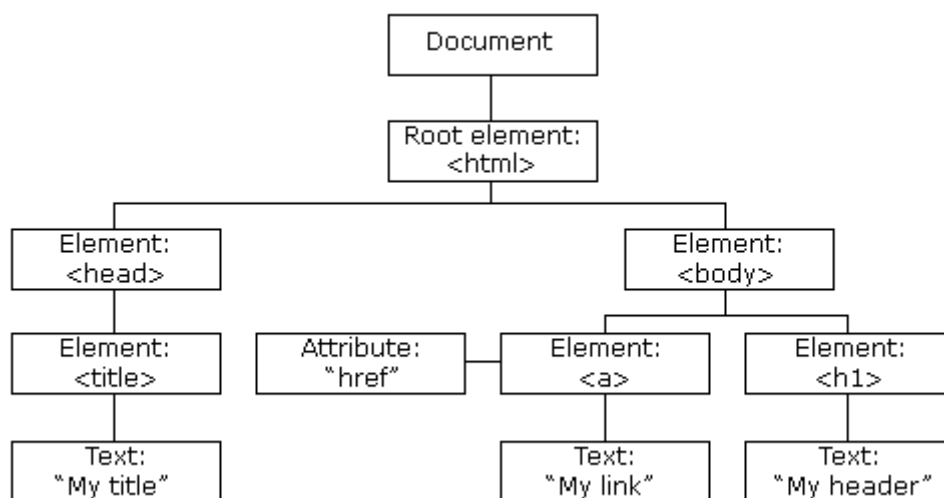


HTML DOM

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, una llista 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 d'estils a través de les propietats

Podem canviar 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 highlight";
```

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 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».

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 aquest formulari HTML:

<https://github.com/xbaubes/DAM/blob/main/MP4-Llenguatges-marques-SGI/HTML/Basic/exampleCode/form/send-form.html>

Modifica'l de manera que no s'envii a través dels mecanismes HTML. Utilitza JavaScript per obtenir el valor de cada camp i del label respectiu i al prémer el botó d'enviar els mostri a la mateixa pàgina.

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)`

2. 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. Hauràs d'incorporar un identificador als elements que vas creant per poder-los esborrar.

→

- 1
- 2
- 3
- 4

→

- 1
- 2
- 3
- 4
- 5

Per aquesta pràctica és possible que necessitis navegar pels nodes del DOM, prova `parentNode`.

BIBLIOGRAFIA I WEBGRAFIA

«MDN Web Docs». https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model/Introduction. 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



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.