

DOM

Mòdul professional 6: desenvolupament web en entorn client



JavaScript

DOM

Mòdul professional 6: desenvolupament web en entorn client

UF1: Sintaxi del llenguatge.
Objectes predefinits del llenguatge.

UF2: Estructures definides pel programador.
Objectes.

UF3: Esdeveniments.
Manejament de formularis.
Model d'objectes del document.

UF4: Comunicació asíncrona client-servidor.

DOM

En aquesta tercera part de la unitat formativa 3 tractarem el model d'objecte del document.

UF3: Esdeveniments.

Manejament de formularis.

Model d'objectes del document

DOM

Dividirem el contingut dels formularis en els següents apartats:

- **Introducció.**
- **Arbre DOM.**
- **Tipus de nodes.**
- **Manipulació del DOM.**

DOM



JavaScript

Introducció

- El **DOM** (*Document Object Model*, Model d'objectes de document) és un model computacional a través del qual els programes i els scripts poden accedir i modificar dinàmicament el contingut, estructura i estil dels documents HTML i XML.
- El seu objectiu és oferir un model orientat a objectes per al tractament i manipulació en temps real (o de forma dinàmica) de documents web.
- Podem realitzar les següents operacions:
 - Canviar i llegir el contingut i la estructura.
 - Canviar i llegir estils CSS.
 - Gestionar events.

