

# JavaScript

---

UD03 Objectes predefinitos del llenguatge

# Objectes

---

- Els objectes es creen per mitjà de la paraula clau **new** seguida del nom de la classe de la qual volem crear una instància :

```
var oObject = new Object();  
var oStringObject = new String();
```

La primera línia crea una nova instància d'Object i l'emmagatzema en la variable oObject; la segona línia crea una nova instància de String i l'emmagatzema en la variable oStringObject.

Els parèntesis no són necessaris quan el constructor no requereix arguments.

- Javascript conté una sèrie de classes predefinides que estudiarem en aquest tema.

# La classe Array.

---

- Un array és una variable que conté distints valors.
  - Podem referenciar aquests valors indicant la seua posició en l'array entre claudàtors començant amb 0.
  - Hem de tindre en compte que en Javascript un array pot contindre al mateix temps diferents tipus de dades : number, string, boolean, object, etc en les seues distintes posicions.
- En ECMAScript hi ha una classe Array.
  - Es pot crear un objecte Array d'aquesta manera :  
`var aValues = new Array();`
  - El constructor pot rebre 0 paràmetres (array buit), 1 paràmetre (el tamany de l'array), o en qualsevol altre cas, es crearà un array amb els elements rebuts i es pot calcular el seu tamany amb la propietat **length**.

```
var aColors = new Array(3);  
aColors[0] = "red";  
aColors[1] = "green";  
aColors[2] = "blue";
```

```
var aColors = new Array ("red", "green", "blue");  
alert (aColors[1]);      //torna "green";
```

# La classe Array.

---

- El tamany de un array pot augmentar i disminuir. És per això que si volem afegir un altre element a l'array anterior, podem fer-ho en la següent posició lliure :

```
var aColor  
  
var aColors = new Array ("red","green","blue");  
alert(aColors.length); // torna 3  
aColors[3] = "purpple";  
alert(aColors.length) // torna "4"
```

- Si afegim un valor en la posició 25 s'emmagatzemarien valors **undefine** de la posició 4 a la 24 i en la posició 25 s'introduiria el valor corresponent.

```
var aColors = new Array ("red","green","blue");  
alert(aColors.length); // torna 3  
aColors[25] = "purpple";  
alert(aColors.length) // torna "26"
```

# La classe Array. Mètodes

---

- Guia de referència de la classe Array per a consultar
  - [https://www.w3schools.com/jsref/jsref\\_obj\\_array.asp](https://www.w3schools.com/jsref/jsref_obj_array.asp)

• **join("separador")** : Mètode de l'objecte Array que **convertix un array en una cadena** de text separant cada element pel separador.

```
alert(aColors.join("|")); // torna "red|green|blue".
```

• **split()** : Mètode de l'objecte Array que convertix una cadena de text en un array. Accepta un únic paràmetre, la cadena de text que es considera el separador dels elements.

```
var sColors = "red, green, blue";  
var sColors = sColors.split(",");
```

Si s'especifica una cadena buida com a separador, el mètode torna un array en què cada element és igual a un caràcter de la cadena.

# La classe Array. Mètodes

---

- **toString()** torna una cadena de text especial, que es construeix mitjançant la invocació del mètode toString() en cada element de l'array i després d'això amb la combinació dels mateixos per mitjà de comes.

```
var aColors = ["red", "green", "blue"]  
alert(aColors.toString()); //torna "red, green, blue"
```

- **push()** - **unshift()** : **push** afig els ítems proporcionats al final de l'array, **unshift** al principi  
**pop()** - **shift()** : **pop** extrau i torna l'últim ítem de l'array, **shift** el primer ítem desplaçant la resta
- **reverse()** : invertix l'array
  - consultar la guia de referència per a més informació

# La classe String.

---

- Guia de referència de la classe String per a consultar
  - [https://www.w3schools.com/jsref/jsref\\_obj\\_string.asp](https://www.w3schools.com/jsref/jsref_obj_string.asp)
- En cas de crear una cadena de text amb "=", implícitament aquesta cadena es convertix en un objecte String amb les propietats i mètodes predefinits.

```
var cadena = "...";  
var cadena = new String("...");
```

- Tenim una propietat molt important que és **length**. Aquesta propietat ens diu quants elements (caràcters) té la cadena de text.
- Per a posar caràcters especials dins d'una cadena de text s'han "d'escapar" mitjançant el caràcter \.
  - Tenim \' \'\' \n (salt de línia) \\ (caràcter \) i \t (tabulador)

# La classe String. Mètodes.

---

- **toLowerCase()** / **toUpperCase()**: torna la cadena de text convertida a minúscules / majúscules.
- **concat(cadena)**: torna l'objecte amb el valor de cadena de text concatenat al final. També amb **+=**
- **charAt(posicion)**: torna el caràcter que es trobe en la posició sol.licitada. Hem de tindre en compte que les posicions comencen a comptar des de zero.
- **indexOf(text,[index])** / **lastIndexOf(text,[index])** torna la primera / l'última posició on es troba el text buscat, començant a buscar des de la posició index. **lastIndexOf** busca “cap arrere”.
  - **search i match** busquen expresions regulars que verem en el tema de formularis
- **includes(cadena)** : indica si s'inclou la cadena,  
**startsWith(cadena)** indica si comença amb la cadena i  
**endsWith(cadena)** indica si acaba amb la cadena.
  - Tornen true o false



