

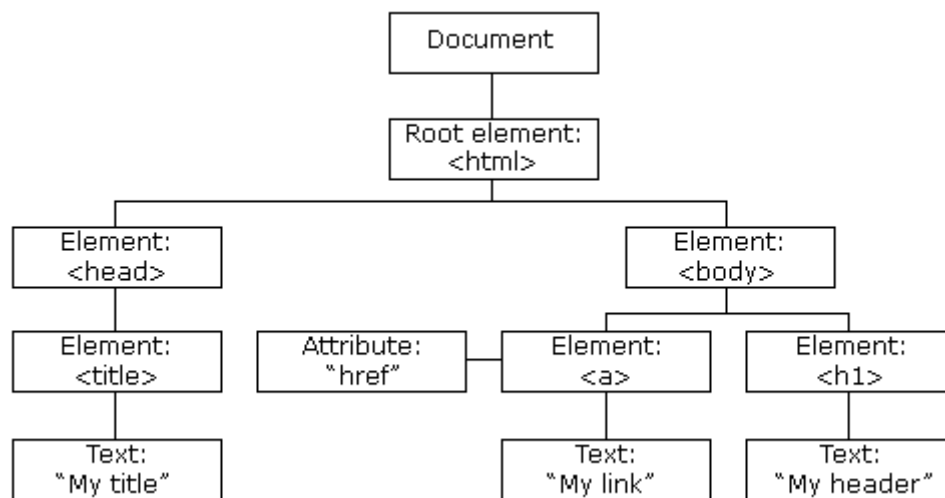
## HTML DOM

---

DOM (o Document Object Model) defineix l'estructura lògica dels documents HTML i XML, i la forma per accedir a l'estructura, contingut i estils per tal de manipular-los.

### Organització jeràrquica del DOM

El DOM està format per cada element del document, cadascun d'aquests elements del DOM s'anomena *node*.



El node superior és el node arrel i els elements `<head>` i `<body>` són els seus fills. Entre ells són germans (siblings), ja que estan al mateix nivell. Els atributs HTML com «`href`» formen part de l'estructura però no de la relació pare/fill.

### Accés al DOM

Podem accedir al DOM utilitzant JavaScript. Podem obtenir el contingut que necessitem a partir de les propietats de l'objecte *document*.

- a) **document.documentElement** : Permet accedir al contingut de l'etiqueta `<html>`.
- b) **document.head** : Permet accedir al contingut de l'etiqueta `<head>`.
- c) **document.body** : Permet accedir al contingut de l'etiqueta `<body>`.
- d) **document.forms** : Obtens una llista amb tots els elements `<form>` del document.

...

També es pot accedir al DOM utilitzant les següents funcions segons criteri de cerca:

- a) **document.getElementById()** : Segons id de l'element.
- b) **document.getElementsByClassName()** : Segons class de l'element.

Si el selector permet obtenir més d'un resultat es retorna una col·lecció HTML, un array que es pot iterar de la següent manera:

```
let main = document.getElementById("main");
let divs = main.getElementsByClassName("container");

for(let divIt of divs) {
  console.log(divIt.id);
}
```

A més fixa't que es pot utilitzar també sobre qualsevol element HTML, no només del document sencer.

- c) **document.getElementsByName()** : Segons name de l'element.
- d) **document.getElementsByTagName()** : Segons etiqueta de l'element.
- e) **document.querySelectorAll()** : Segons el selector CSS.

HTML

```
<ul>
  <li>Bread</li>
  <li>Coffee</li>
  <li>Apple</li>
</ul>
```

JavaScript

```
let meal = document.body.querySelectorAll("ul li:first-child");
```

En aquest exemple obtenim el primer element de la llista.

## Manipulació del DOM

Utilitzant JavaScript també podem manipular el DOM.

### Canvi CSS a través de les propietats

Podem canviar dinàmicament l'estil d'un element utilitzant l'atribut *style*:

```
var elem = document.getElementById("intro");  
elem.style.fontWeight = "bold";
```

Podem canviar l'estil d'un element assignant-li una classe:

```
elem.className = "highlight";  
elem.className += " biggest colored";
```

També podem usar la propietat *classList*.

