

此页面由社区从英文翻译而来。了解更多并加入 MDN Web Docs 社区。

使用选择器定位 DOM 元素

Selectors API 提供了通过与一组[选择器](#)匹配来轻松快速地从 DOM 检索 [Element](#) 节点的方法。这比以前的技术要快得多，例如，过去的技术需要在 JavaScript 代码中使用循环来定位你需要查找的特定项目。

NodeSelector 接口

此规范向实现 [Document](#)、[DocumentFragment](#) 或 [Element](#) 接口的任何对象添加了两种新方法：

[querySelector\(\)](#)

返回节点子树内与之相匹配的第一个 [Element](#) 节点。如果没有匹配的节点，则返回 `null`。

[querySelectorAll\(\)](#)

返回一个包含节点子树内所有与之相匹配的 [Element](#) 节点的 [NodeList](#)，如果没有匹配的节点，则返回一个空的 [NodeList](#)。

备注： 由 [querySelectorAll\(\)](#) 返回的 [NodeList](#) 不是动态实时的，这意味着在 DOM 中的更改不会在该列表中反映。这和其他 DOM 查询方法返回动态实时节点列表不一样。

你可以在 [Element.querySelector\(\)](#) 和 [Element.querySelectorAll\(\)](#) 方法的文档中找到更多示例和细节。

选择器

选择器方法接受若干[选择器](#)来决定应该返回什么元素或哪些元素。这包括[选择器列表](#)，所以你可以在一个查询中分组多个选择器。

为了保护用户的隐私，一些[伪类](#)不被支持或表现得不同。例如，[:visited](#) 将不返回任何匹配，[:link](#) 则被视为 [:any-link](#)。

只有元素可以被选择，故不支持[伪类](#)。

示例

要选择所有包含 `warning` 或 `note` 类的所有段落（`p`）元素，你可以这样做：

JS

```
const special = document.querySelectorAll("p.warning, p.note");
```

也可以通过 ID 来查询，例如：

JS

```
const el = document.querySelector("#main, #basic, #exclamation");
```

执行上述代码后，`el` 包含文档中的第一个元素，其 ID 是 `main`、`basic` 或 `exclamation` 之一。

`querySelector()` and `querySelectorAll()` 里可以使用任何 CSS 选择器。

参见

- [选择器规范](#)
- [CSS 选择器](#)
- [Element.querySelector\(\)](#)
- [Element.querySelectorAll\(\)](#)
- [Document.querySelector\(\)](#)
- [Document.querySelectorAll\(\)](#)



Help improve MDN

Was this page helpful to you?

Yes

No

[Learn how to contribute.](#)

This page was last modified on 2023年7月22日 by [MDN contributors](#).

