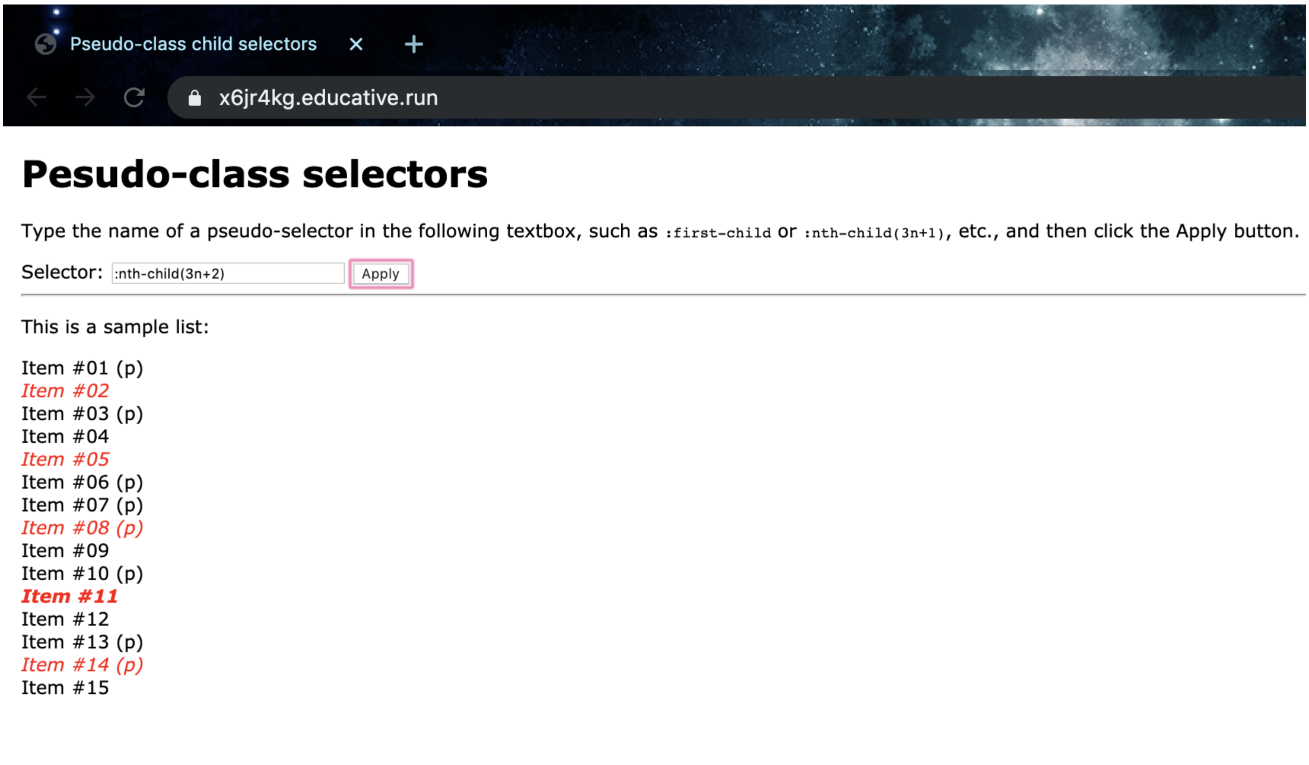
```
<!DOCTYPE html>
<html>
<head>
  <title>Pseudo-class child selectors</title>
  <style>
    body {
      font-family: Verdana, Helvetica, sans-serif;
      margin-left: 16px;
    }

    #selector {
      width: 200px;
    }

    div p {
      margin: 0;
    }

    div span, div strong {
      display: block;
    }
  </style>
  <style id="childStyle">
  </style>
  </style>
  <script>
    function applyStyle() {
      var child = document.getElementById('selector')
        .value;
      var rule = 'div ' + child +
        '{ color: red; font-style: italic }';
      var styleTag = document.getElementById('childStyle');
      styleTag.innerText = rule;
    }
  </script>
</head>
<body>
  <h1>Pesudo-class selectors</h1>
  <p>
    Type the name of a pseudo-selector in the
    following textbox, such as <code>:first-child</code>
    or <code>:nth-child(3n+1)</code>, etc., and
    then click the Apply button.
  </p>
  Selector:
  <input id="selector" type="text" autofocus />
  <button onclick="applyStyle()">Apply</button>
  <hr />
  <p>This is a sample list:</p>
  <div>
    <p>Item #01 (p)</p>
    <span>Item #02</span>
    <p>Item #03 (p)</p>
    <span>Item #04</span>
    <span>Item #05</span>
    <p>Item #06 (p)</p>
    <p>Item #07 (p)</p>
    <p>Item #08 (p)</p>
    <span>Item #09</span>
    <p>Item #10 (p)</p>
    <strong>Item #11</strong>
    <span>Item #12</span>
    <p>Item #13 (p)</p>
    <p>Item #14 (p)</p>
    <span>Item #15</span>
  </div>
</body>
</html>
```

Figure 9-9 shows what you see when you use the `:nth-child(3n+2)` selector.



Pesudo-class selectors

Type the name of a pseudo-selector in the following textbox, such as `:first-child` or `:nth-child(3n+1)`, etc., and then click the Apply button.

Selector: :nth-child(3n+2)

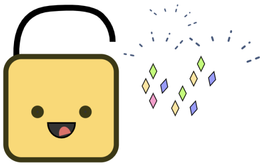
This is a sample list:

Item #01 (p)
Item #02
Item #03 (p)
Item #04
Item #05
Item #06 (p)
Item #07 (p)
Item #08 (p)
Item #09
Item #10 (p)
Item #11
Item #12
Item #13 (p)
Item #14 (p)
Item #15

Using the pseudo-class child selectors

Show Useful Info

Achievement unlocked!



Congratulations! You've learned to make use of child selectors in CSS!

Good job! Give yourself a round of applause!

Now that we have understood the usage and significance of child selectors, we will go onto see another type of selectors in the *next lesson*, the sibling selectors.