

CSS specificity

The specificity as a ranking of CSS rules, which CSS rules are higher, is applied to the element.

What is CSS Specificity?

If there are two or more CSS rules that conflict with the same element, the browser will follow some rules to determine the priority for rules, which ones have the highest priority. will be applied.

We have the following example to make it easier to imagine:

In the code below, the background-color section for the div element was simultaneously reviewed twice in both #test selector with red background and selector div with yellow background, running the result will produce The output has a red background because the id #test has a higher priority than the other object.

This is the div element

To make it easier to imagine, imagine your Specificity as a ranking of CSS rules, which CSS rule is higher, and it applies to the element.

So what is that ranking like? Let's find out in TipsMake.com in the Specificity Hierarchy hierarchy below.

Specificity rankings - Specificity Hierarchy

If there are two or more CSS rules that target one element, then the browser will follow only one rule, see in the CSS rule list which is the highest rule to select and apply to the element.

Here we will have the rankings for CSS rules so that the browser obeys when applying CSS, arranged in priority order from top to bottom:

1. **Inline style** : set CSS properties directly inside an element.
For example:

2. **ID:** set CSS properties for a unique identifier in a page.
For example: `#menu`, `#header`
3. **Class, attribute, pseudo-class:** eg class like `.menu`, `.header` ., attributes like `a[target]` and pseudo-class like: `hover`, `:focus` .
4. **Element, pseudo-element:** element such as `h1`, `h2`, `div`, `p` . and pseudo-element such as `::before`, `::after`, `::selection`.

Calculation Specificity

Calculate the value of CSS Specificity to see which rules apply to the element, making it easy to make mistakes when writing CSS.

Specificity is counted as positive integers. As follows:

1. **Inline style** is 1000.
2. **ID** is 100.
3. **Class, attribute, pseudo-class** is 10.
4. **Element, pseudo-element** is 1.

For example:

```
A: h1
B: #content h1
C:
```

1. A is 1 (an element is h1).
2. B is 101 (an ID is content, an element is h1).
3. C is 1000 (inline style).

Because *1 rule (C) has the highest priority, so the background color for h1 element is white (#ffffff).*

Rules for Specificity

1. If there are equal Specificities, the last write rule will be applied

Where we have two equal CSS rules like this:

Obviously, the element's specificity value is the same (all h1 elements) now we will rule which selector is closest to that element to be applied. So the result will be:

Website Quản Trị Mạng - Heading 1

2. ID selector will have a higher specificity value than Attribute selector

In case we have CSS rules like this:

Website TipsMake.com

By law, ID selector will be applied. The result is:

Website Quản Trị Mạng

3. Class selector will have a higher specificity value than element selector

A class selector like .intro will "defeat" every single element like h1, p, div .

Website TipsMake.com

Website Quản Trị Mạng

4. Specific context selector will have higher specificity value

For example, in the following situation:

From external CSS file:

In the HTML file:

The second rule will prevail.

5. Universal selector (*) and inherited value with specificity of 0

Universal selector or CSS's entire selection symbol (*) will select all objects of any type.

```
/* Ch?n t?t c? ??i t??ng */
* {
  color: green;
}
```

Universal selector has specificity of 0 and properties have values ??that are inherited, also have specificity of 0.

Last lesson: Unit - Unit of measurement in CSS

Next lesson: Rounded Corner in CSS

You finished reading the article "**CSS specificity**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

