

What is the
querySelectorAll()
method?

in JavaScript DOM

dc●de

JS

ISAAC SOLER FORES

querySelectorAll()

querySelectorAll es una función en JavaScript que se utiliza para seleccionar y devolver una `NodeList` (una colección de nodos) de elementos que coinciden con un selector CSS dado. Esta función permite realizar consultas sobre el DOM (Document Object Model) y recuperar los elementos que cumplen con las condiciones especificadas en el selector.

Algunas características clave de `querySelectorAll` incluyen:

- **Sintaxis de Selector CSS:** La función toma como argumento un selector CSS válido que describe los elementos que se deben seleccionar. Puede ser un id, una clase, un tipo de elemento o combinaciones de ellos.
- **Retorno de NodeList:** `querySelectorAll` devuelve una `NodeList` que contiene todos los elementos que coinciden con el selector proporcionado. Aunque la `NodeList` se asemeja a un array, no es exactamente un array, pero se puede iterar a través de ella usando bucles o convertirla en un array con métodos como `Array.from()`.

Aquí hay un ejemplo básico:

```
javascript

// Selecciona todos los elementos con la clase "ejemplo"
var elementos = document.querySelectorAll(".ejemplo");

// Itera sobre la NodeList
elementos.forEach(function(elemento) {
  console.log(elemento.textContent);
});
```

En este ejemplo, `document.querySelectorAll(".ejemplo")` selecciona todos los elementos con la clase "ejemplo" y devuelve una `NodeList` que luego se itera para imprimir el contenido de cada elemento.

`querySelectorAll` es útil cuando se necesitan seleccionar múltiples elementos que cumplen con ciertos criterios sin la necesidad de iterar manualmente sobre ellos. Es especialmente común en el contexto de manipulación del DOM con JavaScript.

A continuación se muestran todos los selectores:

Selector	Example	Example description
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>.class1.class2</u>	.name1.name2	Selects all elements with both <i>name1</i> and <i>name2</i> set within its class attribute
<u>.class1 .class2</u>	.name1 .name2	Selects all elements with <i>name2</i> that is a descendant of an element with <i>name1</i>
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element.class</u>	p.intro	Selects all <p> elements with class="intro"
<u>element,element</u>	div, p	Selects all <div> elements and all <p> elements
<u>element element</u>	div p	Selects all <p> elements inside <div> elements
<u>element>element</u>	div > p	Selects all <p> elements where the parent is a <div> element

<u>element+element</u>	div + p	Selects the first <p> element that is placed immediately after <div> elements
<u>element1~element2</u>	p ~ ul	Selects every element that is preceded by a <p> element
<u>[attribute]</u>	[target]	Selects all elements with a target attribute
<u>[attribute=value]</u>	[target="_blank"]	Selects all elements with target="_blank"
<u>[attribute~=value]</u>	[title~="flower"]	Selects all elements with a title attribute containing the word "flower"
<u>[attribute =value]</u>	[lang="en"]	Selects all elements with a lang attribute value equal to "en" or starting with "en-"
<u>[attribute^=value]</u>	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
<u>[attribute\$=value]</u>	a[href\$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
<u>[attribute*=value]</u>	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"
<u>:active</u>	a:active	Selects the active link
<u>::after</u>	p::after	Insert something after the content of each <p> element
<u>::before</u>	p::before	Insert something before the content of each <p> element

<u>:checked</u>	input:checked	Selects every checked <input> element
<u>:default</u>	input:default	Selects the default <input> element
<u>:disabled</u>	input:disabled	Selects every disabled <input> element
<u>:empty</u>	p:empty	Selects every <p> element that has no children (including text nodes)
<u>:enabled</u>	input:enabled	Selects every enabled <input> element
<u>:first-child</u>	p:first-child	Selects every <p> element that is the first child of its parent
<u>::first-letter</u>	p::first-letter	Selects the first letter of every <p> element
<u>::first-line</u>	p::first-line	Selects the first line of every <p> element
<u>:first-of-type</u>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<u>:focus</u>	input:focus	Selects the input element which has focus
<u>:fullscreen</u>	:fullscreen	Selects the element that is in full-screen mode
<u>:hover</u>	a: hover	Selects links on mouse over
<u>:in-range</u>	input:in-range	Selects input elements with a value within a specified range

<u>:indeterminate</u>	input:indeterminate	Selects input elements that are in an indeterminate state
<u>:invalid</u>	input:invalid	Selects all input elements with an invalid value
<u>:lang(<i>language</i>)</u>	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)
<u>:last-child</u>	p:last-child	Selects every <p> element that is the last child of its parent
<u>:last-of-type</u>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<u>:link</u>	a:link	Selects all unvisited links
<u>::marker</u>	::marker	Selects the markers of list items
<u>:not(<i>selector</i>)</u>	:not(p)	Selects every element that is not a <p> element
<u>:nth-child(<i>n</i>)</u>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<u>:nth-last-child(<i>n</i>)</u>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<u>:nth-last-of-type(<i>n</i>)</u>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child

<u>:nth-of-type(<i>n</i>)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<u>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<u>:only-child</u>	p:only-child	Selects every <p> element that is the only child of its parent
<u>:optional</u>	input:optional	Selects input elements with no "required" attribute
<u>:out-of-range</u>	input:out-of-range	Selects input elements with a value outside a specified range
<u>::placeholder</u>	input::placeholder	Selects input elements with the "placeholder" attribute specified
<u>:read-only</u>	input:read-only	Selects input elements with the "readonly" attribute specified
<u>:read-write</u>	input:read-write	Selects input elements with the "readonly" attribute NOT specified
<u>:required</u>	input:required	Selects input elements with the "required" attribute specified
<u>:root</u>	:root	Selects the document's root element
<u>::selection</u>	::selection	Selects the portion of an element that is selected by a user
<u>:target</u>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)

<u>:valid</u>	input:valid	Selects all input elements with a valid value
<u>:visited</u>	a:visited	Selects all visited links

EJERCICIOS

Utilizando querySelectorAll realiza los siguientes cambios en la web que se adjunta.

1. Seleccionar todos los párrafos dentro de un elemento <div> y cambiar su color de fondo a gris
2. Seleccionar todos los elementos con clase "rojo" y cambiar su texto a mayúsculas
3. Seleccionar todos los elementos dentro del elemento con id "tabla" y cambiar su color de texto a azul
4. Seleccionar todos los elementos con clase "verde" y agregar una clase "fondogris"
5. Seleccionar el primer elemento <h1> en la página y cambiar su color de texto a verde
6. Seleccionar todos los elementos con clase "azul" y agregar una clase "fondogris"
7. Seleccionar todos los elementos <label> y cambiar su estilo a cursiva (italic)
8. Seleccionar todos los elementos <input> de tipo "checkbox" y cambiar su fondo a amarillo
9. Seleccionar todos los elementos <h2> dentro de un <div> y agregar una clase "resaltado"
10. Seleccionar todos los elementos con id que contiene la letra "o" y cambiar su color de texto a morado
11. Seleccionar todos los elementos y y cambiar su color de texto a verde
12. Seleccionar todos los elementos <div> con id que comienza con "div" y cambiar su fondo a cyan
13. Seleccionar todos los elementos <p> que contienen la palabra "párrafo" y cambiar su tamaño de fuente a 18px
14. Seleccionar el último elemento dentro del elemento con id "tabla" y cambiar su estilo a negrita
15. Selecciona todos los vacíos y ponles un fondo verde.
16. Selecciona los asientos en negrita y aplícales el color rojo
17. Selecciona todos los <h1> excepto los que tienen la clase rojo y ponles un fondo amarillo