CSS Specificity & Cascade | Cheat Sheet

1. Specificity

Specificity is how browsers decide which CSS property values are the **most relevant** to an HTML element and, therefore, will be applied.

The following list of CSS Selectors is in the lowest to highest order by specificity.

1. Type (tag name) Selector
2. Class Selector
3. ID Selector

1.1 Type Selector & Class Selector

A Class Selector is **more specific** compared to Type (tag name) Selector as it selects only the HTML elements that have **specific class attribute value** in the HTML document.

Help

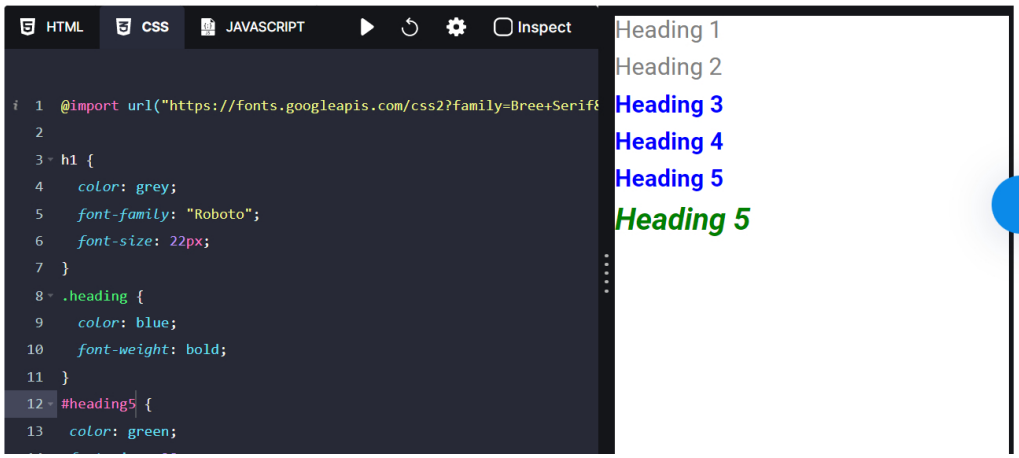
Help

Note

It doesn't overwrite the entire CSS Ruleset but only overwrites the CSS properties that are the same.

1.2 Class Selector & ID Selector

An ID Selector is more specific when compared to a Class Selector as we provide a unique ID within the HTML document and it selects only a **single** HTML Element.



Help

```
  font-size: 28px;
  font-style: italic;
}
```

2. Inline Styles

The Inline styles are applied **directly** to an HTML element. They use the HTML `style` attribute, with CSS property values defined within it.

Help

Syntax:

```
<tag style = "property1: value1; property2: value2; ...">Content</tag>
```

A HTML `style` attribute value can consist of one or more CSS property values.

HTML CSS JAVASCRIPT Inspect

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.j
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/di
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.
  </head>
  <body>
    <h1>Heading 1</h1>
    <h1>Heading 2</h1>
    <h1 class="heading">Heading 3</h1>
    <h1 class="heading">Heading 4</h1>
    <h1 class="heading" id="heading5" style="color: white; backgr
  </body>
</html>
```

Heading 1
Heading 2
Heading 3
Heading 4
Heading 5

Help

Note

Inline Styles have the highest specificity. They overwrite any other styles specified using CSS Selectors.

Using Inline Styles is not recommended because

- Inline Styles are not reusable.
- Writing HTML and CSS separately increases code readability.

3. CSS Cascade

Help

The source order of CSS Rulesets matters. When two CSS Rulesets have equal specificity, the one that comes last in the CSS is applied.

HTML CSS JAVASCRIPT Inspect

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.j
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/di
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.
  </head>
  <body>
    <h1 class="style-2 style-1">About India</h1>
  </body>
</html>
```

About India

Help

Note

The styles that apply to the HTML Elements are not determined by the **order the classes** defined in the HTML `class` attribute, but instead the order in which they appear in the CSS.

3.1 The !important exception

It is a special piece of CSS used to make a particular CSS property and value the **most specific thing**, irrespective of source order and specificity.

HTML

```html
<h1 class="style-1">About India</h1>
```

CSS

```css
.style-1 {
  color: green;
}
h1 {
  color: orange !important;
}
```

Help

The only way to override a `!important` property value is to include another `!important` property value. The added property value should either come later in the order or should be of higher specificity.

HTML CSS JAVASCRIPT Inspect

```css
@import url("https://fonts.googleapis.com/css2?family=Bree+Serif&
.style-1 {
  color: green !important;
}
h1 {
  color: orange !important;
}
.style-2 {
  color: blue !important;
}
```

About India

Help

Note

Always look for a way to use specificity before even considering `!important` .

Only use `!important` when you want to override foreign CSS (from external libraries, like Bootstrap).

Help

Submit Feedback

MARKED AS COMPLETE

Notes Discussions

Notes

+ NEW NOTE

Title
Take a Note

CANCEL SAVE

Help