

04

Desenvolupament Web en entorn Client

Cicle: Desenvolupament d'aplicacions web

Curs 2024-2025

©José Masip Alonso

NOTA: ALS EXERCICIS DE DOM TENIM QUE UTILITZAR LA METODOLOGIA APRESA A AQUEST DOCUMENT.

1. Introducció

En aquest document aprendrem a manipular el DOM.

2. Contingut

2.1. DOM.

Quan es carrega una pàgina web, el navegador crea un «**DOCUMENT Object Model**» de la pàgina.

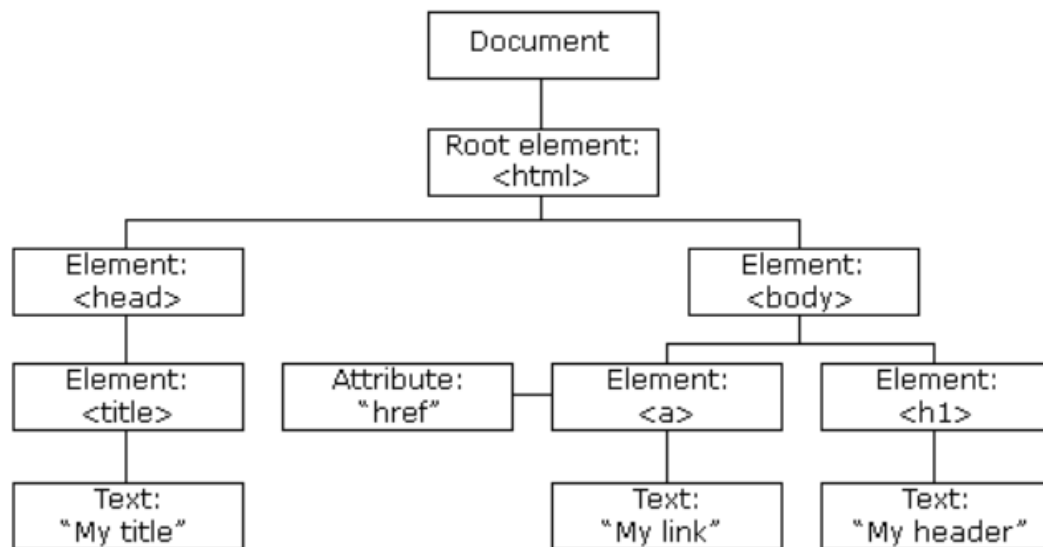
Defineix un estàndard per a accedir i manipular documents HTML.

Presenta un document html com una estructura d'arbre.

"El documento de W3C Object Model (DOM) es una interfaz de plataforma y lenguaje neutro que permite que los programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo de un documento".

Els nodes d'un arbre tenen una relació jeràrquica entre sí.

- En un arbre de nodes, el node superior s'anomena arrel.
- Cadascun dels nodes té exactament un pare, excepte l'arrel (que no té pare)
- Un node pot tindre qualsevol número de fills
- Els germans son els nodes amb el mateix pare.



2.2. Número de fills

Amb el vector `childNodes` podem saber el número de fills i accedir a ells.

```

<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Plantilla</title>
  <meta name="description" content="descripcio">
  <meta name="author" content="elteunom">
  <link rel="stylesheet" href="css/estilos.css">
</head>

<body>
  <h1>DOM</h1>
  <div id="capa">
    <p style="fontSize:10px">p1</p>
    <p style="fontSize:10px">p2</p>
  </div>
  <script language="JavaScript">
    alert("numero fills"+document.getElementById("capa").childNodes.length);
  </script>

  <p>tots navegadors generen mateix numero de fills?</p>
</body>
</html>

```

Si modifiquem la capa de la manera mostrada a continuació i mostrem el número de fills. **És el mateix?**

```
<div id="capa"><p>p1</p><p>p2</p></div>
```

Al següent exemple d'utilització del vector de fills augmentem la mida del text dels paràgrafs:

```
<script language="JavaScript">
function tamanytext(){
  var obj=document.getElementById('capa');
  text=14;
  for(f=0;f<obj.childNodes.length;f++)
  {
    obj.childNodes[f].style.fontSize=text+'px';
    text=text+4;
  }
}
</script>
```

Si deixem l'opció del div en vots de línia per a cadascun dels paràgrafs, aquesta solució genera un error. Si utilitzem sense espais ni vots de línia funciona correctament.

