

DAI: M3dul 3

Javascript

i DOM

Xavier Noguero

Carles Mateu <http://carlesm.com>

Ci3ncies de la Computaci3 i Intel·lig3ncia Artifici3l

Universitat de Lleida

JavaScript – Història

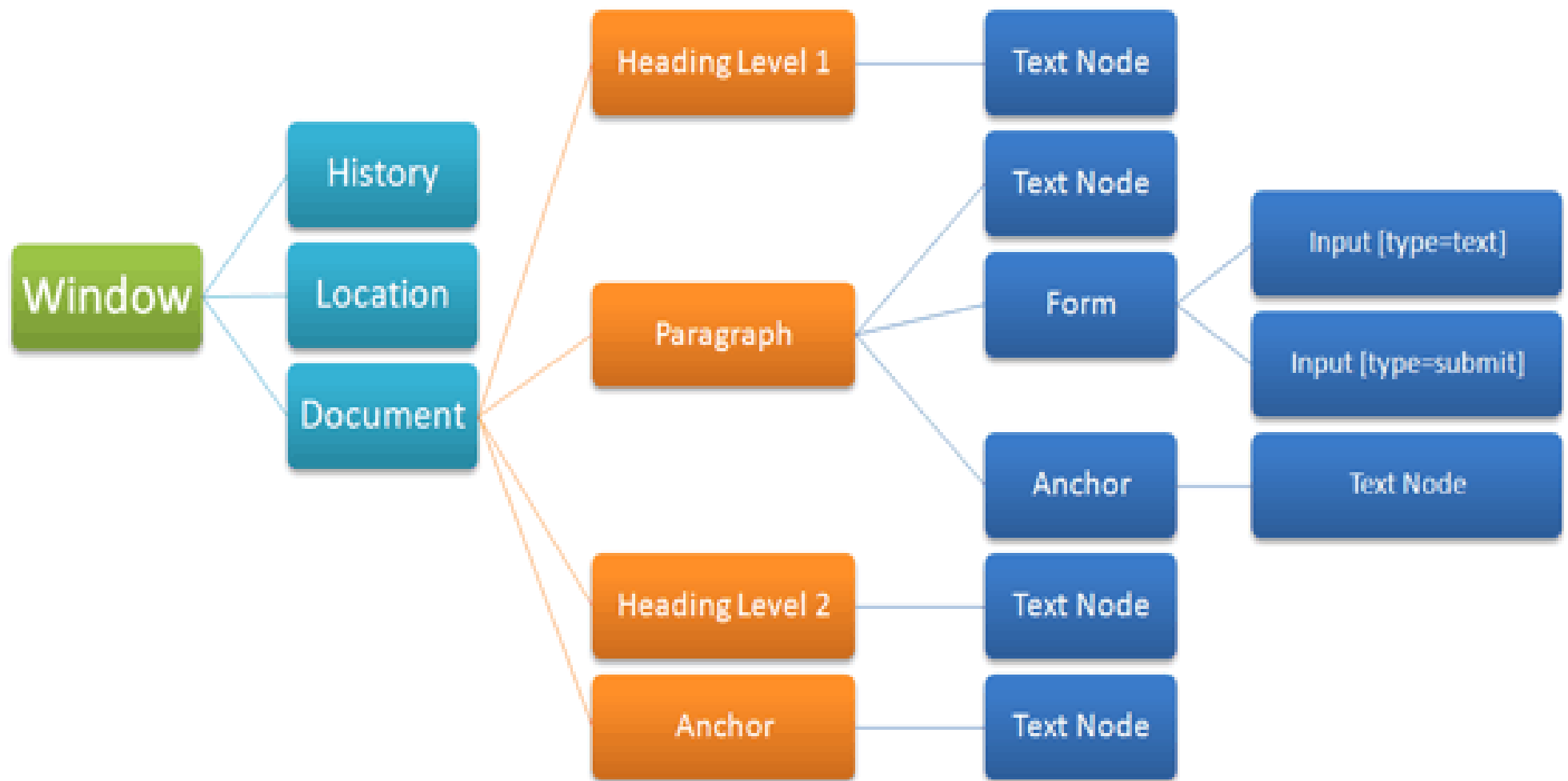
- Basat en l'estàndard ECMAScript (**ECMA-262**)
- Creat per en Brendan Eich, de Netscape
- Originàriament es va anomenar LiveScript
- Aprofitant el momentum de Java, és va reanomenar com a JavaScript

Document Object Model

És l'API a través de la qual Javascript pot interactuar amb els continguts d'una plana Web.

Aquesta interacció permet accedir, recórrer i manipular documents HTML i XML.

