

7. BOM y DOM

E. DOM

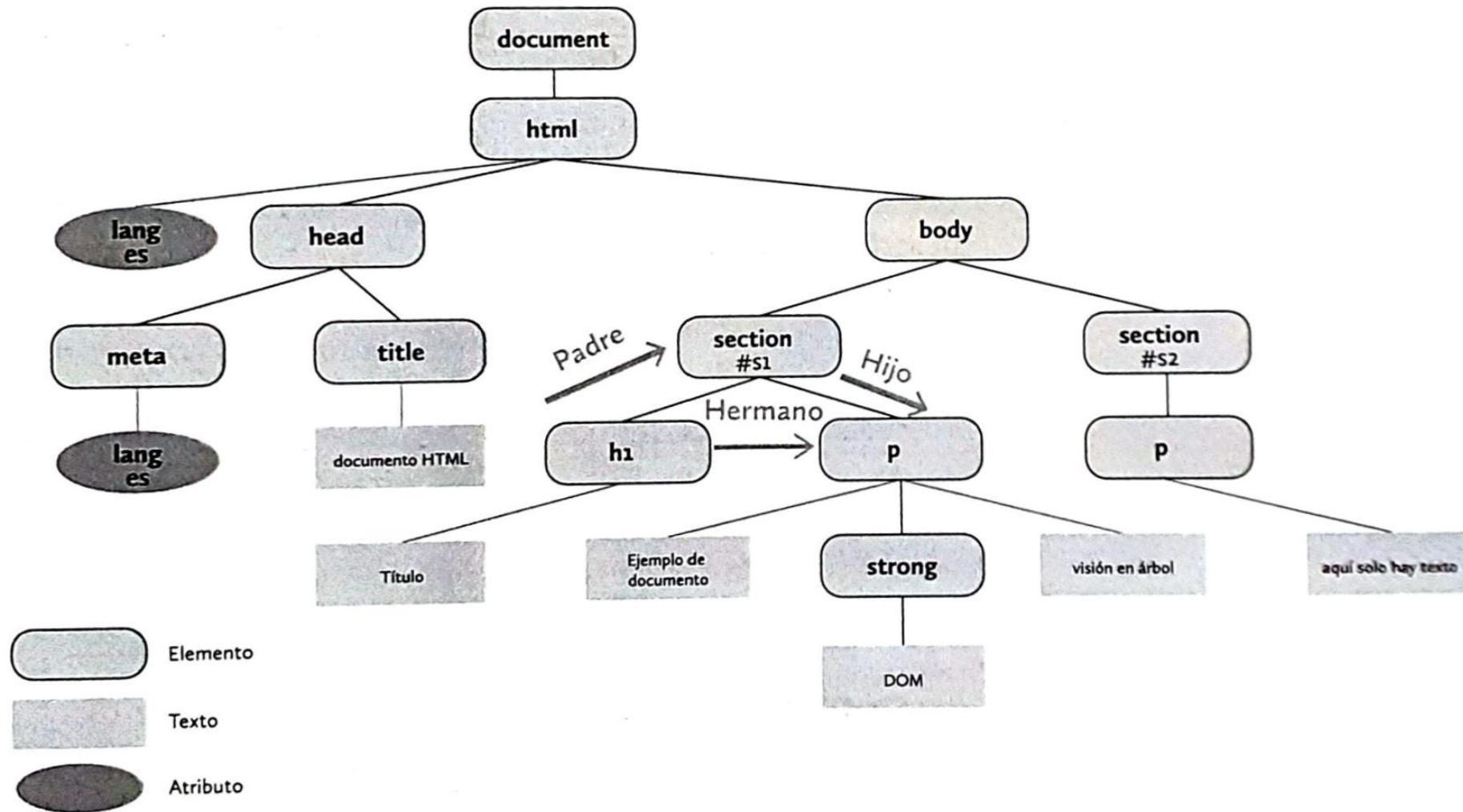
1. ¿Qué es el DOM?

- JavaScript se ideó para que fuera un lenguaje con capacidad de manipular el contenido y apariencia de una página web de forma dinámica. Para hacer esto posible, los elementos de la página web se tienen que poder manipular como una serie de objetos relacionados.
- Así apareció el término DOM, Document Objects Model (Modelo de Objetos del Documento) que define una página web como una estructura organizada de objetos que forman un árbol de recorrido.