2.3. parentNode

Permet accedir al pare d'un node.

Sobre l'exemple anterior:

```
var paragrafs=document.getElementsByTagName('p');
var pare=paragrafs[0].parentNode; //estariem en el div
```

2.4. nodeType.

Aquesta propietat mostra el tipus d'un node.

| Node type | Description | Children |
|-------------------------|---|--|
| 1 Element | Represents an element | Element, Text, Comment, ProcessingInstruction, CDATASection, EntityReference |
| 2 Attr | Represents an attribute | Text, EntityReference |
| 3 Text | Represents textual content in an element or attribute | None |
| 4 CDATASection | Represents a CDATA section in a document (text that will NOT be parsed by a parser) | None |
| 5 EntityReference | Represents an entity reference | Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference |
| 6 Entity | Represents an entity | Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference |
| 7 ProcessingInstruction | Represents a processing instruction | None |
| 8 Comment | Represents a comment | None |
| 9 Document | Represents the entire document (the root-node of the DOM tree) | Element, ProcessingInstruction, Comment, DocumentType |
| 10 DocumentType | Provides an interface to the entities defined for the document | None |
| 11 DocumentFragment | Represents a "lightweight" Document object, which can hold a portion of a document | Element, ProcessingInstruction, Comment, Text, CDATASection, EntityReference |
| 12 Notation | Represents a notation declared in the DTD | None |

Utilitzant aquesta propietat podem solucionar el problema del vots de línia anterior.

```
<script language="JavaScript">
function fexecutar(){
  var obj=document.getElementById('capa');
  text=14;
  for(f=0;f<obj.childNodes.length;f++)
  {
    //alert(obj.childNodes[f].nodeType);
    if(obj.childNodes[f].nodeType==1){
      obj.childNodes[f].style.fontSize=text+'px';
      text=text+4;
    }
  }
}
</script>
```

2.5. Crear Nodes.

Depenent del tipus de node que volem crear tenim que utilitzar:

- `appendChild()`. Annexa el node passat com a paràmetre a l'objecte sobre el que executem.
- `createElement()`. Crea un node que representa una etiqueta.
- `createTextNode()`.
- `cloneNode()`. Clona un node

Exemple on creem un paràgraf:

```
var para=document.createElement("p"); // crear etiqueta
var node=document.createTextNode("NOU PARÀGRAF."); //crear text
para.appendChild(node); // afegir el text al paràgraf

document.body.appendChild(para); // afegir el paràgraf al body
```

Altra funció que ens podria servir es `insertBefore()`.

2.6. Eliminar Nodes.

Per a eliminar un node cal utilitzar:

- `removeChild()`. elimina el node passat com a paràmetre.

Exemple on eliminem un paràgraf:

```
document.body.removeChild(document.body.lastChild);

document.body.removeChild(document.body.childNodes[document.body.childNodes.length-1])
```

2.7. Atributs.

Per a crear o eliminar un atribut cal utilitzar:

- `setAttribute()`. Crea el atribut passat com a primer paràmetre amb el valor passat com a segon.
- `removeAttribute()`. Elimina l'atribut passat com a paràmetre.
- `getAttribute()`. Obte el valor de l'atribut.

Exemple :

```
Node.setAttribute('id','hiper');  
node.getAttribute('id');  
node.removeAttribute('id')
```

