

Specificity is basically a measure of how specific a selector is — how many elements it *could* match. Element selectors have low specificity. Class selectors have a higher specificity, so will win against element selectors. ID selectors have an even higher specificity, so will win against class selectors. The only way to win against an ID selector is to use `!important`.

The amount of specificity a selector has is measured using four different values (or components), which can be thought of as thousands, hundreds, tens and ones — four single digits in four columns:

1. Thousands: Score one in this column if the declaration is inside a `style` attribute (such declarations don't have selectors, so their specificity is always simply 1000.) Otherwise 0.
2. Hundreds: Score one in this column for each ID selector contained inside the overall selector.
3. Tens: Score one in this column for each class selector, attribute selector, or pseudo-class contained inside the overall selector.
4. Ones: Score one in this column for each element selector or pseudo-element contained inside the overall selector.

Note: Universal selector (`*`), combinators (`+`, `>`, `~`, `' '`) and negation pseudo-class (`:not`) have no effect on specificity.

Selector	Thousands	Hundreds	Tens	Ones	Total specificity
h1	0	0	0	1	0001
#important	0	1	0	0	0100
h1 + p::first-letter	0	0	0	3	0003
li > a[href*="en-US"] > .inline-warning	0	0	2	2	0022
#important div > div > a:hover, inside an element's style attribute	1	1	1	3	1113