1. ¿Qué es el DOM?

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Documento HTML</title>
</head>
<body>
  <section id="s1">
    <h1>Título</h1>
    <p>Ejemplo de documento <strong>DOM</strong>Visión en árbol</p>
  </section>
  <section id="s2">
    <p>Aquí sólo hay texto</p>
  </section>
</body>
</html>
```

1. ¿Qué es el DOM?



2. EL OBJETO DOCUMENT

- La forma de poder acceder a los objetos del documento en JavaScript es a través de un objeto llamado precisamente document.
- Este objeto es hijo del objeto window.
- Podemos indicar simplemente la palabra document para acceder a este objeto, aunque su nombre completo es window.document.
- Los diferentes métodos y propiedades del objeto document nos van a permitir acceder a los elementos del documento para su manipulación.

3. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR IDENTIFICADOR

- Se trata de seleccionar un nodo del documento por el valor del atributo id El método que lo permite es **getElementById** que recibe un **string** con el valor buscado.
- Retorna como resultado el elemento en si o bien el valor null si ese identificador no se encuentra en el elemento.
- Partimos del ejemplo anterior.

```
<section id="s1">
  <h1>Título</h1>
  <p>Ejemplo de documento <strong>DOM</strong>Visión en árbol</p>
</section>
```

3. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR IDENTIFICADOR

- Si queremos obtener el valor de la sección con id=s1 debemos utilizar el método descrito **document.getElementById** en nuestro fichero .js

```
let seccion1=document.getElementById("s1");  
console.log(seccion1);
```

- Obtendremos como resultado,
- en la consola:

```
▼ <section id="s1">  
  <h1>Título</h1>  
  ▼ <p>  
    "Ejemplo de documento "  
    <strong>DOM</strong>  
    "Visión en árbol"  
  </p>  
</section>
```

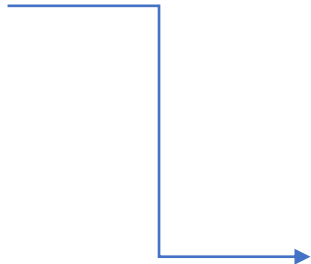
3. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR IDENTIFICADOR

- Ejemplo de uso.
- Para poder usar el ejemplo vamos a ver un evento que aun no hemos visto.
- El evento irá asociado a un botón (input type button) y es el evento **onclick** (que es cuando hacemos clic en él). El input type button quedará de la siguiente forma:
 - `<input type="button" value="calcula" onclick=" javascript: función javascript o código ">`

3. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR IDENTIFICADOR

- El fichero .html llevará este código

```
<body>  
  <input type="text" name="campo" id="campo">  
  <input type="button" onclick="javascript:opera()" value="Calcula tabla">  
  <p id="solucion"></p>  
  
  <script type="text/javascript" src="ejer.js"></script>  
</body>
```



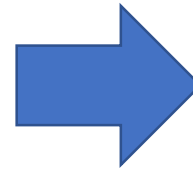
Calcula tabla

3. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR IDENTIFICADOR

- El fichero ejer.js llevará este:

```
function opera() {  
let campo=document.getElementById("campo").value;  
for(var i=1;i<=10;i++) {  
    document.write("<hr>");  
    document.write(i+" x "+ campo + " = "+(i*campo)+"<br>");  
}  
document.write("<input type='button' onclick='javascript:recarga()' value='volver a hacer'>");  
}
```

```
function recarga() {  
    location.reload();  
}
```



1 x 5 = 5

2 x 5 = 10

3 x 5 = 15

4 x 5 = 20

5 x 5 = 25

6 x 5 = 30

7 x 5 = 35

8 x 5 = 40

9 x 5 = 45

10 x 5 = 50

3. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR IDENTIFICADOR

- Podemos mejorar mucho el ejemplo anterior, haciendo uso de un `<div>` y del método `innerHTML`.
- Nos queda el fichero `.html` así:

```
<body>
  <input type="text" name="campo" id="campo">
  <input type="button" onclick="javascript:opera()" value="Calcula tabla">
  <p id="solucion"></p>

  <script type="text/javascript" src="ejer.js"></script>
</body>
```

3. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR IDENTIFICADOR

- Y el fichero ejer.js de la siguiente forma:

