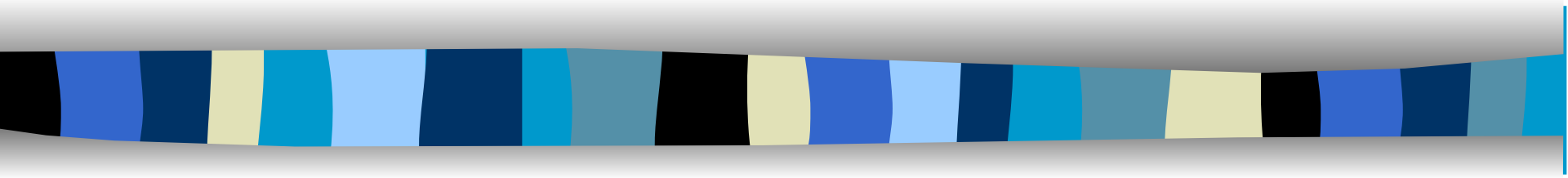
UD06 Model d'objectes de document DOM



JavaScript

Introducció.

- El Model d'objectes de document (DOM) oferix als programadors un accés sense precedents a HTML i els permet manipular i veure HTML com un document XML navegant pels nodes existents que formen la pàgina, podent manipular els seus atributs i inclús crear nous elements

- Més informació :
  - https://es.wikipedia.org/wiki/Document_Object_Model
  - http://www.w3schools.com/js/js_htmldom.asp

- DOM és una representació del document en forma d'arbre. De manera que cada element estarà representat per un node

- El Model d'objectes de document (DOM) és un API d'estructura en arbre esta definit per a XML, independent del llenguatge.
- Ens centrarem a estudiar la implementació de Javascript

Jerarquia de nodes

- El DOM definix diversos tipus de nodes per a representar els distints aspectes del codi XML :
  - **document**: El node de nivell superior a què es connecten els altres nodes.
  - **element**: Representa els continguts d'una etiqueta d'obertura i una altra de tancament, com en <etiqueta></etiqueta>. Este tipus de node només pot incloure atributs i nodes secundaris.
  - **attr**: Representa un atribut de parell de nom i valor. No pot incloure nodes secundaris
  - **text**: Representa text sense processar en un document XML inclòs dins d'etiquetes d'obertura i de tancament. Este tipus de node no pot tindre nodes secundaris.

ACCEDIR A NODES DOM

- **document.getElementById(id);**
  - Torna un element amb el seu atribut id establit en un valor concret. En XHTML, l'atribut id és únic.

    var oDiv1 = document.getElementById("div1");

- **.getElementsByTagName(etiqueta),**
  .getElementsByClassName(text);
  - Torna un array d'elements l'etiqueta / el class dels quals correspon amb el valor establit.

    var imgs = oDiv1.getElementsByTagName("img");

- ***document.getElementsByName(name);***
  - *Torna un array d'elements l'atribut name dels quals correspon amb el valor establit.*

    var oReds = document.getElementsByName("redColor");