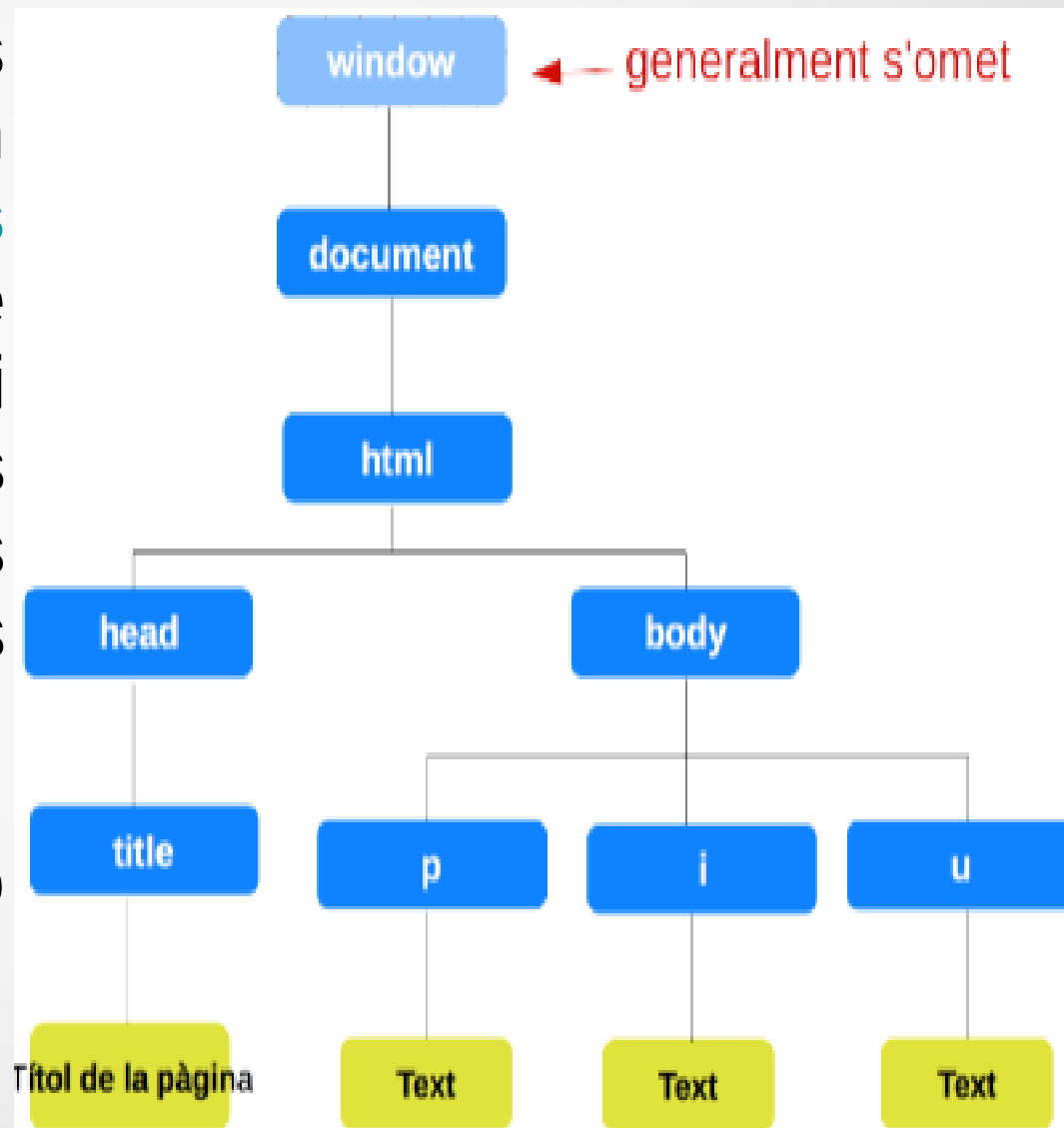
DOM



JavaScript

Arbre DOM

- DOM transforma tots els documents HTML en un **conjunt d'elements** anomenats **nodes**, que estan **interconnectats** i que **representen** els **continguts** de les pàgines web i de les **relacions** entre ells.
- Pel seu aspecte, la unió de tots els nodes es diu "**arbre de nodes**".



- L'arrel de l'arbre de nodes de qualsevol pàgina HTML sempre és la mateixa: un node de tipus especial denominat **Document**.
- A partir d'aquest node arrel, cada etiqueta HTML es transforma en un node de tipus **Element**.
- La conversió d'etiquetes en nodes es realitza de forma jeràrquica. D'aquesta forma, del node arrel solament poden derivar els **nodes HEAD i BODY**.

- A partir d'aquesta derivació inicial, cada etiqueta HTML es transforma en un node que deriva del node corresponent a la seva etiqueta pare.
- La transformació de les etiquetes HTML habituals genera dos nodes:
 - i. el primer és el node de tipus **Element** (corresponent a la pròpia etiqueta HTML) i
 - ii. el segon és un node de tipus **Text** que conté el text tancat per aquesta etiqueta HTML.

DOM. Arbre DOM

4/7

```
<!DOCTYPE html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Arbre DOM</title>
```