Afegir classes a un element:

```
elem.classList.add("highlight");
```

Eliminar classes a un element:

```
elem.classList.remove("hide");
```

Si l'element inclou la classe l'elimina, sinó l'afegeix:

```
elem.classList.toggle("visible");
```

Indica si l'element disposa de la classe indicada:

```
elem.classList.contains("tiny");
```

### Canvi d'atributs

Podem obtenir el valor dels atributs de les etiquetes HTML usant el mètode *getAttribute()*:

```
var link = document.getElementById("myLink");  
var href = link.getAttribute("href");
```

Podem modificar-los usant el mètode *setAttribute()*:

```
link.setAttribute("href", "https://www.google.com/");
```

O podem esborrar-los usant el mètode *removeAttribute()*:

```
link.removeAttribute("href");
```

### Canvi del contingut HTML

Podem obtenir o modificar el contingut dels elements usant **innerHTML**:

```
var div1 = document.getElementById("main").innerHTML;  
var div2 = document.getElementById("second");  
div2.innerHTML = div1;
```

En aquest exemple, obtenim el contingut del div «main» per sobreescriure el contingut del div «second».

També podem afegir-hi etiquetes:

```
div2.innerHTML += "<li><a href='#>Item</a></li>";
```

Si l'string inclou cometes és recomanable usar `` en lloc de "" per delimitar-lo.

## Afegir elements

Podem afegir elements HTML usant el mètode `document.createElement()`:

```
var currentDiv = document.getElementById("main");  
var newButton = document.createElement("button");
```

Un cop creat l'element hem de fer que pengi d'algun element usant `appendChild()`:

```
currentDiv.appendChild(newButton);
```

## Eliminar elements

Localitzant prèviament el node pare, podem eliminar elements HTML usant el mètode `removeChild()`:

```
var parentElem = document.getElementById("main");  
var childElem = document.getElementById("hint");  
parentElem.removeChild(childElem);
```

Eliminem l'element amb id «hint» que penja de «main».

## ACTIVITATS

1. Utilitza el següent formulari HTML: <https://github.com/xbaubes/DAM/tree/main/MP4-Llenguatges-marques-SGI/HTML/Basic/exampleCode/form>

Modifica'l de manera que la pàgina de recepció mostri les dades enviades amb el nom del camp que li correspongui.

Les dades s'envien amb el mètode GET, de manera que les pots obtenir de la URL, les dades s'envien en el següent format:

**nomDelCamp=valorDelCamp&nomDelCamp=valorDelCamp**

Pots obtenir aquesta informació usant *location.search* i extreure'n camps concrets usant *URLSearchParams*: <https://developer.mozilla.org/en-US/docs/Web/API/Location/search>

2. Utilitza el mateix formulari HTML de l'exercici anterior.

No és necessari modificar-lo perquè no s'envii a través dels mecanismes HTML. Pots utilitzar JavaScript per evitar l'enviament usant el mètode *preventDefault()* de l'event *submit*. Amb JavaScript també pots obtenir el valor de cada camp i del label respectiu i que al prémer el botó d'enviar els mostri a la mateixa pàgina. Pots modificar l'HTML per habilitar una zona per mostrar el contingut.

Però abans hauràs de validar que tots els camps estan plens, si no és així, no es mostraran els valors i el border de l'element buit es mostrarà en vermell.

Utilitza una funció amb la següent capçalera per realitzar els canvis de color: *changeBorder(elementHTML,color)*

3. Amb JavaScript hauràs de modificar el DOM d'aquest document HTML: <https://github.com/xbaubes/DAM/blob/main/MP4-Llenguatges-marques-SGI/JavaScript/exercici-dom-2/index.html>

Al prémer el botó «CREATE» s'afegeix una fila a la llista que hauràs creat al div «#listElems»; aquesta llista ha d'incloure el text de l'input i un botó que ha de permetre esborrar la fila. Incorpora un identificador als elements que vas creant.

The diagram illustrates the state of a web form before and after clicking the 'CREATE' button. On the left, the input field contains the value '5' and the list below it contains 4 items, each with a 'REMOVE' button. On the right, the input field is empty and the list contains 5 items, each with a 'REMOVE' button. This demonstrates that clicking 'CREATE' adds a new item to the list.

Per aquesta pràctica és possible que necessitis navegar pels nodes del DOM, prova *parentNode* si necessites accedir al pare d'un element seleccionat.

## BIBLIOGRAFIA I WEBGRAFIA

«MDN Web Docs». [https://developer.mozilla.org/es/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model/Introduction). [https://developer.mozilla.org/en-US/docs/Learn/Forms/Form\\_validation](https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation)

«Tutorial Republic». <https://www.tutorialrepublic.com/javascript-tutorial/>

«W3 Schools». [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp)



Autor: Xavier Baubés Parramon

Aquest document es llicència sota Creative Commons versió 4.0.  
Es permet compartir i adaptar el material però reconeixent-ne l'autor original.