

4.2 CSS Selectors



claw machine toys - answered by Keith Aquino

- A **CSS selector** selects the HTML element(s) you want to style.
- CSS selectors are used to "find" (or select) the HTML elements you want to style.

Selection	Selector syntax	Sample code
The element name	The element's name	<ul style="list-style-type: none">• This selects all paragraph elements: <pre>p { text-align: center; color: red; }</pre>
An element's <code>id</code>	<code>#id</code>	<ul style="list-style-type: none">• This selects the element with the <code>id</code> <code>para1</code>: <pre>#para1 { text-align: center; color: red; }</pre>
An element's <code>class</code>	<code>.class</code>	<ul style="list-style-type: none">• This selects all elements with the class, <code>center</code>: <pre>.center { text-align: center; }</pre>

		<pre>color: red; }</pre>
		<ul style="list-style-type: none"> This selects all <code><p></code> elements with the class, <code>center</code>: <pre>p.center { text-align: center; color: red; }</pre>
		<ul style="list-style-type: none"> This selects all <code><p></code> elements with the classes <code>journal</code> and <code>grid</code> (they must have both of these classes) <pre>.journal.grid { color: gray; }</pre>

Grouping Selector

- The grouping selector selects all the HTML elements with the same style definitions.
- Look at the following CSS code (the `h1`, `h2`, and `p` elements have the same style definitions):

```
h1 {
  text-align: center;
  color: red;
}
```

```
h2 {
  text-align: center;
  color: red;
}
```

```
p {
  text-align: center;
  color: red;
}
```

- It will be better to group the selectors, to minimize the code.
- To group selectors, separate each selector with a comma.

```
h1, h2, p {
  text-align: center;
}
```

```
color: red;  
}
```

CSS Cascading Rules

- CSS follows a priority system in the event of conflicting styles
 1. **Specificity.** The more specific a selector is, the higher the priority it has.
 - For example, an ID selector (`#elementID`) has higher specificity than a class selector (`.classname`), and a class selector has higher specificity than an element selector (e.g., `p`).
 - In the case that a tag is selected by multiple selectors, the following precedence will be followed (from highest priority to last priority)
 1. Inline styles
 2. Internal id selector
 3. Internal class selector
 4. External id selector
 5. External class selector
 6. External element selector
 - Styles that are applied to a **parent element** apply to the **child elements** by default.
 - However, selecting a specific element also have a higher priority than elements that select that element's parent element.

- As an example, take a look at the following code:

```
<style>  
  p {  
    color: red;  
  }  
  div {  
    color: blue;  
  }  
</style>  
<div>  
  <p>This is a child paragraph inside a parent div.</p>  
  <p>This is another paragraph inside a parent div.</p>  
</div>
```

- In the output, both paragraphs are **red**, because the `p` selector is prioritized over its parent element's selector.

2. **Importance.** Some styles are considered more important than others.

- The `!important` CSS tag is a declaration that can be applied to a CSS property value to give it higher specificity, making it override other conflicting styles.

- When an attribute is marked as `!important`, it gains priority over non-important styles, even if they have higher priority or are defined later in the stylesheet.
- However, it's important to use `!important` sparingly to avoid making the CSS difficult to manage.
- Here's an example of its application:

```
p {  
  color: red !important;  
}  
  
p {  
  color: blue;  
}
```

- In this case, the text color of all `<p>` elements will be red because the `color: red !important` declaration takes precedence over the `color: blue` declaration. Without `!important`, the text color would be blue.

3. **Source Order:** If two rules have the same specificity and importance, the rule that appears later in the CSS file will take precedence.
 - The rule defined further down the CSS file will override the previous rule for the same element.

Additional Material

- **References**
 - [W3Schools](#)
- **Recommended watch**

7 ways to deal with CSS

