CSS Specificity

❮ Previous Next ❯

What is Specificity?

If there are two or more CSS rules that point to the same element, the selector with the highest specificity value will "win", and its style declaration will be applied to that HTML element.

Think of specificity as a score/rank that determines which style declaration is ultimately applied to an element.

Look at the following examples:

Example 1

In this example, we have used the "p" element as selector, and specified a red color for this element. The text will be red:

```
<html>
<head>
  <style>
    p {color: red;}
  </style>
</head>
<body>

<p>Hello World!</p>
```

```
</body>
</html>
```

Try it Yourself »

Now, look at example 2:

Example 2

In this example, we have added a class selector (named "test"), and specified a green color for this class. The text will now be green (even though we have specified a red color for the element selector "p"). This is because the class selector is given higher priority:

```
<html>
<head>
  <style>
    .test {color: green;}
    p {color: red;}
  </style>
</head>
<body>

<p class="test">Hello World!</p>

</body>
</html>
```

Try it Yourself »

Now, look at example 3:

Example 3

In this example, we have added the id selector (named "demo"). The text will now be blue, because the id selector is given higher priority:

```
<html>
<head>
  <style>
```

```
    #demo {color: blue;}
    .test {color: green;}
    p {color: red;}
  </style>
</head>
<body>

<p id="demo" class="test">Hello World!</p>

</body>
</html>
```

Try it Yourself »

Now, look at example 4:

Example 4

In this example, we have added an inline style for the "p" element. The text will now be pink, because the inline style is given the highest priority:

```
<html>
<head>
  <style>
    #demo {color: blue;}
    .test {color: green;}
    p {color: red;}
  </style>
</head>
<body>

<p id="demo" class="test" style="color: pink;">Hello World!</p>

</body>
</html>
```

Try it Yourself »

Specificity Hierarchy

Every CSS selector has its place in the specificity hierarchy.

There are four categories which define the specificity level of a selector:

1. **Inline styles** - Example: <h1 style="color: pink;">
2. **IDs** - Example: #navbar
3. **Classes, pseudo-classes, attribute selectors** - Example: .test, :hover, [href]
4. **Elements and pseudo-elements** - Example: h1, ::before

How to Calculate Specificity?

Memorize how to calculate specificity!

Start at 0, add 100 for each ID value, add 10 for each class value (or pseudo-class or attribute selector), add 1 for each element selector or pseudo-element.

Note: Inline style gets a specificity value of 1000, and is always given the highest priority!

Note 2: There is one exception to this rule: if you use the `!important` rule, it will even override inline styles!

The table below shows some examples on how to calculate specificity values:

Selector	Specificity Value	Calculation
p	1	1
p.test	11	1 + 10
p#demo	101	1 + 100
<p style="color: pink;">	1000	1000
#demo	100	100
.test	10	10
p.test1.test2	21	1 + 10 + 10
#navbar p#demo	201	100 + 1 + 100

*	0	0 (the universal selector is ignored)

The selector with the highest specificity value will win and take effect!

Consider these three code fragments:

Example

```
A: h1
B: h1#content
C: <h1 id="content" style="color: pink;">Heading</h1>
```

The specificity of A is 1 (one element selector)
The specificity of B is 101 (one ID reference + one element selector)
The specificity of C is 1000 (inline styling)

Since the third rule (C) has the highest specificity value (1000), this style declaration will be applied.

More Specificity Rules Examples

Equal specificity: the latest rule wins - If the same rule is written twice into the external style sheet, then the latest rule wins:

Example

```
h1 {background-color: yellow;}
h1 {background-color: red;}
```

Try it Yourself »

ID selectors have a higher specificity than attribute selectors - Look at the following three code lines:

Example

```
div#a {background-color: green;}
#a {background-color: yellow;}
div[id=a] {background-color: blue;}
```

Try it Yourself »

the first rule is more specific than the other two, and will therefore be applied.

Contextual selectors are more specific than a single element selector - The embedded style sheet is closer to the element to be styled. So in the following situation

Example

```
From external CSS file:
#content h1 {background-color: red;}

In HTML file:
<style>
#content h1 {background-color: yellow;}
</style>
```

the latter rule will be applied.

A class selector beats any number of element selectors - a class selector such as .intro beats h1, p, div, etc:

Example

```
.intro {background-color: yellow;}
h1 {background-color: red;}
```

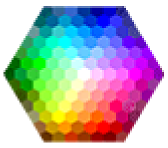
Try it Yourself »

The universal selector (*) and inherited values have a specificity of 0 - The universal selector (*) and inherited values are ignored!

❮ Previous Next ❯

W3schools
Launch Your Web Development Career in Just 6 Months with
W3Schools Bootcamps
Learn fullstack development from expert instructors. All for $2995. Apply now!

COLOR PICKER



Get started

Unlock Full Access

~~$1,465.00~~

$695.00

Save $770

Report Error

Spaces

Upgrade

Newsletter

Get Certified

Top Tutorials

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C++ Tutorial
jQuery Tutorial

Top References

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Get Certified

HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate

FORUM | ABOUT