W01 Learning Activity: CSS Specificity

Overview

CSS stands for Cascading Style Sheets. The term "cascade" represents the prioritization stack of the CSS rules that are applied to a specific element. The specific CSS rule that is applied to an element is determined by the CSS rule **specificity** value.

Prepare

Specificity is represented as a four-part numerical value that reflects the weight of a CSS rule based on the types of **selectors** used. The higher the specificity, the more precisely the rule targets an element, thereby determining which rule takes precedence when multiple rules apply.

This table maps the parts from highest specificity to the lowest.

Category	Example(s)	Specificity Value
Inline	`style="color: red;"`	1 0 0 0
ID	`#currentDate`	0 1 0 0
Classes, attributes, and pseudo-classes	`.myClass`, `[type="text"]`, `:hover`	0 0 1 0
Elements and pseudo-elements	`div, :before`	0 0 0 1

The specificity value is calculated by counting the number of selectors in each category and assigning a value to each category. The values are then combined to form a four-part specificity value.

For example, the CSS rule selector `#sub-menu .nav a:hover` has a specificity value of **0 1 1 2** because it contains:

- 1 ID selector (`#sub-menu`) — 0 1 0 0

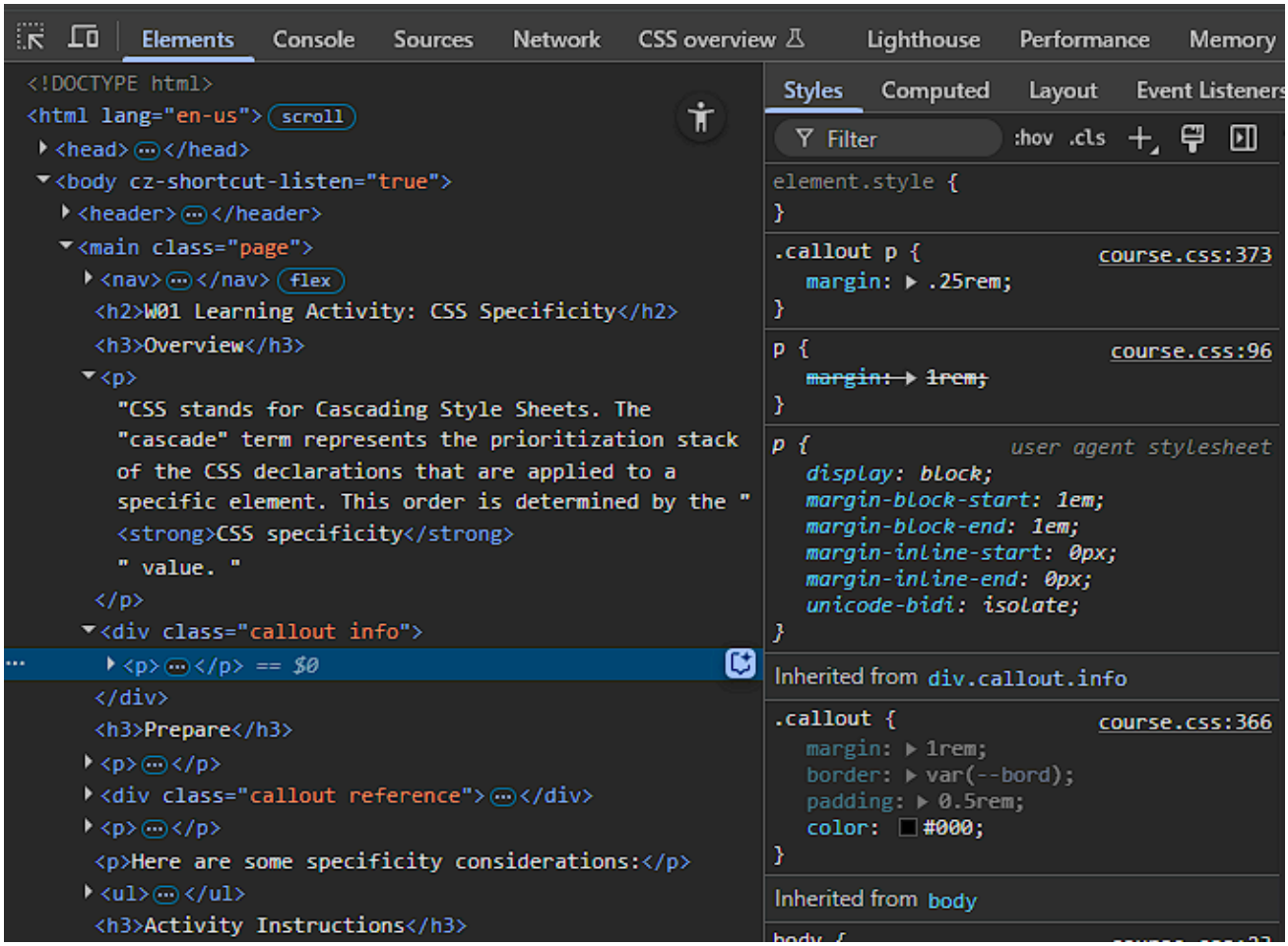
- 1 class selector (`.nav`) — 0 0 1 0
- 1 element selector (`a`) — 0 0 0 1
- 1 pseudo-class selector (`:hover`) — 0 0 0 1

> Typically you do not spend time determining specificity values during design because you will become familiar with CSS assignment patterns. However, the learning the CSS application principles will help you troubleshoot why a rule is or is not being applied to a specific element.

Troubleshooting

Here are two recommendations to help you troubleshoot the application of CSS rules by selectors:

1. Use the browser's built-in DevTools **Elements – Styles** panel. As you inspect an element by clicking on the element name in the Elements panel, the applied CSS rules are displayed for you to analyze and toggle on and off to test.



DevTools Elements-Styles panel screenshot

2. Use a CSS specificity calculator to compare CSS selector's specificity.

> Note that this calculator displays specificity as a three-part value (a b c), omitting the highest category used exclusively for inline styles. Inline styles have an implicit top-level specificity of (1 0 0 0), but since the calculator focuses on selectors written in stylesheets, this top-level value is not shown.

Here are some other specificity considerations:

- The universal selector `*` has no specificity — 0 0 0 0.
- CSS selector combinators such as descendent `div p`, child `div > p`, adjacent sibling selector `div + p`, and others have a specificity of 0 0 0 2 as expected.
- properties with `!important` are applied even if the selector is less specific.

> Using `!important` should be avoided because it can make debugging and maintaining CSS more difficult. It is better to use a more specific selector or to refactor the CSS.

Activity Instructions

1. Open this page in a new browser tab.
2. Open up the browser's **DevTools**.
3. In the **Elements** panel, select the first paragraph `p`.
4. Note in the **Styles** tab how many `color` declarations are applied to the paragraph. This is the CSS stack.

> ▶ **How many colors are declared and part of this stack and why are they in this order?**

5. Use the online CSS Calculator to enter the selectors to verify the stack order. For example, the selector `.myClass` has a specificity value of 0 1 0 because it is a class selector.

Optional Resources

Specificity – MDN

CSS Stats – Enter a URL and see an overview of the CSS stats for the page including the average specificity.

Back