

[Elementos HTML DOM de JavaScript](#)

En la Tarea0 hicimos un paseo por los 6 apartados de la imagen:



Este resumen se centra en el apartado señalado (**acceso a los elementos del DOM**) y encontramos que podemos clasificar los mecanismos de acceso a dichos elementos del DOM de la siguiente forma:

1. Acceso a través de propiedad

Existen elementos que pueden ser accedidos directamente a través de las propiedades del **objeto document**. Estos elementos se caracterizan por ser únicos dentro del documento, lo que permite diferenciarlos de forma inequívoca.

Ejemplo:

let cabecera = document.head // permite acceder al elemento head del documento

2. Acceso a través de colección:

A través de acceso a una colección (*HTMLCollection*). Utilizado para elementos que no tienen por qué ser únicos. El objeto document ofrece propiedades que devuelven una colección ordenada de esos elementos, como *forms*, *links*, *images*, *scripts*, *styleSheets*.

Ejemplo:

```
<!DOCTYPE html>
<html>
<body>

<h2>Finding HTML Elements Using document.forms</h2>

<form action="">
First name: <input type="text" name="fname" value="Donald">
<input type="submit" value="Submit">
</form>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
"Number of forms: " + document.forms.length;
</script>

</body>
</html>
```

Finding HTML Elements Using document.forms

First name:

Number of forms: 1

3. Recorrido de nodos: (lo veremos más adelante...)

Explorando el árbol de nodos. Utilizando las propiedades *childNodes*, *firstChild*, *lastChild*, ... una serie de propiedades procedentes de la interfaz nodo, con las que es posible explorar los distintos elementos.

4. Métodos de búsqueda

Los métodos de búsqueda ofrecidos por el **objeto document** permiten utilizar como criterio de búsqueda tanto *atributos del elemento* como *selectores CSS del elemento* en cuestión.

- Si empleamos **atributos** de los elementos disponemos de los métodos *getElementById()*, *getElementsByName()*, *getElementsByTagName()* y *getElementsByClassName()*, con los que podemos buscar:
 - el elemento cuyo id es uno concreto, *getElementById()*
 - los elementos que comparten un mismo valor del atributo name, *getElementsByName()* *(En XHTML no es muy útil puesto que para la mayor parte de los elementos el uso de name como identificador ha sido depreciado en favor de id, pero es bueno saber que forma parte del arsenal a nuestra disposición a la hora de trabajar con elementos que pueden compartir name pero no, obviamente, id, como son los radiobotones y las casillas de un formulario.)* Nosotros en el curso no lo emplearemos
 - los elementos html de un tipo concreto, *getElementsByTagName()* y
 - los elementos que pertenecen a una clase determinada, *getElementsByClassName()*
- Si el criterio de búsqueda son **selectores** CSS, se ofrecen los métodos *querySelector* y *querySelectorAll*, el primero devuelve el primer elemento que coincide con el selector, y el segundo devuelve un *nodeList* con todos los elementos que coinciden con el selector especificado.