```
function opera() {  
  let campo=document.getElementById("campo").value;  
  var salida="Solucion:<hr> ";  
  for(var i=1;i<=10;i++) {  
    salida=salida+i+" x "+ campo + " = "+(i*campo)+"<br>";  
  }  
  let solucion=document.getElementById("solucion");  
  solucion.innerHTML=salida;  
  solucion.innerHTML+="}  
  
function recarga() {  
  location.reload();  
}
```

3. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR IDENTIFICADOR

- Ahora el resultado es mucho más elegante:

Solucion:

1 x 4 = 4
2 x 4 = 8
3 x 4 = 12
4 x 4 = 16
5 x 4 = 20
6 x 4 = 24
7 x 4 = 28
8 x 4 = 32
9 x 4 = 36
10 x 4 = 40

4. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR CLASE

- La selección por etiqueta permite seleccionar todos los elementos que tengan ese nombre de etiqueta. El método que lo consigue es **getElementsByTagName**.
- En este caso se devuelve una lista de elementos, es decir, no selecciona un elemento (como ocurría con getElementById), sino todos los elementos que usen esa clase.
- Vamos a cambiar el ejemplo por otro distinto.

4. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR CLASE

- En este caso tenemos una lista desordenada (ul) con una serie de elementos (li)

```
<body>
  <ul>
    <li class="verdura">Cebolla</li>
    <li class="carne">Pollo</li>
    <li class="verdura">Ajo</li>
    <li class="carne">Tenera</li>
  </ul>
  <input type="button" value="cambiar verduras" onclick="javascript:cambiaverdura()">
  <script type="text/javascript" src="ejer.js"></script>
</body>
```

4. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR CLASE

- Ahora nuestro código del fichero ejer.js. En este caso nos vamos a servir de la propiedad style que permite cambiar el aspecto de la pagina

```
function cambiaverdura() {  
    let verduras=document.getElementsByClassName("verdura");  
  
    for(let elemento of verduras) {  
        elemento.style.backgroundColor="green";  
    }  
}
```


4. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR CLASE

- Nuestra web quedará así:

- Cebolla
- Pollo
- Ajo
- Tenera

cambiar verduras

- Al hacer clic en cambiar verdura

- Cebolla
- Pollo
- Ajo
- Tenera

cambiar verduras

4. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR CLASE

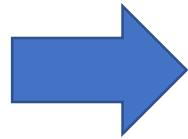
- Haciendo unas pequeñas modificaciones en el fichero .js podemos hacer que se seleccione o no si hacemos clic en el botón.

```
var activado=0;
function cambiaverdura() {
    let verduras=document.getElementsByClassName("verdura");
    if (activado==0) {
        for(let elemento of verduras) {
            elemento.style.backgroundColor="green";    }
        activado=1;
    } else {
        for(let elemento of verduras) {
            elemento.style.backgroundColor="transparent"; }
        activado=0;
    }
}
```

4. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR CLASE

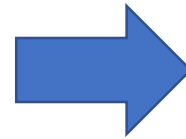
- Cebolla
- Pollo
- Ajo
- Tenera

cambiar verduras



- Cebolla
- Pollo
- Ajo
- Tenera

cambiar verduras



- Cebolla
- Pollo
- Ajo
- Tenera

cambiar verduras

5. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR ETIQUETA

- La selección por etiqueta permite seleccionar todos los elementos que tengan ese nombre de etiqueta.
- El método que lo consigue es **getElementsByTagName**. En este caso se devuelve una lista de elementos, con todo los elementos que usen esa etiqueta.

5. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR ETIQUETA

- Vamos a realizar otro ejemplo. Partimos de un fichero HTML como este:

