

# special effects

CSS pseudo-elements and pseudo-classes

all code examples are in this [demo archive](#)

pseudo-elements

# pseudo-elements

- pseudo-elements are added to selectors to style certain parts of the element specified by the selector.
- syntax:

```
selector:pseudo-element  
{  
    property: value;  
}
```

# sample pseudo-elements

:after

:before

::first-letter

::first-line

- Sometimes you will see double colons (::) instead of just one (:). This is part of CSS3 and an attempt to distinguish between pseudo-classes and pseudo-elements. Most browsers support both values.

# :before and :after

- **:before** creates a pseudo-element that is the first child of the element matched.
- :after matches a virtual last child of the selected element.
- often used to add cosmetic content to an element, by using the **content** property.
- these elements are inline by default.

# :before and :after example

HTML

```
<q>Some quotes</q>, he said,  
<q>are better than none</q>.
```

CSS

```
q::before { content: "»"; }  
q::after { content: «'»; }
```

# ::first-letter

- The **::first-letter** selects the first letter of the first line of a block, if it is not preceded by any other in-line element (such as images) on its line.
- The first letter of an element is not necessary trivial to identify:
  - Punctuation preceding or immediately following the first letter is also matched by this pseudo-element.
  - A combination of the ::before pseudo-element and the content property may inject some text at the beginning of the element. In that case, ::first-letter will match the first letter of this generated content.

# CSS properties inside ::first-letter

- all font properties
- all background properties
- all margin properties
- all padding properties
- all border properties
- the color property
- the text-decoration, text-transform, letter-spacing, line-height, float properties



# ::first-letter example

```
p:first-letter {  
  font-family: "Lucida calligraphy", Forte, Georgia, serif;  
  font-size: 3em;  
  float:left;  
}
```

[link](#)

Note: this effect can be achieved by using `<span>` of a specific class around the first letter of each paragraph

# ::first-line

- The **::first-line** applies styles only to the first line of an element.
- The amount of the text on the first line depends of numerous factors, like the width of the elements or of the document, but also of the font size of the text.
- As all pseudo-elements, the selectors containing ::first-line do not match any real HTML element.
- A first line has only meaning in a block-container elements

# CSS properties inside ::first-line

- all font properties
- all background properties
- the color property
- word-spacing, letter-spacing, text-decoration, text-transform, and line-height

# ::first-line example

```
blockquote::first-line
{
  text-transform: uppercase
}
```

## [link](#)

A special style for the first line cannot be achieved without this pseudo-element selector

pseudo-classes

# pseudo-classes

- A CSS **pseudo-class** is a keyword added to selectors that specifies a special state of the element to be selected.
- Pseudo-classes let you apply a style to an element not only in relation to the content of the document tree, but also in relation to external factors like the history of the navigator (**:visited**, for example), the status of its content (like **:checked** on some form elements), or the position of the mouse (like **:hover** which lets you know if the mouse is over an element or not).

# pseudo-class selectors

**:visited**

**:active**

**:hover**

**:first-child**

**:nth-child**

- Syntax:

