

# Syntax and Selectors

#### Link Element

The k> element is used to link HTML documents to external resources like CSS files. It commonly uses:

- href attribute to specify the URL to the external resource
- rel attribute to specify the relationship of the linked document to the current document
- type attribute to define the type of content being linked

```
<!-- How to link an external stylesheet with href, rel, and type attributes -->
k
href="./path/to/stylesheet/style.css"
rel="stylesheet" type="text/css">
```

# **Purpose of CSS**

CSS, or Cascading Style Sheets, is a language that is used in combination with HTML that customizes how HTML elements will appear. CSS can define styles and change the layout and design of a sheet.

### Write CSS in Separate Files

CSS code can be written in its own files to keep it separate from the HTML code. The extension for CSS files is .css. These can be linked to an HTML file using a <link> tag in the <head> section.

```
<head>
     link href="style.css" type="text/css"
rel="stylesheet">
</head>
```

### Write CSS in HTML File

CSS code can be written in an HTML file by enclosing the code in <style> tags. Code surrounded by <style> tags will be interpreted as CSS syntax.

```
<head>
  <style>
    h1 {
        color: blue;
    }
    </style>
</head>
```

# **Inline Styles**

CSS styles can be directly added to HTML elements by using the style attribute in the element's opening tag. Each style declaration is ended with a semicolon. Styles added in this manner are known as *inline styles*.

```
<h2 style="text-align: center;">Centered
text</h2>
```

```
Blue, 18-point text
```



### Separating HTML code from CSS code

It is common practice to separate content code in HTML files from styling code in CSS files. This can help make the code easier to maintain, by keeping the syntax for each file separate, and any changes to the content or styling can be made in their respective files.

### **Class and ID Selectors**

CSS classes can be reusable and applied to many elements. Class selectors are denoted with a period . followed by the class name. CSS ID selectors should be unique and used to style only a single element. ID selectors are denoted with a hash sign # followed by the id name.

```
/* Selects all elements with
class="column" */
.column {
}
/* Selects element with id="first-item"
*/
#first-item {
}
```

# **Groups of CSS Selectors**

Match multiple selectors to the same CSS rule, using a comma-separated list. In this example, the text for both  $h1\,$  and  $h2\,$  is set to red.

# n2 is set to red.

h1, h2 {

### **Selector Chaining**

CSS selectors define the set of elements to which a CSS rule set applies. For instance, to select all <p> elements, the p selector can be used to create style rules.

### **Chaining Selectors**

CSS selectors can be chained so that rule sets apply only to elements that match all criteria. For instance, to select <h3> elements that also have the section-heading class, the selector h3.section-heading can be used.

```
/* Select h3 elements with the section-
heading class */
h3.section-heading {
  color: blue;
```

```
code cademy
```

```
/* Select elements with the section-
heading and button class */
.section-heading.button {
  cursor: pointer;
}
```

# **CSS Type Selectors**

CSS type selectors are used to match all elements of a given type or tag name. Unlike for HTML syntax, we do not include the angle brackets when using type selectors for tag names. When using type selectors, elements are matched regardless of their nesting level in the HTML.

```
/* Selects all  tags */
p {
}
```

### **CSS** class selectors

The CSS class selector matches elements based on the contents of their class attribute. For selecting elements having calendar-cell as the value of the class attribute, a . needs to be prepended.

```
.calendar-cell {
  color: #fff;
}
```

### HTML attributes with multiple values

Some HTML attributes can have multiple attribute values. Multiple attribute values are separated by a space between each attribute.

```
<div class="value1 value2 value3"></div>
```

### **Selector Specificity**

Specificity is a ranking system that is used when there are multiple conflicting property values that point to the same element. When determining which rule to apply, the selector with the highest specificity wins out. The most specific selector type is the ID selector, followed by class selectors, followed by type selectors. In this example, only color: blue will be implemented as it has an ID selector whereas color: red has a type selector.

```
h1#header {
  color: blue;
} /* implemented */

h1 {
  color: red;
} /* Not implemented */
```

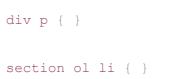
### **CSS ID selectors**

The CSS ID selector matches elements based on the contents of their id attribute. The values of id attribute should be unique in the entire DOM. For selecting the element having job-title as the value of the id attribute, a # needs to be prepended.

```
#job-title {
  font-weight: bold;
}
```

### **CSS** descendant selector

The CSS descendant selector combinator is used to match elements that are descended from another matched selector. They are denoted by a single space between each selector and the descended selector. All matching elements are selected regardless of the nesting level in the HTML.



code cademy