Document Object Model



Js – Tipus de dades

- Numbers
- Strings
- Booleans
- Functions
- Objects

Js – Tipus de dades

- Numbers
- Strings
- Booleans
- Objects
- Function
- Array
- Date
- RegExp
- Null
- Undefined

Js – Numbers

- Valors de doble precisió de 64-bits
- Operadors: suma, resta, mòdul...
- Ops. matemàtiques avançades: Math

```
Math.sin(3.5);  
d = Math.PI * r * r;
```

Js – Numbers

Conv. d'String a Number: parseInt()

```
> parseInt("123")
```

```
123
```

```
> parseInt("010")
```

```
8
```

```
> parseInt("010", 10)
```

```
10
```


Js – Numbers

Valor not-a-number (NaN):

```
> parseInt("hello", 10)
```

```
NaN
```

•NaN és tòxic/dolent/caca:

```
> NaN + 5
```

```
NaN
```

```
> isNaN(NaN)
```

```
true
```

Js – Strings

Seqüència de caràcters **UNICODE (16 bits)**

Alguns mètodes:

```
> "hola".length
```

```
4
```

```
> "hola".charAt(0)
```

```
h
```

```
> "hola, mon".replace("hola", "adeu")
```

```
adeu, mon
```

```
> "hola".toUpperCase()
```

```
HOLA
```

Js – Altres tipus

- **null**: sense valor intencionadament,
- **undefined**: amb valor indefinit (creació)
- **boolean**: true o false

0, "", NaN, null, i undefined => false,
tota la resta => true

Operacions booleans: &&, || i !

> !!""

false

Js – Variables

Declaració: var

```
var a;  
var nom = "Maria";
```

Si no assignem cap valor, és undefined

Js – Operadors

Bàsics: +, -, *, / i %

Assignació composta: += i -=

```
x += 1
```

```
x = x + 1
```

Conversions implícites:

```
> "3" + 4 + 5
```

```
345
```

```
> 3 + 4 + "5"
```

```
75
```

Js – Operadors

Comparacions: <, >, <=, >= i !=

```
> "dog" == "dog"
```

```
true
```

```
> 1 == true
```

```
true
```

Per diferenciar els tipus: === i !==

```
> 1 === true
```

```
false
```

```
> true === true
```

```
true
```

Js – Estructures de control

Similar a C/Java (sintaxi i semàntica).

Conditionals:

```
var nom = "Maria";  
if (nom == "Pep") {  
    nom += "!";  
} else if (nom == "Maria") {  
    nom += "!!";  
} else {  
    nom = "!" + nom;  
}  
nom == "Maria!!"
```

Js – Estructures de control

Iteradors

```
while (true) {  
  // un bucle infinit!  
}
```

```
do {  
  var input = get_input();  
} while (inputIsValid(input))
```

```
for (var i = 0; i < 5; i++) {  
  // S'executarà 5 cops  
}
```


Js – Estructures de control

Condiciona in-line

```
var c = (edat > 18) ? "si" : "no";
```

Selecció (Switch/Case)

```
switch(action) {  
    case 'cas1':  
        fes_aixo();  
        break;  
    case 'cas2':  
        fes_allo();  
        break;  
    default:  
        no_facis_res(); }  
}
```

Js – Objectes

Col·leccions de parells nom-valor

nom: etiqueta `String`

valor: pot ser un valor o un altre objecte

Creació:

```
var obj = new Object();
```

```
var obj = {};
```

Js – Objectes

Un cop creat...

```
obj.nom = "Maria";  
var nom = obj.nom;
```

o bé

```
obj["nom"] = "Maria";  
var nom = obj["nom"];
```

Js – Objectes

Inicialització de tot l'objecte:

```
var obj = {  
  nom: "tomaquet",  
  "for": "Maria",  
  details: {  
    color: "vermell",  
    mida: 12  
  }  
}
```

Js – Objectes

Accés als atributs...

```
> obj.detalis.color  
vermell
```

```
> obj["detalls"]["mida"]  
12
```

Js – Arrays

Accés als valors per número

length: Mida (índex més elevat + 1)

```
> var a = new Array();
```

```
> a[0] = "dog";
```

```
> a[1] = "cat";
```

```
> a[2] = "hen";
```

```
> a.length
```

```
3
```

o bé:

```
> var a = ["dog", "cat", "hen"];
```

Js – Arrays

`a.toString()`, `a.toLocaleString()`, `a.concat(item, ..)`,
`a.join(sep)`, `a.pop()`, `a.push(item, ..)`, `a.reverse()`, `a.shift()`,
`a.slice(start, end)`, `a.sort(cmpfn)`, `a.splice(start, delcount, [item]..)`, `a.unshift([item]..)`

