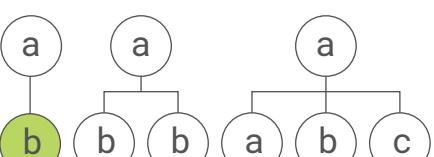
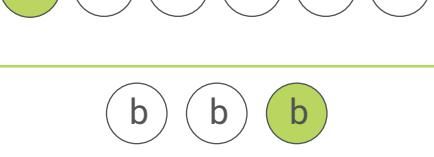
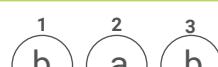
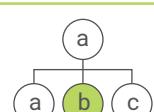
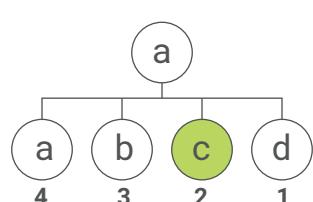
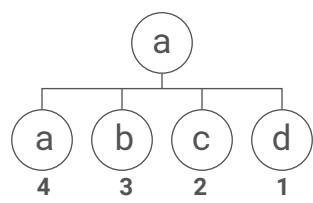
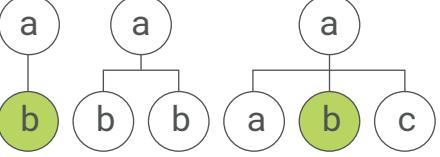
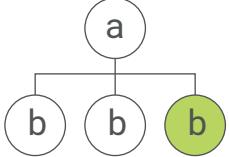
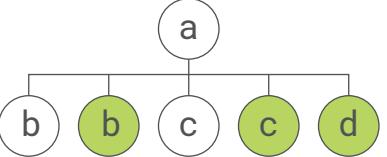
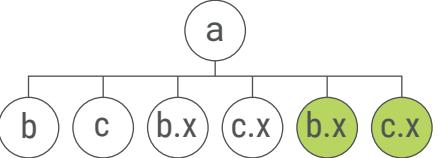
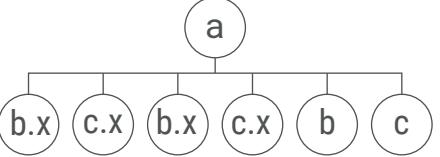


CSS selectors cheatsheet

Selector name	HTML	CSS	Infographic
Type Selector	<a/> <c/>	a {}	Three separate circles labeled 'a', 'b', and 'c'.
ID Selector	<..id="a"/> <..id="b"/> <..id="c"/>	#a {}	Three circles labeled '#a', '#b', and '#c'. <small>* ID is not recommended as a selector</small>
Child Selector	<a> <a/> <c> <a/> </c> 	a > b {}	A tree diagram where 'a' is at the root, with two children 'b' and 'c'. Each 'b' has two children 'a' and 'b'.
Descendant Selector	<a> <a/> <c> <a/> </c> 	a b {}	A tree diagram where 'a' is at the root, with two children 'b' and 'c'. Each 'b' has two children 'a' and 'b'.
Combine Descendant & ID Selectors	 <a/> <c> <a/> </c> 	#a b {}	A tree diagram where 'a' is at the root, with two children 'b' and 'c'. Each 'b' has two children 'a' and 'b'. Only the 'b' under '#a' is highlighted.
Class Selector	<.. class="a"/> <.. class="b"/> <.. class="c"/>	.a {}	Three circles labeled '.a', '.b', and '.c'.
Combine the Class Selector	 <b class="x"/> <c class="x"/>	b.x {}	Three circles labeled 'a.x', 'b.x', and 'c.x'.
Comma Combinator Selector	<a/> <c/> <d/>	a, c {}	Four separate circles labeled 'a', 'b', 'c', and 'd'.
Universal Selector	<a/> <c/> <d/>	* {}	Four separate circles labeled 'a', 'b', 'c', and 'd'.
Combine Universal Selector	<a> <a/> <c> <a/> </c> 	a * {}	A tree diagram where 'a' is at the root, with two children 'b' and 'c'. Each 'b' has two children 'a' and 'b'. Only the 'b' under 'a' is highlighted.

Adjacent Sibling Selector	<a/> 	a + b {}	
General Sibling Selector	<a/> <c/> 	a ~ b {}	
First Child Pseudo Selector	 	b:first-child {}	
	<a/> 		 <small>*first-child is 'a' element, not 'b' element. So there is nothing to be selected.</small>
Only Child Pseudo Selector	<a> <a> <a> <a/><c/> 	b:only-child {}	
		a :only-child {}	
Last Child Pseudo Selector	 	b:last-child {}	
	 <c/>		 <small>*last-child is 'c' element, not 'b' element. So there is nothing to be selected.</small>
Nth Child Pseudo Selector	 	b:nth-child(2) {}	
	<a/> 		
	 <a/> 		
	<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	<a/> <a/> 	b:first-of-type {}	
Nth of Type Selector	<a/> <a/> <a/> <a/>	a:nth-of-type(2) {}	
		a:nth-of-type(even) {}	
		a:nth-of-type(odd) {}	
		a:nth-of-type(2n+1) {}	 * `n` is an every positive integer or zero value.
Only of Type Selector	<a> <a> <a> <a/><c/> 	b:only-of-type {}	
Last of Type Selector	<a/> 	b:last-of-type {}	
	<a> <c/> <c/> <d/> 	a :last-of-type {}	
			
	<a> <c/> <b class=x /> <c class=x /> <b class=x /> <c class=x /> 	.x:last-of-type {}	
	<a> <b class=x /> <c class=x /> <b class=x /> <c class=x /> <c/> 		

* Those items won't be selected as no '.x' is presented

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> <pre><a> <a/><c/> </pre>	b:only-of-type {}	

* Happy codesign today!!! *

This cheatsheet is designed for a quick search on CSS selectors. :)
There are so many CSS selectors with unfamiliar symbols, > . , * + ~ [] etc, so I am often confused with how CSS selectors work.
I wish this infographic helps you find proper CSS selectors. Print this lovely cheatsheet out and stick it on the wall.
And if you need more details on CSS selectors, check out my blog at medium.com/@nanacodesign.

I would love to hear your feedback on how I can make it better. Please leave your comments on my twitter [@nanacodesign](https://twitter.com/nanacodesign).
Thank you !!! 감사합니다!!!