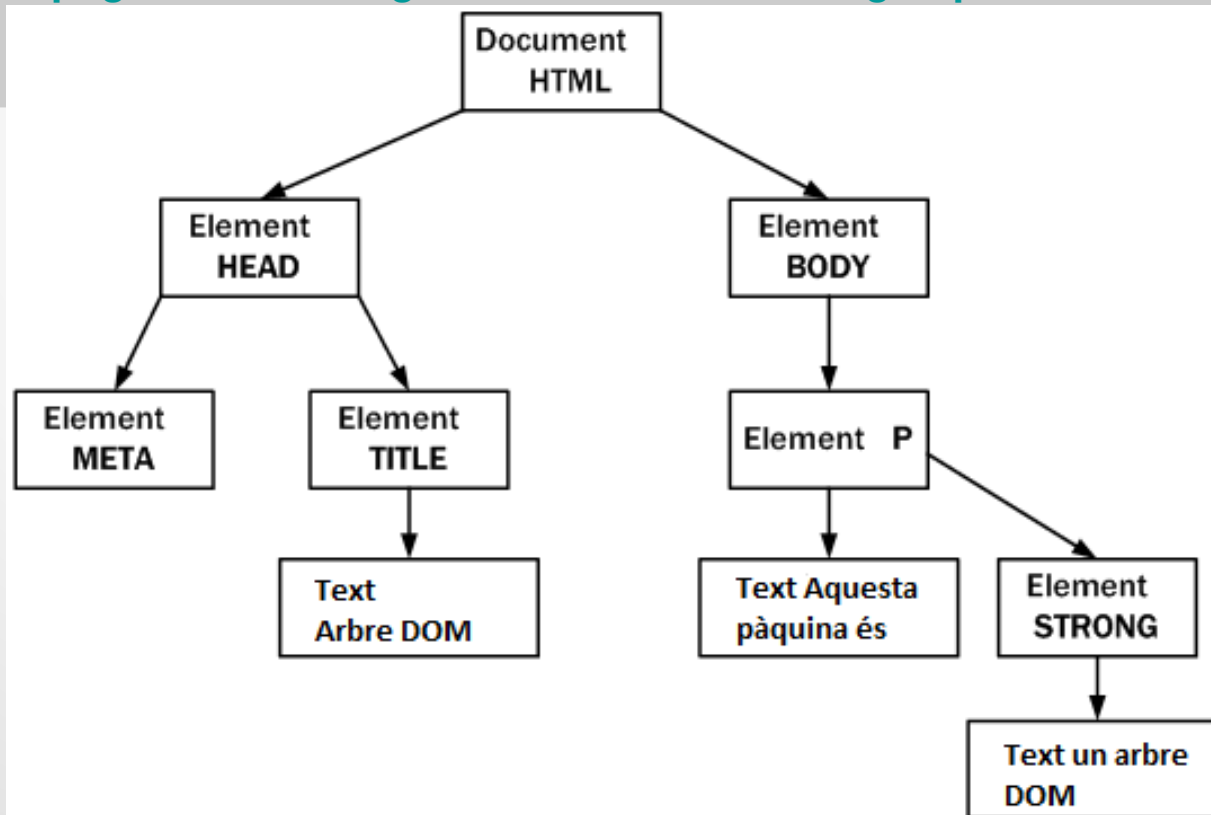
```
</head>
```

```
<body>
```

```
<p>Aquesta pàgina és <strong>un arbre DOM</strong></p>
```