```
<body>
  <p> Primer texto</p>
  <p> Segundo texto</p>
  <p> Tercer texto</p>

  <input type="button" value="cambiar p" onclick="javascript:cambiap()">
  <div id="salida"> </div>
  <script type="text/javascript" src="ejer.js"></script>
</body>
```

Primer texto

Segundo texto

Tercer texto

cambiar p

5. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR ETIQUETA

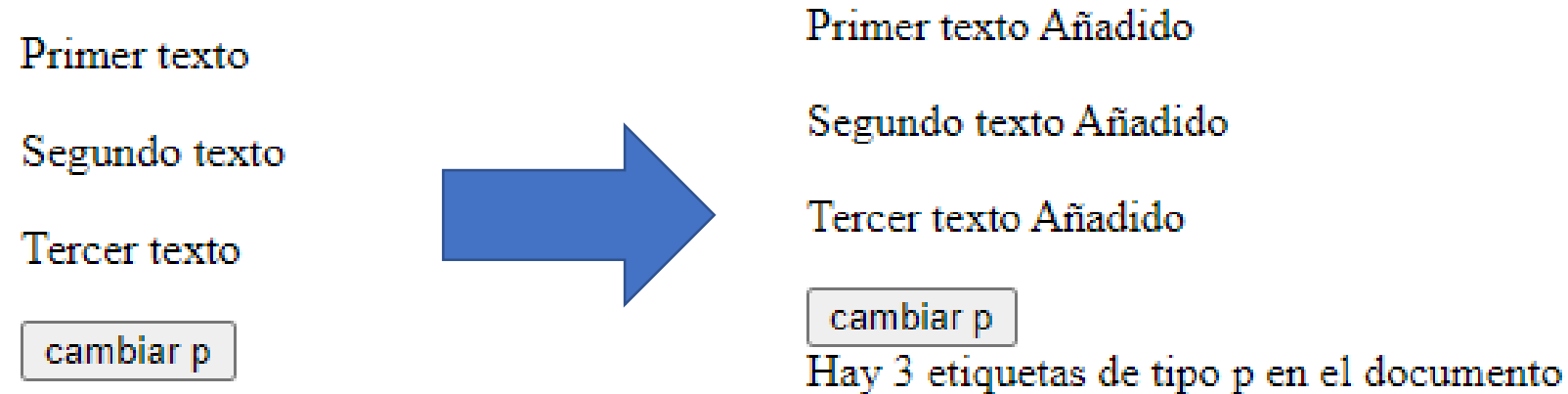
- El código del fichero js seria el siguiente:

```
function cambiap(){  
var arrp=document.getElementsByTagName("p");  
var salida=document.getElementById("salida");  
salida.innerHTML="Hay "+arrp.length+" etiquetas de tipo p en el documento";  
for (let elemento of arrp) {  
    elemento.insertAdjacentHTML("beforeend"," Añadido ");  
}  
}
```

- Básicamente lo que hace es poner en la etiqueta DIV el mensaje “Hay n etiquetas y añade al final de cada p el mensaje Añadido.

5. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR ETIQUETA

- Tenemos entonces:



5. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR ETIQUETA

- Posibles usos: En un formulario de alta, cuando tenemos que repetir la contraseña 2 veces y no coinciden dichas contraseñas.

6. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR NOMBRE

- El método JavaScript **getElementsByName()** es el que permite seleccionar un elemento por su nombre.
- Ejemplo. Código html:

```
<body>
  <p>Evalúe el servicio:</p>
  <p>
    <input type="radio" name="rate" value="muy pobre"> Muy pobre
    <input type="radio" name="rate" value="pobre"> Pobre
    <input type="radio" name="rate" value="ok"> OK
    <input type="radio" name="rate" value="bueno"> bueno
    <input type="radio" name="rate" value="muy bueno"> Muy bueno
  </p>
  <p>
    <button id="btnRate" onclick="javascript:procesarradio()">Enviar</button>
  </p>
  <script type="text/javascript" src="index.js"></script>
</body>
```

6. SELECCIÓN DE ELEMENTOS DEL DOM: SELECCIÓN POR NOMBRE

- Nuestro código .js mostrará un alert con la opción seleccionada

```
function procesarradio() {  
    var radio=document.getElementsByName("rate");  
    for (let elemento of radio) {  
        if (elemento.checked) {  
            alert("Tu has valorado: "+elemento.value);  
        }  
    }  
}
```

7. UN ÚNICO MÉTODO PARA GOBERNARLOS A TODOS. QUERYSELECTOR (ALL)



7. QUERYSELECTOR

- Este método reemplaza a todos los que hemos visto hasta el momento, e incluso incorpora funcionalidades que no se podrían hacer con los métodos anteriores.
- En el documento adjunto “querySelector.pdf” tendremos una tabla donde vamos a presentar los métodos vistos hasta ahora con su equivalente en querySelector y un ejemplo ilustrativo.