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

# position/number-based pseudo classes

**:first-child** - selects the first element of its type within a parent.

**:last-child** - selects the last element of its type within a parent.

**:nth-child(N)** - selects elements based on a simple provided algebraic expression

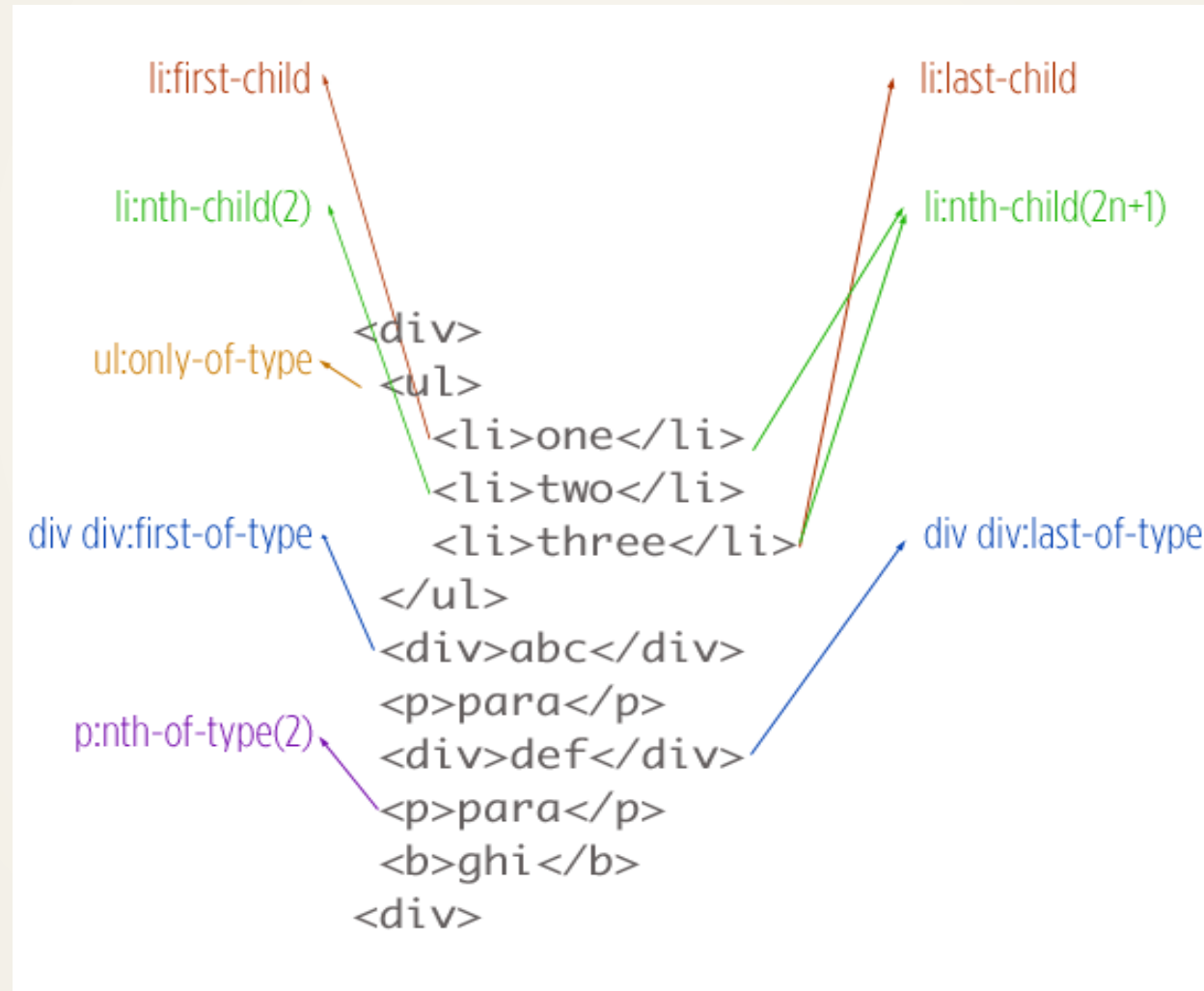
**:nth-of-type(N)** - works like :nth-child, but used in places where the elements at the same level are of different types

**:first-of-type** - selects the first element of this type within a parent.

**:last-of-type** - selects the last element of this type within a parent.



# schema of position/number-based pseudo-classes



# :nth-child example

HTML

CSS

```
<div>  
  <span>This span is limed!</span>  
  <span>This span is not. :( </span>  
  <span>But this one is!</span>  
  <span>Sadly, this one is not...  
    </span>  
</div>
```

```
span:nth-child(odd) {  
  color: lime;  
}
```

This span is limed! This span is not. :( But this one is! Sadly, this one is not...

# :nth-child example

## CSS

```
ul li:nth-child(3n+3)  
{  
  color: #ccc;  
}
```

The above CSS selects **every third** list item inside unordered lists. That is, the 3rd, 6th, 9th, 12th, etc.

