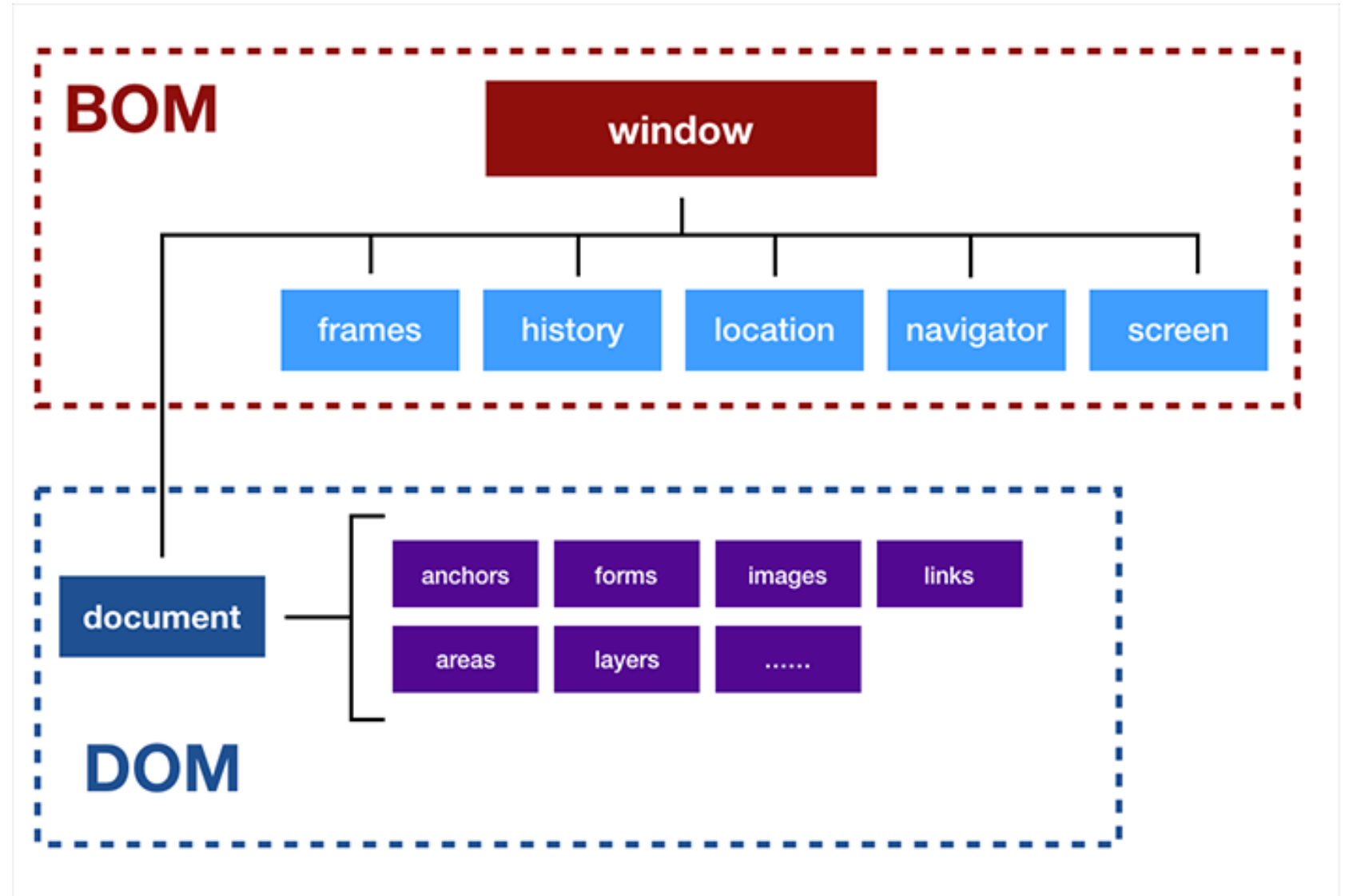


DOM y formularios

Laura Isabel López Naves

DOM (Document Object Model) y BOM (Browser Object Model)