- `concat`: retorna un nou array
- `pop`: elimina i retorna l'últim ítem
- `push`: afegeix un o més ítems al final
- `slice`: retorna un subarray
- `sort`: permet ordenar els ítems

Js – Funcions

Poden tenir 0 o més paràmetres:

```
function suma(x, y) {  
  var total = x + y;  
  return total;  
}
```

variable arguments:

```
var num_arg = arguments.length;
```


Js – Funcions

Funcions anònimes:

```
var avg = function() {  
    var suma = 0;  
    for (var i = 0, j = arguments.length; i < j;  
i++)  
    {  
        suma += arguments[i];  
    }  
    return suma / arguments.length;  
}
```

Usant JS a la Web

Dos mecanismes:

- **Embevent el codi dins la mateixa pàgina:**

```
<script language="JavaScript" type="text/javascript">  
<!-- // Begin  
    // Aquí anirà el codi  
// End -->  
</script>
```

Usant JS a la Web

- Tenint el codi en un fitxer apart (amb extensió .js) i després incloure'l a la part del HEAD de l'HTML:

```
<script type="text/javascript"  
src="fitxer.js"></script>
```

Més DOM

window: és l'objecte global que permet accedir al contingut d'una pàgina, el node root de la jerarquia del document.

document: és l'objecte global que conté tot el document HTML (HEAD i BODY), és part de window. Conté els elements HTML/XML.

Ex: si tenim un paràgraf a l'HTML:

```
<p id="intro">El meu paràgraf...</p>
```

Hi podem interactuar:

```
var parag = document.getElementById('intro');
```

Més DOM

`getElementsByTagName`: retorna una llista d'elements HTML de la pàgina.

```
var l = document.getElementsByTagName('ul')[0];  
  
var lis = l.getElementsByTagName('li');  
  
// Per iterar-los...  
for (var i = 0, length = lis.length; i < length; i++) {  
    // Accedim al text de cada elem de la ul:  
    alert( lis[i].firstChild.data );  
}
```

Recurrent el DOM...

Alguns mètodes útils per a moure'ns per la jerarquia de nodes:

Node.childNodes

Node.firstChild

Node.lastChild

Node.parentNode

Node.nextSibling

Node.previousSibling

L'arbre DOM

```
var theDiv = document.getElementsByTagName('div')[0];
```

```
<div>
```

```
var p = theDiv.firstChild;
```

```
<p> This is text </p>
```

```
var ul = p.nextSibling;
```

```
<ul>
```

```
<li>Apple</li> ul.childNodes[0]
```

```
<li>Pear</li> ul.childNodes[1]
```

```
<li>Melon</li> ul.childNodes[2]
```

```
</ul>
```

```
</div>
```

L'arbre DOM

Alguns mètodes per accedir a les propietats dels elements HTML:

- `Node.innerHTML`
- `Node.nodeName`
- `Node.nodeValue`
- `Node.attributes`

L'arbre DOM

Per afegir i eliminar nodes:

Node.appendChild(*node*)

```
var nouP = document.createElement('p');  
var text = document.createTextNode('Hola!');  
nouP.appendChild(text);  
document.getElementById('unId').appendChild(nouP);
```

Node.removeChild(*node*)

```
var unId = document.getElementById('unId');  
unId.removeChild(unId.lastChild);
```

Node.insertBefore(*newNode*, *oldNode*)

Node.replaceChild(*newNode*, *oldNode*)

Js – Events

Accions que poden ser detectades i processades via JavaScript

Exemples:

- **Un clic d'un ratolí**
- **Una pàgina web carregant-se**
- **Passar per sobre d'un element html**
- **Escriure text en un camp de formulari**
- **Enviar les dades d'un formulari**

Referència d'events

Js – Events

En fer clic sobre un enllaç:

```
<a href="url.html" onclick="alert('He fet clic!')">
```

En modificar el contingut d'un camp de text:

```
<input type="text" id="nom"
onchange="toUpperCase(this.id)">
```

En fer Submit a un formulari:

```
<form action="url" onsubmit="return
checkForm()">
```

Referències JavaScript i DOM

Developer Mozilla

<http://developer.mozilla.org/es/docs/JavaScript>

W3schools Javascript tutorial

<http://www.w3schools.com/js/default.asp>

JavaScript Event Reference

http://www.w3schools.com/jsref/jsref_events.asp

HTML DOM Reference

http://www.w3schools.com/html/dom/dom_reference.asp

JConsole

<http://www.jconsole.com>