Attribute selectors

- Presence and value selectors
- Substring matching selectors
- Summary

As you know from your study of HTML, elements can have attributes that give further detail about the element being marked up. In CSS you can use attribute selectors to target elements with certain attributes. This lesson will show you how to use these very useful selectors.

Prerequisites:	HTML basics (study Basic HTML syntax), Basic CSS selectors.
Learning outcomes:	• The basic concept of attribute selectors. • Presence and value attribute selectors. • Substring matching attribute selectors.

Presence and value selectors

These selectors enable the selection of an element based on the presence of an attribute alone (for example `href`), or on various different matches against the value of the attribute.

Selector	Example	Description
`[attr]`	`a[title]`	Matches elements with an *attr* attribute (whose name is the value in square brackets).
`[attr=value]`	`a[href="https://example.com"]`	Matches elements with an *attr* attribute whose value is exactly *value* — the string inside the quotes.
`[attr~=value]`	`p[class~="special"]`	Matches elements with an *attr* attribute whose value is exactly *value*, or contains *value* in its (space-separated) list of values.

Selector	Example	Description
`[attr|=value]`	`div[lang|="zh"]`	Matches elements with an *attr* attribute whose value is exactly *value* or begins with *value* immediately followed by a hyphen.

In the example below you can see these selectors being used.

- By using `li[class]` we can match any list item with a class attribute. This matches all of the list items except the first one.
- `li[class="a"]` matches a selector with a class of `a`, but not a selector with a class of `a` with another space-separated class as part of the value. It selects the second list item.
- `li[class~="a"]` will match a class of `a` but also a value that contains the class of `a` as part of a whitespace-separated list. It selects the second and third list items.

```
<h1>Attribute presence and value selectors</h1>
<ul>
  <li>Item 1</li>
  <li class="a">Item 2</li>
  <li class="a b">Item 3</li>
  <li class="ab">Item 4</li>
</ul>
```

```
body {
  font-family: sans-serif;
}
li[class] {
  font-size: 120%;
}

li[class="a"] {
  background-color: yellow;
}

li[class~="a"] {
  color: red;
}
```

Try editing the above CSS to add a rule that selects only list items with a `class` attribute value of `ab`, and gives them a `white` text `color` and a `purple` `background-color`.

Substring matching selectors

These selectors allow for more advanced matching of substrings inside the value of your attribute. For example, if you had classes of `box-warning` and `box-error` and wanted to match everything that started with the string "box-", you could use `[class^="box-"]` to select them both (or `[class|="box"]` as described in section above).

Selector	Example	Description
`[attr^=value]`	`li[class^="box-"]`	Matches elements with an *attr* attribute, whose value begins with *value*.
`[attr$=value]`	`li[class$="-box"]`	Matches elements with an *attr* attribute whose value ends with *value* .
`[attr*=value]`	`li[class*="box"]`	Matches elements with an *attr* attribute whose value contains *value* anywhere within the string.

The next example shows usage of these selectors:

- `li[class^="a"]` matches any attribute value which starts with `a`, so matches the first two list items.
- `li[class$="a"]` matches any attribute value that ends with `a`, so matches the first and third list item.
- `li[class*="a"]` matches any attribute value where `a` appears anywhere in the string, so it matches all of our list items.

```
<h1>Attribute substring matching selectors</h1>
<ul>
  <li class="a">Item 1</li>
  <li class="ab">Item 2</li>
  <li class="bca">Item 3</li>
  <li class="bcabc">Item 4</li>
</ul>
```

```
body {
  font-family: sans-serif;
}
li[class^="a"] {
  font-size: 120%;
}
```

```
li[class$="a"] {
  background-color: yellow;
}

li[class*="a"] {
  color: red;
}
```

Try editing the above CSS to add a rule that selects only list items with a `class` attribute value that has `b` or `c` at the end of it, and gives them a `2px`-wide, `solid`, `black` `border`. You may have to use a selector list to solve this one.

Summary

Now that we are done with attribute selectors, you can continue on to the next article and read about pseudo-class and pseudo-element selectors.

□□□ #3
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ cocobook-sync □□□ 2026-11-04 17:32:53 UTC