

O que é uma *NodeList*

Quando você usa:

```
const elementos = document.querySelectorAll("p");
```

Você está pedindo ao navegador:

“Me dê **todos os elementos** do HTML que **combinam** com esse **seletor CSS**.”

O resultado é uma **NodeList** — *uma coleção de nós (elementos HTML) que **podem ser percorridos, manipulados e estilizados via JavaScript**.*

Então... o que pode ser pesquisado?

Qualquer elemento HTML

Você pode selecionar elementos pelo nome da **tag**:

```
document.querySelectorAll("p");           // todos os <p>
document.querySelectorAll("div");         // todas as <div>
document.querySelectorAll("img");        // todas as <img>
```

Por classe

Use **.** (ponto) antes do nome da classe:

```
document.querySelectorAll(".botao");      // todos com class="botao"
document.querySelectorAll(".item");       // todos com class="item"
```

 *Dica:* um elemento pode ter várias classes:

```
<div class="card destaque"></div>
```

Você pode pesquisar só pelos com “.destaque”.

3 Por ID

IDs são únicos, mas você ainda pode buscá-los com #:

```
document.querySelectorAll("#menu"); // elemento com id="menu"
```

⚠ `querySelectorAll()` sempre retorna uma `NodeList`, mesmo que só haja **um único** elemento.

4 Por hierarquia (descendência)

Você pode combinar seletores como no CSS:

```
document.querySelectorAll("ul li"); // todos os <li> dentro de <ul>
document.querySelectorAll(".menu a"); // todos os <a> dentro de .menu
document.querySelectorAll("section article"); // todos os <article> dentro de <section>
```

5 Por atributos

Você pode pesquisar elementos com base em atributos HTML:

```
document.querySelectorAll("[src]"); // todos com atributo src
document.querySelectorAll("[href^='https']"); // href começa com "https"
document.querySelectorAll("[data-role='admin']"); // data-role="admin"
```

💡 Esses seletores são poderosos para encontrar *botões, links, imagens, inputs, etc.*

6 Por pseudoclasses (como no CSS!)

Você pode usar seletores avançados:

```
document.querySelectorAll("li:first-child"); // primeiro item de
cada lista
document.querySelectorAll("li:nth-child(2)"); // segundo item
document.querySelectorAll("input:checked"); // inputs marcados
(checkbox, radio)
document.querySelectorAll("a:hover"); // (funciona apenas em
CSS)
```

⚠ Algumas pseudoclasses dinâmicas como `:hover` não retornam resultados em JS.

7 Combinações e filtros complexos

Você pode combinar tudo:

```
document.querySelectorAll("div.card[data-ativo='true'] img");
```

➡ Isso retorna **todas as imagens ()** que estão dentro de uma `<div>` com classe `card` e atributo `data-ativo="true"`.

Resumo rápido

Tipo de busca	Exemplo	O que retorna
Por tag	"p"	Todos os <p>
Por classe	".item"	Todos os elementos com <code>class="item"</code>
Por id	"#menu"	O elemento com <code>id="menu"</code>
Por hierarquia	"ul li"	Todos os dentro de
Por atributo	"[data-user]"	Todos com o atributo <code>data-user</code>
Por combinação	"div.card img"	Todas as dentro de <code>.card</code>

✂ Exemplo prático

HTML:

```
<ul class="lista">
  <li data-prioridade="alta">Estudar JS</li>
  <li data-prioridade="media">Fazer exercícios</li>
  <li data-prioridade="baixa">Ver séries</li>
</ul>
```

JS:

```
const tarefasImportantes =
document.querySelectorAll("[data-prioridade='alta']");
tarefasImportantes.forEach(tarefa => {
  tarefa.style.color = "red";
});
```

➡ Isso pinta de vermelho apenas os `` com prioridade “alta”.