mdn web docs

属性选择器

正如您通过学习 HTML 了解到的那样，元素可以具有属性，这些属性提供有关被标记元素的更多详细信息。在 CSS 中，您可以使用属性选择器来定位具有特定属性的元素。本课将向您展示如何使用这些非常有用的选择器。

先决条件：	基本的计算机知识、安装的基本软件、使用文件的基本知识、HTML 基础知识（学习 HTML 简介）以及 CSS 的工作原理（学习 CSS 第一步）。
客观的：	了解什么是属性选择器以及如何使用它们。

存在和价值选择器

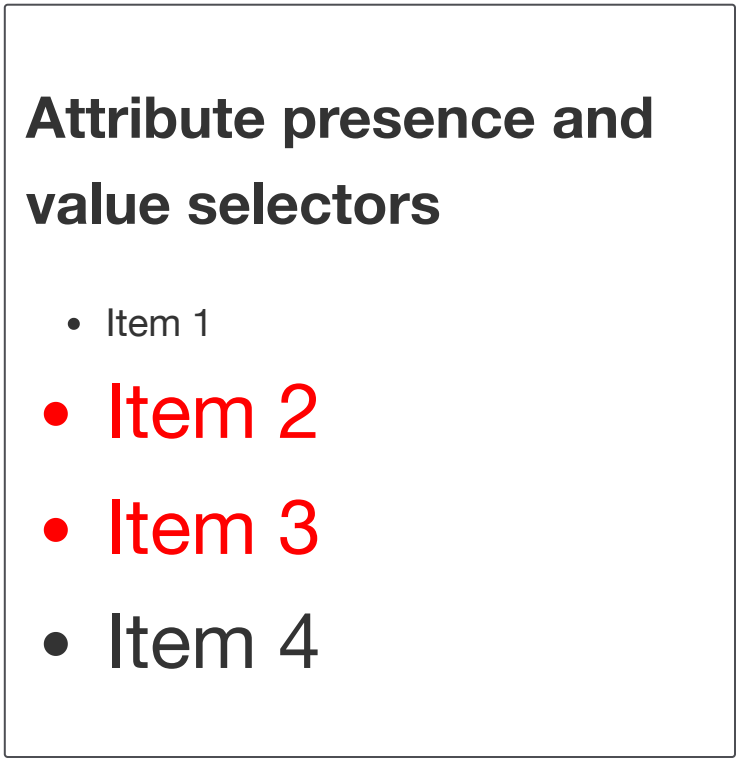
`href` 这些选择器可以仅根据属性的存在（例如）或根据与属性值的各种不同匹配项来选择元素。

选择器	例子	描述
`[attr]`	`a[title]`	*匹配具有attr* 属性的 元素（其名称是方括号中的值）。
`[attr=value]`	`a[href="https://example.com"]`	*匹配具有attr* 属性的 元素，其值恰好是 *value* —

选择器	例子	描述
		引号内的字符串。
`[attr~=value]`	`p[class~="special"]`	*匹配具有attr* 属性的元素，其值恰好是 *value* ，或者在其（空格分隔的）值列表中 包含*value* 。
`[attr|=value]`	`div[lang|="zh"]`	*匹配具有attr* 属性的 元素，该属性的值恰好是 *value*或以 *value*开头，紧跟一个连字符。

在下面的示例中，您可以看到正在使用这些选择器。

- 通过使用 `li[class]` 我们可以将任何列表项与类属性匹配。这匹配除第一个之外的所有列表项。
- `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.



Interactive editor

```
li[class] {
    font-size: 200%;
}

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

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

```
<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>
```

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

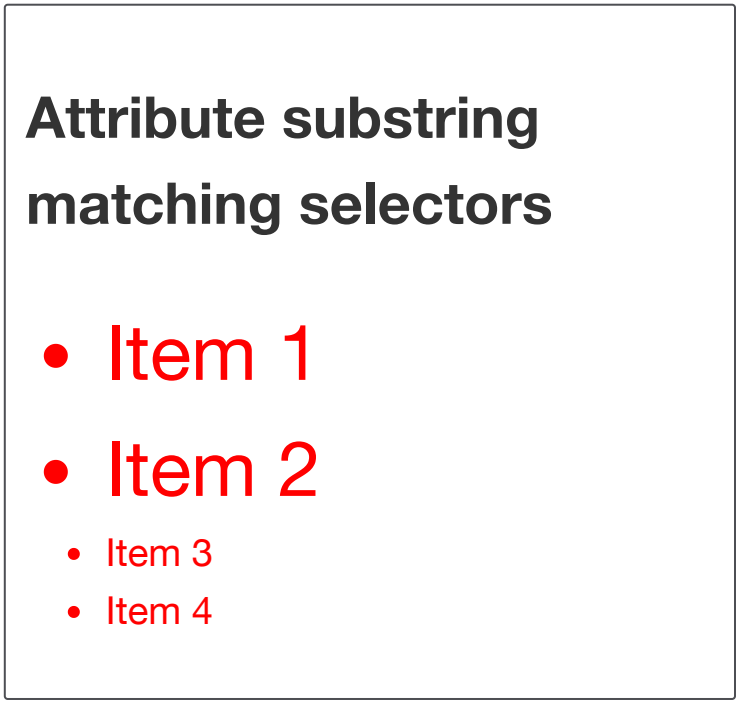
started with the string "box-", you could use `[class^="box-"]` to select them both (or `[class|="box"]` as described in section above).

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.

(Aside: It may help to note that `^` and `$` have long been used as *anchors* in so-called *regular expressions* to mean *begins with* and *ends with* respectively.)

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.



Interactive editor

```
li[class^="a"] {
    font-size: 200%;
}

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

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

```
<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>
```

Case-sensitivity

If you want to match attribute values case-insensitively you can use the value `i` before the closing bracket. This flag tells the browser to match ASCII characters case-insensitively. Without the flag the

values will be matched according to the case-sensitivity of the document language — in HTML's case it will be case sensitive.

In the example below, the first selector will match a value that begins with `a` — it only matches the first list item because the other two list items start with an uppercase A. The second selector uses the case-insensitive flag and so matches all of the list items.

Case-insensitivity

- Item 1
- Item 2
- Item 3

Interactive editor

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

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

```
<h1>Case-insensitivity</h1>
<ul>
    <li class="a">Item 1</li>
    <li class="A">Item 2</li>
    <li class="Ab">Item 3</li>
</ul>
```

Reset

Note: There is also a newer value `s`, which will force case-sensitive matching in contexts where matching is normally case-insensitive, however this is less well supported in browsers and isn't very useful in an HTML context.

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.

This page was last modified on Feb 23, 2023 by MDN contributors.