Attribute selectors

Previous
Overview: Introduction to CSS
Next

Attribute selectors are a special kind of selector that will match elements based on their attributes and attribute values. Their generic syntax consists of square brackets (`[]`) containing an attribute name followed by an optional condition to match against the value of the attribute. Attribute selectors can be divided into two categories depending on the way they match attribute values: **Presence and value** attribute selectors and **Substring value** attribute selectors.

Presence and value attribute selectors

These attribute selectors try to match an exact attribute value:

- `[attr]` : This selector will select all elements with the attribute `attr`, whatever its value.
- `[attr=val]` : This selector will select all elements with the attribute `attr`, but only if its value is `val`.
- `[attr~=val]`: This selector will select all elements with the attribute `attr`, but only if `val` is one of a space-separated list of words contained in `attr`'s value.

Let's look at an example featuring the following HTML snippet:

```
Ingredients for my recipe: <i lang="fr-FR">Poulet basquaise</i>
<ul>
  <li data-quantity="1kg" data-vegetable>Tomatoes</li>
  <li data-quantity="3" data-vegetable>Onions</li>
  <li data-quantity="3" data-vegetable>Garlic</li>
  <li data-quantity="700g" data-vegetable="not spicy like chili">Red pepper</li>
  <li data-quantity="2kg" data-meat>Chicken</li>
  <li data-quantity="optional 150g" data-meat>Bacon bits</li>
  <li data-quantity="optional 10ml" data-vegetable="liquid">Olive oil</li>
  <li data-quantity="25cl" data-vegetable="liquid">White wine</li>
</ul>
```

And a simple style sheet:

```
/* All elements with the attribute "data-vegetable"
   are given green text */
```

```
[data-vegetable] {
  color: green;
}

/* All elements with the attribute "data-vegetable"
   with the exact value "liquid" are given a golden
   background color */
[data-vegetable="liquid"] {
  background-color: goldenrod;
}

/* All elements with the attribute "data-vegetable",
   containing the value "spicy", even among others,
   are given a red text color */
[data-vegetable~="spicy"] {
  color: red;
}
```

The result is as follows:

Ingredients for my recipe: *Poulet basquaise*

- Tomatoes
- Onions
- Garlic
- Red pepper
- Chicken
- Bacon bits
- Olive oil
- White wine

Note: The `data-*` attributes seen in this example are called data attributes. They provide a way to store custom data in an HTML attribute so it can then be easily extracted and used. See How to use data attributes for more information.

Substring value attribute selectors

Attribute selectors in this class are also known as "RegExp-like selectors", because they offer flexible matching in a similar fashion to regular expression (but to be clear, these selectors are not true regular expression):

- `[attr^=val]` : This selector will select all elements with the attribute `attr` for which the value starts with `val`.
- `[attr$=val]` : This selector will select all elements with the attribute `attr` for which the value ends with `val`.

- `[attr*=val]` : This selector will select all elements with the attribute `attr` for which the value contains the substring `val`. (A substring is simply part of a string, e.g. "cat" is a substring in the string "caterpillar".)

> **Note**: There's also a very specific alternative version of the first option — `[attr|=val]`. This selector selects all elements with the attribute `attr` for which the value is exactly `val` or starts with `val-` (the dash here isn't a mistake). This was implemented specifically to match language codes, e.g. lang="en" or lang="en-US", and you probably won't use it very often.

Let's continue our previous example and add the following CSS rules:

```
/* Classic usage for language selection */
[lang|="fr"] {
  font-weight: bold;
}

/* All elements with the attribute "data-quantity", for which
   the value starts with "optional" */
[data-quantity^="optional"] {
  opacity: 0.5;
}

/* All elements with the attribute "data-quantity", for which
   the value ends with "kg" */
[data-quantity$="kg"] {
  font-weight: bold;
}

```

```
/* All elements with the attribute "data-vegetable" containing
   the substring "not spicy" are turned back to green */
[data-vegetable*="not spicy"] {
  color: green;
}
```

With those new rules, we will get this:

Ingredients for my recipe: ***Poulet basquaise***

- **Tomatoes**
- Onions
- Garlic
- Red pepper
- **Chicken**
- Bacon bits

Active learning: Styling soccer results

In this active learning, we'd like you to try your hand at adding attribute selectors to some rules to style a simple soccer results listing. There are three things to try to do here:

- The first three rules add a UK, German, and Spanish flag icon respectively to the left hand side of the list items. You need to fill in appropriate attribute selectors so that the teams are given their correct country flags, matched by language.

- Rules 4–6 add a background color to the list items to indicate whether the team has gone up in the league (green, `rgba(0,255,0,0.7)`), down (red, `rgba(255,0,0,0.5)`), or stayed in the same place (blue, `rgba(0,0,255,0.5)`.) Fill in the appropriate attribute selectors to match the correct colors to the correct teams, matched by the `inc`, `same` and `dec` strings that appear in the `data-perf` attribute values.
- Rules 7–8 make teams that are set to be promoted **bold**, and teams that are in danger of being relegated *italic and gray*. Fill in appropriate attribute selectors to match these styles to the correct teams, matched by the `pro` and `rel` strings that appear in the `data-perf` attribute values.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the *Show solution* button to see a potential answer.

HTML Input

```
<ol>
  <li lang="en-GB" data-perf="inc-pro">Manchester United</li>
  <li lang="es" data-perf="same-pro">Barcelona</li>
  <li lang="de" data-perf="dec">Bayern Munich</li>
  <li lang="es" data-perf="same">Real Madrid</li>
  <li lang="de" data-perf="inc-rel">Borussia Dortmund</li>
  <li lang="en-GB" data-perf="dec-rel">Southampton FC</li>
</ol>
```

CSS Input

```
li[] {
   background: url("http://mdn.github.io/learning-area/css/introduction-to-css/css-s
center no-repeat;
 }

li[] {
```

```
li[] {
   background: url("http://mdn.github.io/learning-area/css/introduction-to-css/css-s
center no-repeat;
 }
```

Output

1. Manchester United
2. Barcelona
3. Bayern Munich
4. Real Madrid
5. Borussia Dortmund
6. Southampton FC

Reset Show solution

See also

See the Attribute selectors reference page for more useful examples.

Coming up next

Next we will move things up a gear, looking at Pseudo-classes and pseudo-elements.

Previous | Overview: Introduction to CSS | Next

In this module