- El DOM y el BOM definen la estructura de objetos en el navegador:



Ejemplos de uso de objetos BOM

- Abrir una nueva ventana y carga la página “pag.html”:

v=window.open("pag.html","nom","toolbar=no");

- Cierra la ventana: **v.close();**
- Carga la página cuya URL se especifica:

window.location="https://www.miurl.org";

- Navega a la URL anterior:

window.history.back();

- Define el valor que aparece en la barra de estado:

window.status="Página de prueba";

- Imprime el documento contenido en la ventana:

window.print();

Nota: El BOM es muy dependiente del tipo de navegador

Ejemplos de usos de objetos DOM

- El objeto **document** representa la página web actual. Su método `write()` permite escribir código html en el documento.

Ejemplo: `document.write("Prueba");`

← Escribe antes de cualquier código html del documento -> No se recomienda.

- Algunos ejemplos de propiedades de `document` son:
 - **body**: la etiqueta `<body>`. Ejemplo: `document.body.style.background="red";`
 - **links[]**: array con los enlaces. Ejemplo: `alert(document.links[1]);`
 - **images[]**: array con las imágenes (``) del documento.
(La propiedad `src` de un objeto `image` indica su origen). Ejemplo:

```
<script>
...function f()-{
...    document.images["uno"].src = "1.jpg";
...    document.images["dos"].src = "2.jpg";
...}
</script>
```

```
<body onload="f()">
... <img name="uno">
... <img name="dos">
</body>
```

Nodos del DOM

- Las etiquetas de una página web (o document) forman un conjunto de nodos ordenados jerárquicamente. Ejemplo:

```
<body>
  <div id="d1">Lorem Ipsum is simply dummy text of the printing
  <div id="d2">Lorem Ipsum is simply dummy text of the printing
  <form name="fo" action="#">
    <input type="submit" value="enviar">
  </form>
  <img name="uno">
  <img name="dos">
</body>
```

En este ejemplo los nodos “div” contienen en su interior un nodo de tipo “**text**” donde se almacena el texto de ejemplo: “Lorem Ipsum...”

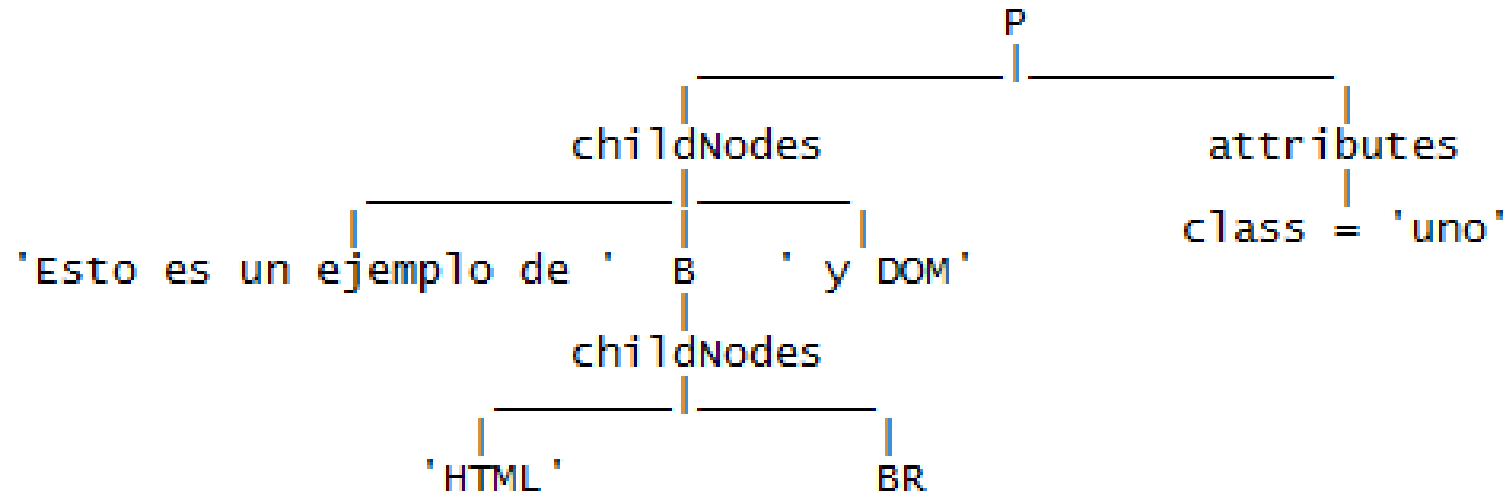
- “document” contiene un nodo “body” que contiene dos nodos “div”, un nodo “form” y dos nodos “img”.
- Existen varios tipos de nodos, los más comunes son: **document** (la página), **element** (representa una etiqueta o tag), **attr** (representa un atributo) y **text** (texto dentro de una etiqueta).