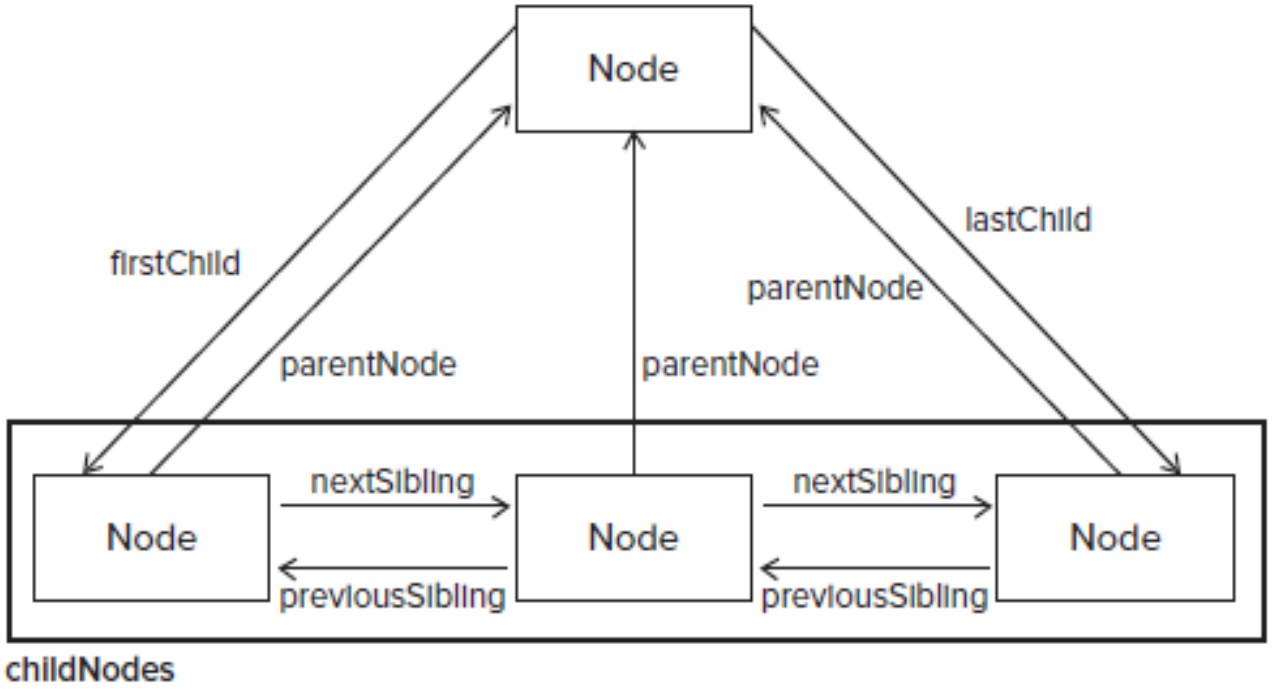
- ***document.querySelector("selector");***
  document.querySelectorAll("selector");
  - Torna el primer element o un array d'elements el selector CSS del qual correspon amb l'especificat

    var parrafos = document.getElementsByName(".redColor");

ACCEDIR A NODES DOM

- **document.head**
  - Torna l'element head
- **document.body**
  - Torna l'element body
- ***element.children***
  - *Torna un array amb els descendents fills. També existix* ***element.childNodes*** *però inclou nodes tipus text.*
- ***element.parentNode – element.parentElement***
  - Torna el node/element pare d'un element
- ***element.nextElementSibling - element.previousElementSibling***
  - *Torna el següent o l'anterior node del mateix nivell*
- ***element.firstChild - element.lastChild***
  element.firstElementChild – element.lastElementChild
  - *Torna el primer o l'últim node/element fill*

Jerarquia de nodes. Relacions entre nodes.



ACCEDIR A INFORMACIÓ DELS NODES

- **Node.nodeValue**
  - Torna el valor d'un node
- **Node.nodeName**
  - Torna el nom d'un node
- ***Node.nodeType***
  - *Torna el tipus d'un node.*
- ***Node.textContent***
  - *Posa i torna el contingut tipus text d'un node*
- ***Element.innerHTML***
  - *Posa i torna el contingut d'un element*
- ***Element.innerText***
  - *Posa I torna el contingut tipus text d'un element. Molt semblant a textContent. No està suportat per Firefox, en el seu lloc utilitzar innerHTML o textContent o nodeValue*

CREAR I MANIPULAR NODES

- **document.createElement(tag);**
  - Genera un element HTML. Encara no estarà en el DOM.
- **document.createTextNode(“text”);**
  - Genera un node de text que podem introduir dins d'un element. Equival a **element.innerText** = “text”
- **element.appendChild(childElement);**
  - Afig un nou element fill al final de l'element pare.
- **element.insertBefore(newChildElement, childElement);**
  - Inserix un nou element fill abans de l'element fill rebut com a segon paràmetre.
- **element.innerHTML=”...”;**
  - Torna o establix la sintaxi HTML descrivint els descendents de l'elemento.Al establir-se es reemplaça la sintaxi HTML de l'element per la nova.

createElement(), createTextNode() y appendChild().

```
<html>
  <head>
    <title>exemple de
    CreateElement()</title>
  </head>
  <body>
  </body>
</html>
```

A esta pàgina se li desitja afegir el codi següent:

*** Hola mundo ***

1.Generem l'element .

var oP = document.createElement("p");

2. Generem el node de text:

var oText = document.createTextNode("Hola mundo");

3. *Ara hem de connectar l'element* amb el seu element text.

oP.appendChild(oText);

4. Ara hem de connectar l'element amb l'element on vullguem col·locar-lo , en este cas el body :

document.body.appendChild(oP);

CREAR I MANIPULAR NODES

- **element.removeChild(childElement);**
  - Elimina el node fill que rep per paràmetre.

