

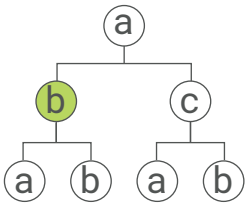
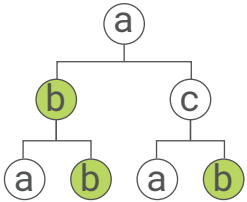
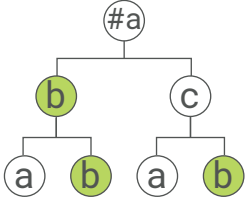




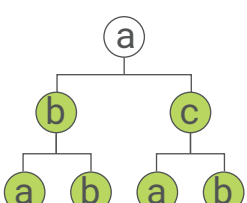






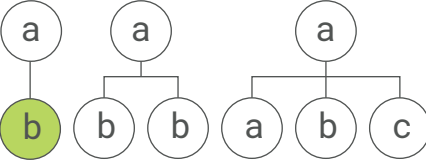


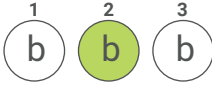
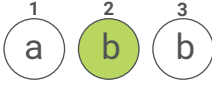
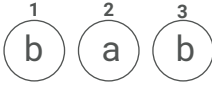
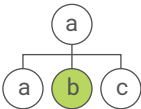
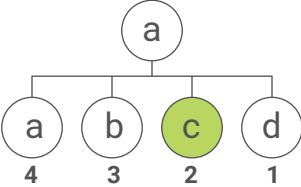
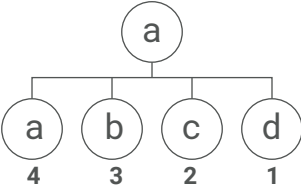
CSS selectors

> . , * + ~ : [] () { }

par Sacha RESTOUEIX

CSS selectors cheatsheet

Selector name	HTML	CSS	Infographic
Type Selector	<code><a/></code> <code></code> <code><c/></code>	<code>a {}</code>	
ID Selector	<code><..id="a"/></code> <code><..id="b"/></code> <code><..id="c"/></code>	<code>#a {}</code>	 <small>* ID is not recommended as a selector</small>
Child Selector	<code><a></code> <code></code> <code><a/></code> <code></code> <code><c></code> <code><a/></code> <code></c></code> <code></code>	<code>a > b {}</code>	
Descendant Selector	<code><a></code> <code></code> <code><a/></code> <code></code> <code><c></code> <code><a/></code> <code></c></code> <code></code>	<code>a b {}</code>	
Combine Descendant & ID Selectors	<code></code> <code></code> <code><a/></code> <code></code> <code><c></code> <code><a/></code> <code></c></code> <code></code>	<code>#a b {}</code>	
Class Selector	<code><.. class="a"/></code> <code><.. class="b"/></code> <code><.. class="c"/></code>	<code>.a {}</code>	
Combine the Class Selector	<code></code> <code><b class="x"/></code> <code><c class="x"/></code>	<code>b.x {}</code>	
Comma Combinator Selector	<code><a/></code> <code></code> <code><c/></code> <code><d/></code>	<code>a, c {}</code>	
Universal Selector	<code><a/></code> <code></code> <code><c/></code> <code><d/></code>	<code>* {}</code>	
Combine Universal Selector	<code><a></code> <code></code> <code><a/></code> <code></code> <code><c></code> <code><a/></code> <code></c></code> <code></code>	<code>a * {}</code>	

Adjacent Sibling Selector	<a/> 	a + b {}	
General Sibling Selector	<a/> <c/> 	a ~ b {}	
First Child Pseudo Selector	 	b:first-child {}	
	<a/> 		<div>first-child</div>  <p>*first-child is `a` element, not `b` element. So there is nothing to be selected.</p>
Only Child Pseudo Selector	<a> <a> <a> <a/><c/> 	b:only-child {}	
		a :only-child {}	
Last Child Pseudo Selector	 	b:last-child {}	
	 <c/>		<div>last-child</div>  <p>*last-child is `c` element, not `b` element. So there is nothing to be selected.</p>
Nth Child Pseudo Selector	 	b:nth-child(2) {}	
	<a/> 		
	 <a/> 		 <p>*nth-child(2) is `a` element, not `b` element. So there is nothing to be selected.</p>
	<a> <a/><c/> 	a :nth-child(2) {}	
Nth Last Child Selector	<a> <a/> <c/> <d/> 	a :nth-last-child(2) {}	
		c:nth-last-child(2) {}	
		a:nth-last-child(2) {}	
		b:nth-last-child(2) {}	
		d:nth-last-child(2) {}	

First of Type Selector	<pre><a/> <a/> </pre>	b:first-of-type {}	
Nth of Type Selector	<pre><a/> <a/> <a/> <a/></pre>	a:nth-of-type(2) {}	
		a:nth-of-type(even) {}	
		a:nth-of-type(odd) {}	
		a:nth-of-type(2n+1) {}	<p>* `n` is an every positive integer or zero value.</p>
Only of Type Selector	<pre><a> <a> <a> <a/><c/> </pre>	b:only-of-type {}	
Last of Type Selector	<pre><a/> </pre>	b:last-of-type {}	
	<pre><a> <c/> <c/> <d/> </pre>	a :last-of-type {}	
	<pre><a> <c/> <c/> <d/> </pre>		
	<pre><a> <c/> <b class="x"/> <c class="x"/> <b class="x"/> <c class="x"/> </pre>	.x:last-of-type {}	
	<pre><a> <b class="x"/> <c class="x"/> <b class="x"/> <c class="x"/> <c/> </pre>		<p>* Those items won't be selected as no `x` is presented</p>

Empty Selector	<pre><a/> <a>hello <a> </pre>	a:empty {}	<p>* `empty` indicates no children elements or text.</p>
Negation Pseudo-class Selector	<pre><a/> <b class="x"/></pre>	a:not(.x) {}	
	<pre><a/> <a/> <a/></pre>	a:not(:last-of-type) {}	
Attribute Selector	<pre><a/> </pre>	[for]	
	<pre><a/> <b for="y"/> <c for="z"/></pre>	a[for]	
Attribute Value Selector	<pre><a/> </pre>	a[for="x"]	
Attribute Starts with Selector	<pre> </pre>	[for^="x"]	
Attribute Ends with Selector	<pre> </pre>	[for\$="x"]	
Attribute Wildcard Selector	<pre> </pre>	[for*="x"]	

Only Child Pseudo VS Only of Type

Only Child Pseudo Selector	<pre><a> </pre>	b:only-child {}	
Only of Type Selector	<pre><a> </pre>	b:only-of-type {}	