CSS Selectors

Intro

We know about the basic CSS properties and how we can fiddle with the style of elements. Now we need a way to actually apply styles to specific elements and that is where selectors come in. Selectors, unsurprisingly, let us select element to apply styles to. Broadly speaking we can select things based on its type, its class or its id. When two selectors try to apply different values to the same property things targeting id have the highest precedence, then class then type.

The ID Selector

The ID selector targets the single element with a matching ID and is of the following syntax `#id` . So to target a form with an id of "form1" we would use `#form1` . This is fairly straight forward as there should only ever be one element with that ID.

The Class Selector

The class selector selects all elements of a given class. The syntax is `.class` . So `.navigation` will target all elements with the class attribute of 'navigation'.

Type Selectors

Type selectors get significantly more complicated. These selectors select elements based on their type and relationship to other elements.

Basic Type
You can specify an element name directly. For example `p` will target all `p` elements.

Descendant Selector
This will target all elements of type 'x' which are a descendant of type 'y' with the syntax `y x` . So `nav ul` will apply a style to all `ul` elements which are contained in a `nav` element.

Child Selector
The child selector will target all element of type x which are direct children of type y. The syntax is `y>x` . So `nav>ul` would target `ul` elements who's direct parent is a `nav` element. But unlike the decedent selector it would not target a `ul` within a `div` within a `nav` because the ul is a child of the div, not the nav.

Adjacent Sibling
This will target all of element type x which is at the same level on the tree and immediately after an element of type y. The syntax is `y+x` . So `h1+p` would target a `p` if it directly followed an `h1` . You might use this if you want your headline paragraph to always look different than the rest of the paragraphs in a news article for example.

General Sibling
This is like the adjacent sibling except it does not need to immediate follow. The syntax here is `y~x` . So `h1~p` would target all paragraphs on the same level as the `h1` element, not just the one immediately after it.

Activity

At this point you have all the tools you need to make a basic web page and change its looks. Draw a simple layout on paper, then try to duplicate it using HTML and CSS.

Review

Having completed this sub-module you should be able to create rules to select pretty much any element or set of elements on a page. Sometimes you may need to add classes to elements where that makes sense but that is no big deal.