  Exemple :

  childElement.parentNode.removeChild(childElement)

- **element.replaceChild(newChildElem, oldChildElem);**
  - Reemplaça un node fill amb un nou node.

  Exemple :

  oldChildElement.parentNode.removeChild(oldChildElement, newChildElem)

  Per a saber si un node pare té fills es pot utilitzar la funció

  element.hasChildNodes();

TREBALLAR AMB ATRIBUTS

- **element.getAttribute(attrName) :**
  - Torna el valor de l'atribut
- **element.setAttribute(attrName, newValue) :**
  - Canvia el valor de l'atribut
- ***element.removeAttribute(attrName) :***
  - Elimina l'atribut
- **element.hasAttribute(attrName) :**
  - Torna cert si l'element té un atribut amb el nom especificat

ATRIBUTS COM A PROPIETATS

- Amb el DOM HTML, podem gestionar els atributs com a propietats, tenint en compte :
  - Per a l'atribut class, s'ha d'utilitzar className

- Les propietats CSS també es poden accedir de forma directa a través del node DOM
  - *En primer lloc, les propietats CSS* ***s'accedixen a través de l'atribut style de l'element***
  - *En el cas de les propietats CSS amb nom compost, el seu nom es transforma a la notació camelCase típica (s'eliminen els guions i la 1a lletra de cada paraula anirà en majúscula excepte la primera paraula).*

    Exemple : *oDiv.style.margin="10px";*
    oDiv.style.backgroundColor = "gray";

  - *Per a propietats CSS definides amb full CSS es pot utilitzar :* ***window.getComputedStyle(element).getPropertyValue(propietat)*** *- només lectura*

ATRIBUTS COM A PROPIETATS

- **element.classList :** Array de classes CSS de l'element.

  Té mètodes molt útils per a consultar i modificar classes :

  - **element.classList.contains(“classe”**):
    - true si té la classe.
  - **element.classList.replace(“classe1”,”classe2”):**
    - substituïx la classe 1 per la classe “classe2”.
  - **element.classList.add(“classe1”):**
    - Afig la classe “classe1” a l’element.
  - **element.classList.remove(“classe1”):**
    - Elimina la classe “classe1”.
  - **element.classList.toggle(“classe1”):**
    - Si no té “classe1”, l’afig. En cas contrari, l’elimina.

MÈTODES DE TAULES

- Element <table> :
  - i**nsertRow (posición):** Afig una fila en la posició indicada.
  - **deleteRow(posición):** Elimina la fila en la posició indicada.
- Element <tr /> :
  - **cells:** Array de cel·les de l'element <tr />
  - **insertCell(posición)**: Afig una cel·la en la posició indicada.
  - **deleteCell(posición):** Elimina la cel.la en la posició indicada.

GUIA DE REFERÈNCIA

- Object **document** (DOM XML-HTML):
  - https://www.w3schools.com/xml/dom_document.asp
  - https://www.w3schools.com/jsref/dom_obj_document.asp

- Object **element** (DOM XML-HTML):
  - https://www.w3schools.com/xml/dom_element.asp
  - https://www.w3schools.com/jsref/dom_obj_all.asp

- Object **node** :
  - https://www.w3schools.com/xml/dom_node.asp

- Object **nodeList** / **NamedNodeMap** :
  - https://www.w3schools.com/xml/dom_nodelist.asp
  - https://www.w3schools.com/xml/dom_namednodemap.asp

GUIA DE REFERÈNCIA

- Object **text** :
  - https://www.w3schools.com/xml/dom_text.asp
- Object **comment** :
  - https://www.w3schools.com/xml/dom_comment.asp
- Object **attr** (DOM XML-HTML) :
  - https://www.w3schools.com/xml/dom_attribute.asp
  - https://www.w3schools.com/jsref/dom_obj_attributes.asp