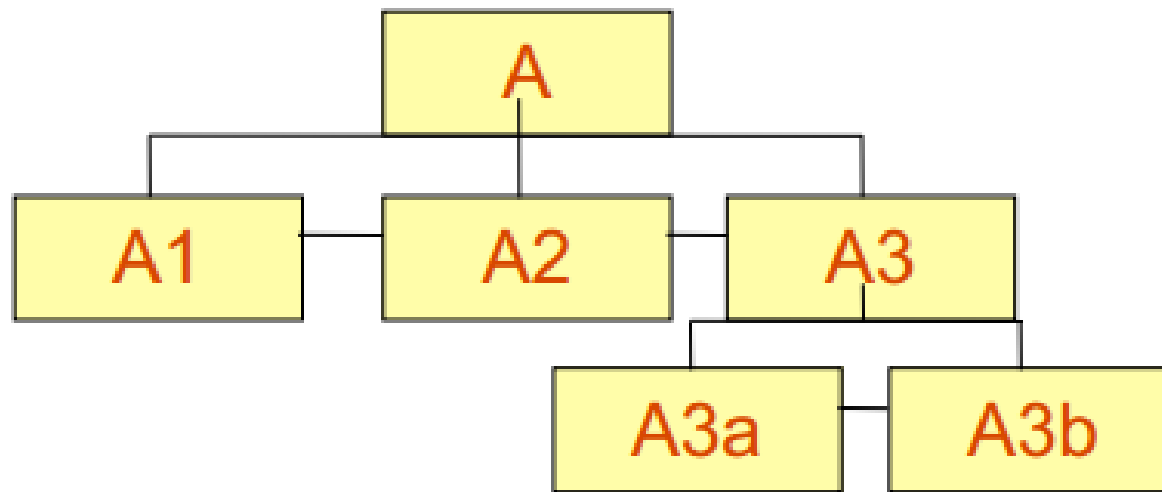
```
</body>
```

```
</html>
```



- L'especificació completa de DOM defineix 12 tipus de nodes (**Document**, DocumentFragment, DocumentType, ProcessingInstruction, EntityReference, **Element**, **Attr**, **Text**, CDATASection, **Comment**, Entity, Notation).
- Els més importants són:
 - **Document (nodeType 9)**, node arrel del que deriven tots els altres nodes de l'arbre.
 - **Element (nodeType 1)**, representa cadascuna de les etiquetes HTML. Es tracta de l'únic node que pot contenir atributs i l'únic del que poden derivar altres nodes.
 - **Attr (nodeType 2)**, es defineix un node d'aquest tipus per representar cadascun dels atributs de les etiquetes HTML, és a dir, un per cada parell atribut=valor.
 - **Text (nodeType 3)**, node que conté el text tancat per una etiqueta HTML.
 - **Comment (nodeType 8)**, representa els comentaris inclosos a la pàgina HTML.

- Cada node té diverses propietats que reflecteixen el grau parentiu amb els altres.
- Algunes d'aquestes **propietats** són les següents:
 - **childNodes/children**: array amb els nodes fills.
 - **firstChild/firstElementChild**: primer node fill.
 - **lastChild/lastElementChild**: últim node fill.
 - **parentNode**: node pare.
 - **nextSibling/nextElementSibling**: següent germà. (node al mateix nivell i amb el mateix pare).
 - **lastSibling/lastElementSibling**: germà anterior. (node al mateix nivell i amb el mateix pare).



`A.firstChild = A1`

`A.lastChild = A3`

`A.childNodes.length = 3`

`A.childNodes[0] = A1`

`A.childNodes[1] = A2`

`A.lastChild.firstChild = A3a`

`A3b.parentNode.parentNode = A`

`A1.nextSibling = A2`

`A3.prevSibling = A2`

`A3.nextSibling = null`

DOM



JavaScript

Manipulació del DOM

- És important recordar que l'accés als nodes, la seva modificació i la seva eliminació solament és possible quan l'arbre DOM ha estat construït completament, és a dir, després que la pàgina HTML es carregui per complet.

```
//Javascript
window.onload = function(){
    ...
}

//jQuery
$(document).ready(function(){
    ...
});
```


- A continuació es mostren les diferents formes per aconseguir un node de l'arbre de manera directa, sense haver de recórrer-lo:
 - **getElementsByTagName("nomEtiqueta")**: Obté tots els **elements** de la pàgina HTML que coincideixen amb el paràmetre que se li passa a la funció.
 - **getElementsByName("nomElement")**: Obté tots els **elements** l'atribut dels quals name sigui igual al paràmetre proporcionat.
 - **getElementById("idelement")**: Retorna l'**element** HTML l'atribut del qual coincideix amb el paràmetre indicat en la funció.
 - **getElementsByClassName("nomclass")**: Obté tots els **elements** en el document amb el nom de classe especificat, com un objecte nodelist. L'objecte nodelist representa una col·lecció de nodes. Els nodes poden ser accedits per nombres d'índex. L'índex comença en 0.

//Javascript

```
var Elements_tagName =  
    document.getElementsByTagName("p");  
var Elements_name =  
    document.getElementsByName("nameABuscar");  
var Element_id =  
    document.getElementById("idABuscar");  
var Elements_class =  
    document.getElementsByClassName("classABuscar");
```

//jQuery

```
var Elements_tagName = $("p");  
var Elements_name = $("input[name=nameABuscar]");  
var Element_id = $("#idABuscar");  
var Elements_class = $(".classABuscar");
```

- **Creació de nodes HTML simples:**

Crear i afegir a la pàgina un nou element HTML senzill consta de quatre passos diferents:

- 1) **Creació** d'un **node** de tipus **Element** que representi a l'element.
- 2) **Creació** d'un **node** de tipus **Text** que representi el contingut de l'element.
- 3) **Afegir** el **node Text** com a node **fill** del **node Element**.
- 4) **Afegir** el **node Element a la pàgina**, en forma de node fill del node corresponent al lloc en el qual es vol inserir l'element.