# how nth-child works

- Why does it select every third element?
- Think of "n" as starting at zero and then a set of all positive integers. Then complete the equation.
- So the 3n is "3xn", and the whole equation together is "(3xn)+3". Now substituting in the zero and positive integers, we get:

$$\begin{aligned} (3 \times 0) + 3 &= 3 = 3\text{rd Element} \\ (3 \times 1) + 3 &= 6 = 6\text{th Element} \\ (3 \times 2) + 3 &= 9 = 9\text{th Element} \\ &\text{etc.} \end{aligned}$$

- How about the :nth-child(2n+1)?

$$\begin{aligned} (2 \times 0) + 1 &= 1 = 1\text{st Element} \\ (2 \times 1) + 1 &= 3 = 3\text{rd Element} \\ (2 \times 2) + 1 &= 5 = 5\text{th Element} \\ &\text{etc.} \end{aligned}$$

# Test your understanding with examples

- <http://css-tricks.com/examples/nth-child-tester/#>
- <http://css-tricks.com/useful-nth-child-recipes/>

# nth-child example: table rows

CSS

```
tr:nth-child(3n+2)
{
  background: rgba(220,0,0,0.4);
}
```

[link](#)

# nth-child example: list items

CSS

```
ol li:nth-child(odd)
{
  background: rgba(0,220,0,0.4);
}
```

```
ol li:nth-child(even)
{
  background: rgba(0,0,220,0.4);
}
```

[link](#)

# :nth-of-type(N)

- works like :nth-child, but used in places where the elements at the same level are of different types: for example alternating images and paragraphs
- if inside a div you have a number of paragraphs and a number of images and you want to select all the odd images. :nth-child won't work there, you'd use

`div img:nth-of-type(odd).`



# nth-of-type example: definition list

**HTML**

```
<dl>
  <dt>Term 1</dt>
    <dd>Definition 1.</dd>
  <dt>Term 2</dt>
    <dd>Definition 2.</dd>
  <dt>Term 3</dt>
    <dd>Definition 3.</dd>
  <dt>Term 4</dt>
    <dd>Definition 4.</dd>
  <dt>Term 5</dt>
    <dd>Definition 5.</dd>
</dl>
```

**CSS**

```
dt:nth-of-type(odd) {
  background: rgb(230,230,230);
}
dd:nth-of-type(odd) {
  background: rgb(220,220,220);
}
```

[link](#)

# event-related pseudo-classes

- **:hover** - when the mouse cursor rolls over an element, that element is in its hover state and this will select it.
- **:active** - selects when the element is being activated (being clicked on or otherwise activated). For example, for the "pressed" state of a button-style link or to make all links feel more button-like.

# :hover pseudo-class

- The **:hover** pseudo-class matches when the user designates an element with a pointing device, but does not necessarily activate it.

**Usage Note:** on touch screens :hover is problematic or impossible. The :hover pseudo-class never matches, or matches for a short moment after touching an element. As touchscreen devices are very common, it is important for web developer not to have content accessible only when hovering over it, as this content would be hidden for users of such devices.

# hover example: drop-down menu

```
<ul>
  <li>
    <a href="example.html">Menu</a>
    <ul>
      <li> <a href="ex.html">Link</a> </li>
      <li> <a href="ex.html">Submenu</a>
        <ul>
          <li> <a href="ex.html">Submenu</a>
            <ul>
              <li><a href="ex.html">Link</a></li>
              <li><a href="ex.html">Link</a></li>
            </ul>
          </li>
          <li><a href="example.html">Link</a></li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
```

HTML

CSS

```
.menu-bar ul ul {
  display: none;
}
```

```
.menu-bar li:hover > ul {
  display: block;
}
```

[link](#)

# hover example: image gallery

- [link](#)

# hover example: image filters

- [link](#)

# link-related pseudo-classes

- **:link** - selects only those elements which have an href attribute
- **:visited** - selects links that have already been visited by the current browser.

# Example: changing default appearance of links

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

```
a:visited { color: red; background-color: white }
```