DOM

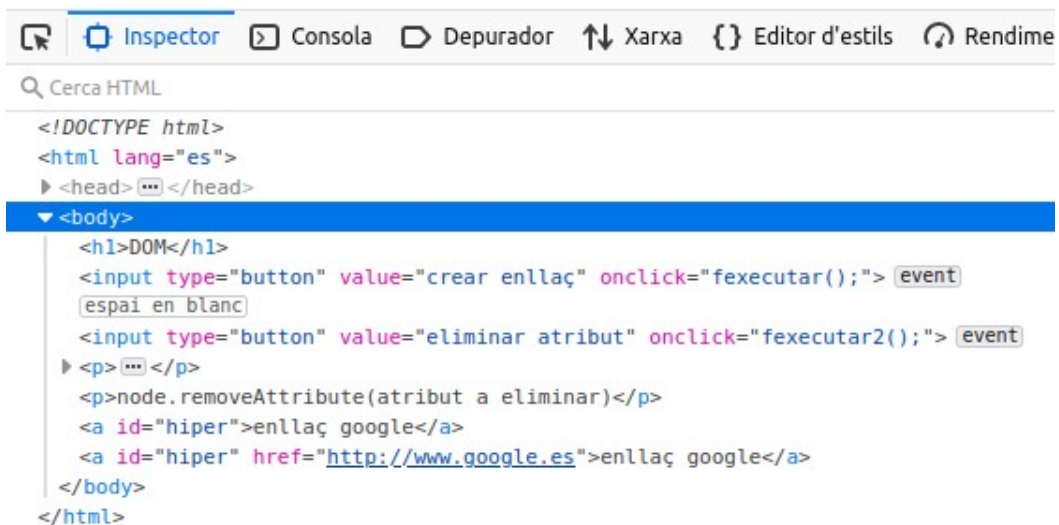
crear enllaç

eliminar atribut

node.setAttribute(atribut a definir, valor assignem)

node.removeAttribute(atribut a eliminar)

enllaç google [enllaç google](#)



2.8. Events.

Per a crear o eliminar un event cal utilitzar:

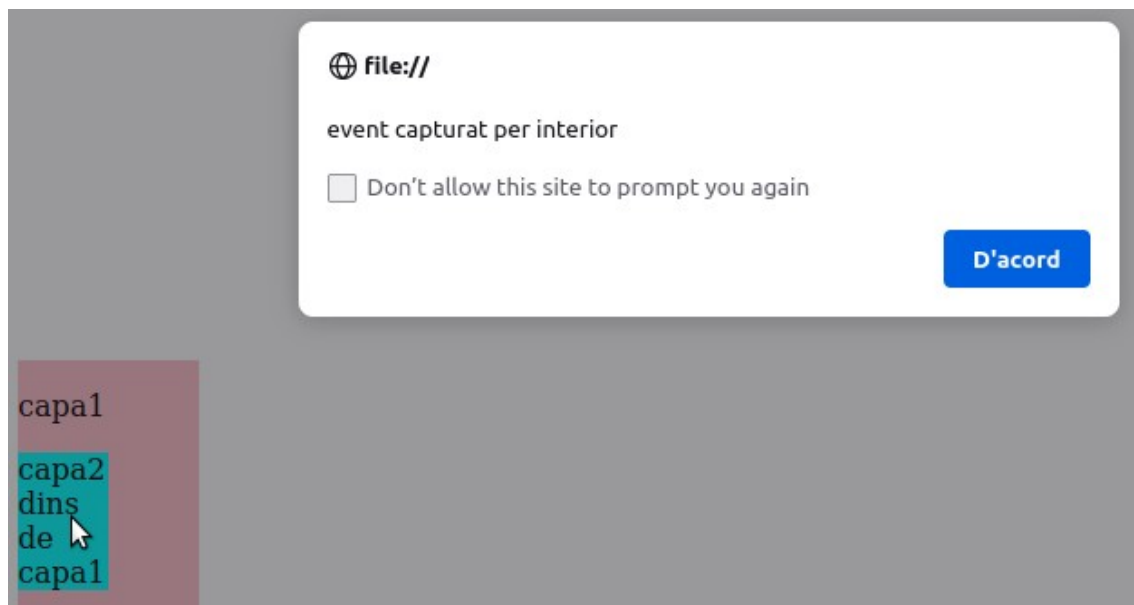
- `addEventListener()`. Crea l'event passat com a primer paràmetre que executa la funció passada com a segon paràmetre. Amb el tercer paràmetre indiquem l'ordre en que es captura l'event.
- `event.target`. Ens indica l'objecte genera event

- event.type. Ens indica el nom de l'event
- event.button. Ens indica quin boto del ratolí hem pulsat (0 esquerra, 1 mig, 2 dreta)
- event.stopPropagation(). no es propagaria l'event als demes objectes.