Nodos del DOM

El siguiente código:

`<p class='uno'>Esto es un ejemplo de HTML
 y DOM</p>`

se transforma en el árbol DOM en:



Cada parte del árbol es un nodo (u objeto javascript). Los nodos “P”, “B” y “BR” son de tipo **element**.

“**childNodes**” y “**attributes**” son colecciones, el par **class=‘uno’** es un nodo de tipo atributo y las cadenas de caracteres son nodos de tipo **text**.

Acceso a los nodos

- Se puede utilizar el camino jerárquico completo para acceder a un nodo. Por ejemplo:

document.body.childNodes[0].childNodes[0].nodeValue

document.childNodes[1].childNodes[1].childNodes[0].childNodes[0].nodeValue

pero es un método es engorroso y difícil de leer.

- Es mejor utilizar los siguientes métodos para localizar nodos (por su versatilidad los más utilizados son “**querySelector**” y “**querySelectorAll**”):

Method	Searches by...	Can call on an element?	Return
querySelector	CSS-selector	Yes	One element
querySelectorAll	CSS-selector	Yes	An array
getElementById	id	No	One element
getElementsByName	name	No	An array
getElementsByTagName	tag	Yes	An array
getElementsByClassName	class	Yes	An array

Ejemplos de acceso a nodos: `getElementsByTagName()`

- **`document.getElementsByTagName("img")`**: devuelve todos los nodos del documento cuya etiqueta es "img".
- **`parrafo.getElementsByTagName("div")`**: devuelve todos los nodos contenidos en el nodo "parrafo" (debe ser un nodo tipo `Element`) cuya etiqueta sea "div".

- **`const ed=document.getElementsByTagName("div");`**
`for (var i=0;i<ed.length;i++)`
`alert(ed[i].childNodes[0].nodeValue);`

Esta línea equivale a:
`alert(ed[i].firstChild.nodeValue);`

Visualiza el texto de todas las etiquetas "div" del documento.

- **`for (const d of document.getElementsByTagName("div"))`**
`console.log(d.childNodes[0].nodeValue);`

Visualiza el texto de todas las etiquetas "div" del documento (igual que el anterior).

- A tener en cuenta:
 - **`childNodes`** es colección de solo lectura de nodos accesible por índice (no es un array, es un `NodeList`).
 - Cada nodo tiene tres propiedades fundamentales:
 - **`nodeName`** (en los nodos de tipo `Element` es el nombre de la etiqueta)
 - **`nodeType`** (1=`Element`, 2=`Attribute`, 3=`Text`, etc...)
 - **`nodeValue`** (en los nodos tipo `Text` contiene el string de texto).

Ejemplos: `getElementsByName()` y `getElementById()`

- **`document.getElementsByName("uno")`** : devuelve todos los nodos del documento con el atributo `name="uno"`. Normalmente el atributo `"name"` se aplica a etiquetas `"form"`, `"input"`, `"img"` y `"a"`. Hay etiquetas que no lo admiten.
- **`document.getElementById("id")`** : devuelve un solo elemento que contenga el `"id"` indicado.

```
<form name="fo" action="#">
  <input name="nombre" type="text" value="Pepe" />
  <input type="submit" value="enviar">
</form>
<div id="d1">Lorem Ipsum is simply</div>
<script>
  const e1=document.getElementsByName("nombre");
  console.log(`${e1[0].nodeName} ${e1[0].nodeType}`); // INPUT 1
  console.log(`${e1[0].type} ${e1[0].value}`); // text Pepe
  const e2=document.getElementById("d1");
  console.log(`${e2.firstChild.nodeName} ${e2.firstChild.nodeType}`); // #text 3
  console.log(`${e2.firstChild.nodeValue}`); // Lorem Ipsum is simply
</script>
```