//Javascript

// Crear node de tipus Element

var **parrafo** = **document.createElement("p");**

// Crear node de tipus Text

var **contenido** = **document.createTextNode("Hola Mundo!");**

// Afegir el node Text com a fill del node Element

parrafo.appendChild(contenido);

// Afegir el node Element com a fill de la pagina

document.body.appendChild(parrafo);

//02

//jQuery

var **parrafo** = **document.createElement("p");** // Crear node de tipus Element

// Crear node de tipus Text

var **contenido** = **document.createTextNode("Hola Mon!");**

// Afegir el node Text com a fill del node Element

\$(parrafo).append(contenido);

\$('#body').append(contenido); // **\$(document.body).append(contenido);**

// Transforma les etiquetes i texts a nodes i els afegeix

\$('#body').append('<p>Hola Mundo!</p>');

//03

• Inserció i eliminació de nodes HTML en JS:

- Afegir un element al final dels elements fills.

```
document.getElementById('div').appendChild(nou);
```

- Afegir un nou element després de l'element seleccionat.

```
document.getElementById('div').insertBefore(nou,  
document.getElementById('p2'));
```

- Eliminar un fill seleccionat.

```
var cont = document.getElementById('div');  
cont.removeChild(cont.lastElementChild);  
nou.parentNode.removeChild(nou);
```

- Reemplaçar un node per un altre

```
cont.replaceChild(nou, vell);
```

- **Inserció i eliminació de nodes HTML en jQuery (1/2):**

- Afegeir un element al principi dels elements fills.

```
$("#div").prepend("<p>paràgraf afegit</p>");
```

- Afegeir un element al final dels elements fills.

```
$("#div").append("<p>paràgraf afegit</p>");
```

- Afegeix un nou element abans de l'element seleccionat.

```
$("#<p>nou element</p>").insertBefore("div");
```

- Afegeix un nou element després del seleccionat.

```
$("#<p>altre element</p>").insertAfter("div");
```

- **Inserció i eliminació de nodes HTML en jQuery (2/2):**

- Afegeir un element abans de l'element.

```
$("#div").after("<p>paràgraf afegit</p>");
```

- Afegeir un element després de l'element.

```
$("#div").before("<p>paràgraf afegit</p>");
```

- Elimina l'element seleccionat

```
$("#div").remove();
```

- Afegeix un nou element després del seleccionat.

```
$("#div").replaceWith("<div><p>nou node</p></div>");
```

- **Inserció i eliminació de nodes HTML en jQuery (2/2):**

- Afegeir un element abans de l'element.

```
$("#div").after("<p>paràgraf afegit</p>");
```

- Afegeir un element després de l'element.

```
$("#div").before("<p>paràgraf afegit</p>");
```

- Elimina l'element seleccionat

```
$("#div").remove();
```

- Afegeix un nou element després del seleccionat.

```
$("#div").replaceWith("<div><p>nou node</p></div>");
```


- **Selecció dels nodes pare i fills en JS:**

- Obtenir l'element pare d'un node.

```
document.getElementById('div').parentNode;
```

- Obtenir el primer node fill.

```
document.getElementById('div').firstChild;
```

- Obtenir el primer node element fill.

```
document.getElementById('div').firstElementChild;
```

- Obtenir l'últim node fill.

```
document.getElementById('div').lastChild;
```

- Obtenir l'últim node element fill.

```
document.getElementById('div').lastElementChild;
```

