Specificity in CSS

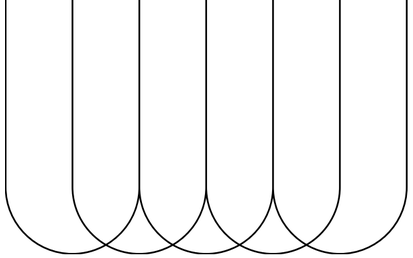
CSS Specificity

Specificity in CSS determines which styles are applied to an element when multiple conflicting rules are present. It's essentially a ranking system that CSS uses to decide which rule "wins" when more than one rule is applied to the same element.

The Cascade Rule

Understanding the cascade algorithm helps you understand how the browser resolves conflicts of multiple styles rules. The cascade algorithm has 3 different stages:

1. **Position and order of appearance:** The order in which CSS appears and where it comes from whether that is a browser style, CSS from a browser extension, or your authored CSS.
2. **Specificity:** It is a mechanism browsers use to determine most specific style rule for an element.
3. **Importance:** Certain CSS rules carry greater weight than others, particularly when the !important declaration is used.



CSS Specificity

Position and order of appearance

When two CSS rules have selectors of equal specificity, the last one declared takes precedence. In an HTML page, styles can be applied in various ways: using a <link> tag, a <style> tag, or directly within an element's style attribute. If there are multiple <link> tags, the styles from the one at the bottom of the page will be applied, as well as for <style> tags. The styles in a lower-positioned <style> tag will have priority.

An inline style declared directly within an element's style attribute will override all other CSS, regardless of its position in the document.

Specificity

CSS specificity determines which style rules get applied to an element when there are conflicts. Higher specificity means the style will be used.

Basis for Specificity

1. **Inline Styles :** applied directly to the element using the style attribute
2. **ID Selectors**
3. **Classes, attributes, and pseudo-classes:** (e.g., .header, [type="text"], :hover)
4. **Element types and pseudo-elements:** (Eg, div, nav, ::before, ::after)

CSS Specificity

How Specificity is Calculated?

To calculate specificity, assign a value to each part of the selector as given below::

- Universal Selector: 0
- Element selectors and pseudo-elements: 1
- Class selectors, attribute selectors, and pseudo-classes: 10
- ID selectors: 100
- Inline styles: 1000

Then, add up the values of all the parts in the selector.

Example:

```
main p{color: green;}  /*Calculation:  Element selector (1) + Element selector (1) = 2 */

#banner p{color: yellow;} /*Calculation: Id selector (100) + Element selector (1) = 101 */

h1 + p{color: yellow;} /*Calculation: Element selector (1) + Element selector (1) = 2 */
```

In above example the paragraph will be rendered in Yellow color as it has the highest specificity of 101.

CSS Specificity

How Specificity is Calculated?

Calculate the specificity of following:

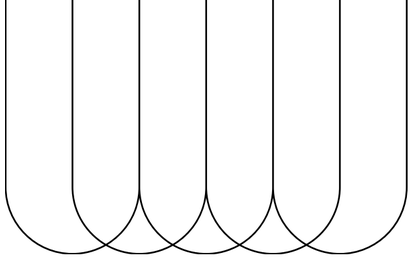
```
a.btn.btn[href]:hover {color: red;}
```

Importance

The !important flag in CSS is a special rule that can be used to give a CSS property higher priority over other conflicting rules, regardless of their specificity. When a rule is marked as !important, it will override any other rule that might apply to the same property, unless that rule also has the !important flag and has a higher specificity.

```
#banner p.banner-text strong{color: green}
p strong{color: red !important}
```

In above example text inside the strong tag will be red even though the specificity of that rule is far less than the rule above it.



indixpert

CSS Specificity

Conclusion

Browser applies css on html based on cascade rules that is based on three major factors:

- Position & Order of the rule
- Specificity of the rule
- and Importance of the rule.

CSS rule with higher specificity will override the other ones.

If specificity is same for more rules applied to an element the order will determine which rule will be applied

If a rule has !important flag it would override all other rules applied even those rules have higher specificity.

THANK YOU

PHONE NUMBER

(+91) 778 899 2897

WEBSITE

www.indixpert.com