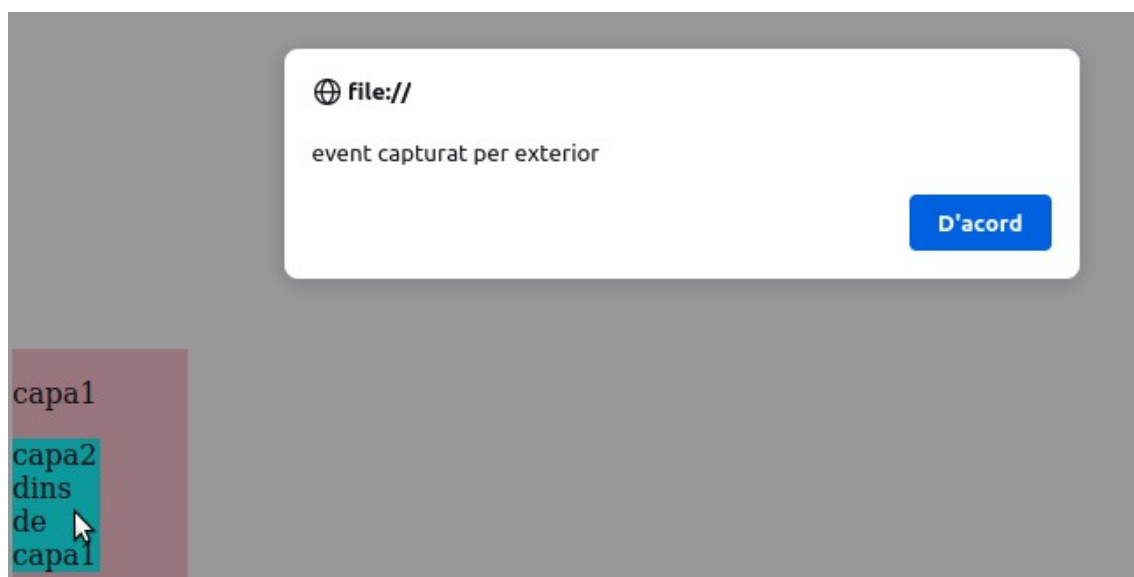
Exemple:

```
<!doctype html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Plantilla</title>
  <meta name="description" content="descripcio">
  <meta name="author" content="Jose Masip">
  <link rel="stylesheet" href="css/estilos.css">
  <script language="JavaScript">
    window.addEventListener('load',ini,false);
    function ini(){
//interior i despres es propaga a la exterior
document.getElementById('capa1').addEventListener('click',fexecutar,false);
document.getElementById('capa2').addEventListener('click',fexecutar2,false);
    }

    function fexecutar(){
      alert("event capturat per exterior")
    }
    function fexecutar2(e){
      alert("event capturat per interior");
    }
  </script>
</head>
<body>
  <h1>DOM</h1>
  qui captura l'event?
  <div id="capa1" style="background-color:pink;position:absolute;width:100px;top:400px;">
    <p>capa1</p>
    <div id="capa2" style="background-color:cyan;width:50px;">
      <p>capa2 dins de capa1</p>
    </div>
  </div>
</body>
</html>
```

Si modifiquem el false per un true, aconseguirem que primer capture l'event la capa exterior i després la interior.



Si a la funció que s'executa primer afegim el `e.stopPropagation()` provocarem que l'event no es propague i la segon funció no s'execute.

3. Activitats.

- E01. Crea una pàgina amb DOM amb la següent funcionalitat:

DOM

Crea 3 paragrafs. afegeix el paragraf al principi

NOU PARAGRAF.

p1

p2

p3

executar

- Afegir un paràgraf al principi de la capa.
- E02. Crea una pàgina amb DOM amb la següent funcionalitat:

DOM

Crea 3 paragrafs. eliminar el primer paragraf. Crear al final

p2

p3

NOU PARAGRAF 0.

NOU PARAGRAF 1.

crear

eliminar

- Eliminar el primer paràgraf.
 - Afegir al final numerant el número de paràgraf creat.

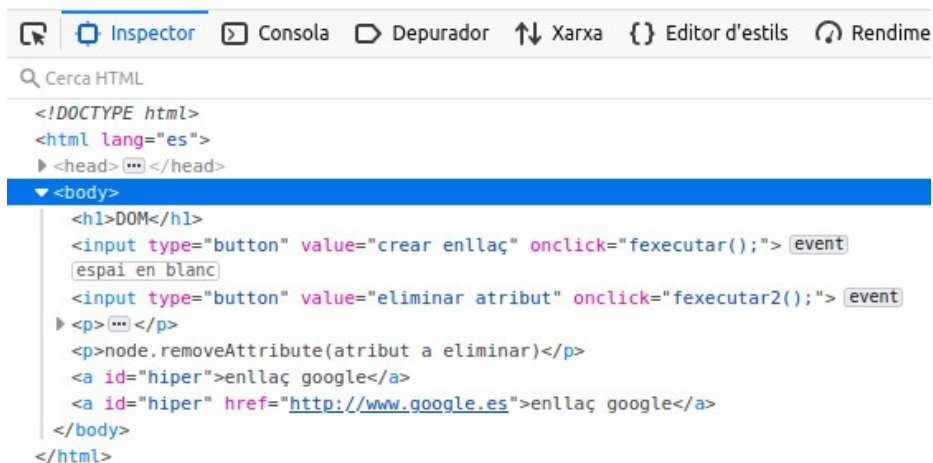
- E03. Crea una pàgina amb DOM amb la següent funcionalitat:

DOM

node.setAttribute(atribut a definir, valor assignem)

node.removeAttribute(atribut a eliminar)

enllaç google [enllaç google](#)



- Crea un hipervincle amb un identificador al polsar el boto.
- Elimina l'atribut href al polsar el boto corresponent.

- E04. Crea una pàgina amb DOM amb la següent funcionalitat:

| | |
|--|---|
| <div>3 ▾</div> <div>crear formulari</div> <div>camp 1 <input type="text"/></div> <div>camp 2 <input type="text"/></div> <div>camp 3 <input type="text"/></div> | <div>6 ▾</div> <div>crear formulari</div> <div>camp 1 <input type="text"/></div> <div>camp 2 <input type="text"/></div> <div>camp 3 <input type="text"/></div> <div>camp 4 <input type="text"/></div> <div>camp 5 <input type="text"/></div> <div>camp 6 <input type="text"/></div> |
|--|---|

- Crea un select amb X valors (variable al principi del document).
 - Crea tants camps com el número seleccionat al camp anterior.
 - Aquest camps cal que siguin funcionals per a rebre's quan enviarem el formulari.
- E05. Crea una pàgina amb DOM amb la següent funcionalitat:

☐ obrir en un altra pagina

[google](#)

[yahoo](#)

[terra](#)

- Si marquem la casella de verificació obrim l'enllaç a una altra pestanya.
 - Si no està marcada obrim a la finestra actual.
 - El número d'hipervincles pot variar i funcionarà exactament igual el nostre codi.

- E06. Crea una pàgina amb DOM amb la següent funcionalitat:

[google](#)

[yahoo](#)

[terra](#)

[google](#)

[yahoo](#)

[terra](#)

[google](#)

[yahoo](#)

[terra](#)

[obrir tots](#)



- Si fem click sobre un enllaç, obrim aquest.
- Si fem click sobre «obrir tots» obrim tots els enllaços a distintes pestanyes (cal evitar que aquest hipervincle ens redirigisca a altra pàgina).
- El número d'hipervincles pot variar i funcionarà exactament igual el nostre codi.

- E07. Modifica el «ahorcado» fet anteriorment per a:
 - Els camps de les lletres els creem amb DOM. Sols crearem els camps necessaris per a la paraula de la partida. Es a dir:
 - si la paraula es «cua» crearem 3 camps.
 - Si la paraula es «porta» crearem 5 camps.

4. Activitats entregables.

Entrega les següents activitats:

- E04
- E06
- E07

5. Bibliografia.

- https://www.w3schools.com/js/js_htmlDOM.asp
- <https://dom.spec.whatwg.org/>
- <https://www.w3schools.com/jsref/default.asp>

6. Aclariments

No intentes copiar exercicis d'altres companys o internet. ni tans sols pegar una miradeta. Si que pots consultar la sintaxis d'una funció, etc. Però no et serveix per a res intentar buscar la solució.

És el major error que pot realitzar qui comença a programar. Sols et deixarà una falsa sensació d'aprenentatge i després no sabràs solucionar els problemes per tu mateix.

Cal intentar que el resultat que genere el vostre codi siga el mes paregut possible a l'exercici (si hi ha imatge d'exemple).

Com els Navegadors no son compatibles entre sí al 100% jo corregire els examens amb FIREFOX. Per tant, es recomanable utilitzar aquest per als nostres exercicis.

Crea uns arxius per a cada exercici o apartat. Així un error a un no afectarà a la resta.

Per a entregar els exercicis crear una carpeta amb el vostre nom i comprimiu aquesta (el nom de l'arxiu comprimit serà elteunom.zip o elteunom.tar.gz). Si entregueu arxius solts o arxius genèrics sense identificar el vostre nom puc confondre'm i no puntuar-vos algun exercici al no saber de qui és.

Cal entregar tots els arxius necessaris per a l'exercici (fotos, llibreries, etc)