• Selecció dels nodes pare i fills en jQuery (1/2):

- Obtenir l'element pare d'un node.

```
$('div').parent();
```

- Obtenir tots els elements pare d'un node.

```
$('div').parents().css({"color": "red"});
```

- Obtenir tots els elements pare d'un node d'un tipus.

```
$('li').parents('ul').css({"color": "blue"});
```

- Obtenir els elements pare d'un node fins un en concret.

```
$('li').parentsUntil('div').css({"color": "blue"});
```

• Selecció dels nodes pare i fills en jQuery(2/2):

- Obtenir el primer node li que és primer fill del seu pare.

```
$('ul li:first-child').css("color", "red");
```

- Obtenir l'últim node li que és primer fill del seu pare.

```
$('ul li:last-child').css("color", "blue");
```

- Obtenir tots els nodes element fills.

```
$('ul').children().css("color", "green");
```

- Obtenir tots els nodes element fills d'un tipus concret.

```
$('ul').children('li').css("color", "green");
```

- Obtenir tots els descendents d'un tipus concret.

```
$('div').find('li').css("color", "red"); //find('*') tots
```

- **Selecció de nodes adjacents en JS:**

- Obtenir el node adjacent posterior d'un node.

```
document.getElementById('ul').nextSibling;
```

- Obtenir el node element adjacent posterior d'un node.

```
document.getElementById('ul').nextElementSibling;
```

- Obtenir el node adjacent anterior d'un node.

```
document.getElementById('ul').previousSibling;
```

- Obtenir el node element adjacent anterior d'un node.

```
document.getElementById('ul').previousElementSibling;
```

- **Selecció de nodes adjacents en jQuery(1/2):**

- Obtenir els nodes adjacents d'un node.

```
$('#li2').siblings();
```

- Obtenir el node adjacent posterior d'un node.

```
$('#li2').next();
```

- Obtenir els nodes adjacents posteriors d'un node.

```
$('#li2').nextAll();
```

- Obtenir els nodes adjacents posteriors d'un node fins a.

```
$('#li2').nextUntil('#li4');
```

- **Selecció de nodes adjacents en jQuery(2/2):**

- Obtenir el node adjacent anterior d'un node.

```
$('#li4').prev();
```

- Obtenir els nodes adjacents anteriors d'un node.

```
$('#li4').prevAll();
```

- Obtenir els nodes adjacents anteriors d'un node fins a.

```
$('#li4').prevUntil('#li2');
```

- **Selecció de nodes atributs en JS:**

- Obtenir els nodes atribut d'un node.

```
document.getElementById('ul').attributes[0].nodeName;
```

- Obtenir el valor d'un atribut d'un node.

```
document.getElementById('ul').getAttribute('id');
```

- Actualitza el valor d'un atribut d'un node.

```
document.getElementById('ul').setAttribute('id','jo');
```

- Elimina un atribut d'un node.

```
document.getElementById('ul').removeAttribute('id');
```

- **Selecció de nodes atributs en jQuery:**

- Obtenir el valor d'un atribut d'un node.

```
$('#li2').attr('name');
```

- Actualitza el valor d'un atribut d'un node.

```
$('#li2').attr('name', '#li2');
```

- Actualitza el valor de diversos atributs d'un node.

```
$("img").attr({"width":500, "height":500});
```

- Elimina el valor d'un atribut d'un node.

```
$('#li2').removeAttr('name');
```


- **Selecció directa de nodes en jQuery:**

- Obtenir el primer node element.

```
$('li').first().css("color", "red");
```

- Obtenir l'últim node element.

```
$('li').last().css("color", "blue");
```

- Obtenir el node element d'una posició concreta.

```
$('li').eq(3).css("color", "red"); // pimer = 0
```

- Obtenir tots els nodes element amb una condició.

```
$('ul').filter(".classe");
```

- Obtenir els nodes que no compleixen una condició.

```
$('ul').not(".classe");
```