- Observar que `getElementsByName()` devuelve una colección de nodos pero `getElementById()` solo devuelve un nodo.
- Dentro de cada nodo algunos atributos se convierten en propiedades del objeto, por ejemplo `"type"` o `"value"`.
- Todo nodo tiene una colección de subnodos denominada `"childNodes"` y también las propiedades `"firstChild"`, `"lastChild"`, `"parentNode"`, etc.

Ejemplos: `querySelector()` y `querySelectorAll()`

- **`querySelector("selector")`**: devuelve el nodo que contenga el selector CSS.
- **`querySelectorAll("selector")`**: devuelve una colección de nodos que cumplen con el selector CSS indicado.

Se pueden utilizar otros selectores más complicados como:

- **`"div.example"`**: "div" de la clase "example".
- **`"div>img"`**: nodo "img" contenido en un "div".
- **`"[href='value']"`**: nodo que contenga el atributo "href" con el valor "value".
- **`"div p"`**: nodos "div" y "p".
- Etc.

```
<div id="d1">Lorem Ipsum is simply </div>
<ul>
  <li>uno</li>
  <li class="a">dos</li>
  <li class="a">tres</li>
</ul>
```

```
let d1=document.querySelectorAll("*"); // todos los nodos
console.log(`${d1[0].nodeName}`); // HTML
console.log(`${d1[1].nodeName}`); // HEAD
let d2=document.querySelector("#d1"); // nodo con el id "d1"
console.log(`${d2.nodeName}`); // DIV
let d3=document.querySelectorAll("div"); // todos los nodos "div"
console.log(`${d3[0].id}`); // d1
let d4=document.querySelectorAll(".a"); // anodos class="a":
for (const e of d4) // LI dos
  console.log(`${e.nodeName} ${e.firstChild.nodeValue}`); // LI tres
```

Propiedades de los nodos

- Una vez localizado un nodo (aislado o dentro de una colección) es posible:
 - Acceder al nombre del nodo: **n.nodeName** o **n.tagName**;
 - Acceder a su nodo hijo de tipo Text (si existe) con: **n.firstChild.nodeValue**;
 - Acceder a sus colecciones **childNodes** y **attributes**.
 - Leer atributos con `getAttribute()`: **n.getAttribute("id");**
 - Establecer atributos con `setAttribute()`: **n.setAttribute("class","a");**
 - Usar algunos atributos como si fueran propiedades. Ejemplo:
 - **n.href=<http://google.com>;**
 - **n.style.background="blue";**
 - **n.style.height="100px";**
 - En nodos de tipo element (o etiquetas) se puede acceder o modificar su contenido mediante **innerHTML**:
 - **console.log(n.innerHTML);**
 - **n.innerHTML="<h1>Lorem Ipsum</h1>";**
 - En nodos de tipo text se puede acceder o modificar su contenido mediante las propiedades `nodeValue` o `data` (son equivalentes): **n.firstChild.data="Prueba";**

Ejercicio 1 (ej1.htm)

- Partiendo de una página web con el siguiente contenido:

```
<div id="id0">Esto es un contenedor  
  <a href="http://uno.com">uno</a>  
  <div id="id1">Esto es un bloque  
    <a href="http://dos.com">dos</a>  
    <input name="prueba1" />  
    <input name="prueba2" />  
  </div>  
</div>
```

- Añadir un script al final de la página que permita:
 - Visualizar en consola el id y el texto de todas las etiquetas DIV del documento.
 - Visualizar en consola las URL contenidas en las etiquetas A del documento.
 - Crear una función denominada “verNodos” que recibe como parámetro un nodo y visualiza en consola todos los subnodos que hay debajo, diferenciado aquellos que son de tipo text y mostrando su texto.