# La classe String. Mètodes.

---

- **replace(text1,text2):** busca totes les cadena de text text1 i les reemplaça per text2.
- **split(caracter, [trossos]):** separa en un array la cadena de text mitjançant un caràcter separador. trossos indica el màxim de separacions. Si no s'indica, es faran totes les separacions possibles.
- **substring(inici, [fi]):** torna la subcadena situada entre la posició inici i la posició fi. Si fi no s'indica, es pren com a valor el final de la cadena. Veure mètodes semblants **slice** i **substr**
- **trim(cadena) :** torna la cadena llevant els espais del principi i del final.
  - consultar la guia de referència per a més informació

# La classe Date.

---

- Guia de referència de la classe Date per a consultar
  - [https://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](https://www.w3schools.com/jsref/jsref_obj_date.asp)
- La classe Date emmagatzena la data com el nombre de milisegons transcorreguts des de les 12 AM de l'1 de gener de 1970 GMT.
- Per a crear un nou objecte Date s'utilitza l'operador **new** i admet múltiples formats (veure guia de referència) :  
*var d = new Date(); ( Crea l'objecte amb data i hora actuals )*  
*var d = new Date(any,mes (0..11),dia,hora,minut,second.);*
- Utilitzar **el mètode estàtic UTC** per a crear una data  
*var d = new Date(Date.UTC(2004, 1, 5));*  
*//indica la fecha 05/01/2004*  
*var d = new Date(Date.UTC(2004, 1, 5,13,5));*  
*//indica la fecha 05/01/2004: 13 horas y 5 minutos.*

# La classe Date. Mètodes

---

- **valueOf():** Torna la representació en mil·lisegons de la data.
- **toString():** Torna una cadena de text en un format llegible per als humans i específic de la implementació.
  - **toDateString():** Torna només la part de la data de Date (el mes, el dia i l'any) en un format dependent de la implementació.
  - **toTimeString():** Torna només la part temporal de Date (h, m, s y zona horària) en un format dependent de la implementació
  - **toLocaleString():** Torna la data i l'hora de Date en format específic de la configuració regional.
  - **toLocaleTimeString():** Torna només la part temporal de Date (h, m, s y zona horària) en un format específic de la configuració regional.
  - **toUTCString():** Torna la data UTC de una data en un format específic de la configuració regional.
  - **toGMTString():** Torna una cadena de text que utilitza les convencions d'Internet amb la zona horària GMT.

# La classe Date. Mètodes getter

---

`getDate():`

Torna el dia del mes actual com un sencer entre 1 i 31.

`getDay():`

Torna el dia de la setmana actual com un sencer entre 0 i 6.

`getFullYear():`

Torna l'any com un nombre de quatre dígit.

`getHours():`

Torna l'hora del dia actual com un sencer entre 0 i 23.

`getMinutes():`

Torna els minuts de l'hora actual com un sencer entre 0 i 59.

`getMonth():`

Torna el mes de l'any actual com un sencer entre 0 i 11.

`getSeconds():`

Torna els segons del minut actual com un sencer entre 0 i 59.

`getTime():`

Dtorna el temps trascorregut en mil.lisegons des de l'1 de gener de 1970 fins al moment actual.

# La classe Date. Mètodes setter

---

- `setDate(día_mes)`: Posa el dia del mes actual en l'objecte Date que estem utilitzant.
- `setHours(horas)`: Posa l'hora del dia actual en l'objecte Date que estem utilitzant.
- `setMinutes(minutos)`: Posa els minuts de l'hora actual en l'objecte Date que estem utilitzant.
- `setMonth(mes)`: Posa el mes de l'any actual en l'objecte Date que estem utilitzant.
- `setSeconds(segundos)`: Posa els segons del minut actual en l'objecte Date que estem utilitzant.
- `setTime(milisegundos)`: Posa els mil·lisegons que han passat des de l'1 de gener de 1970 en el objecte Date que estem utilitzant.
- `setFullYear(año)`: Posa l'any actual en el objecte Date que estem utilitzant.

- consultar la guia de referència per a més informació

# L'objecte Math.

---

- Guia de referència de l'objecte Math per a consultar
  - [https://www.w3schools.com/jsref/jsref\\_obj\\_math.asp](https://www.w3schools.com/jsref/jsref_obj_math.asp)
- L'objecte Math es l'objecte incorporat que conté les fórmules matemàtiques.  
**MÉTODOS**  
–
- **Math.min()** y **Math.max()** s'utilitzen per a determinar el major o menor d'un grup de números.
- **Math.abs()**: Torna el valor absolut d'un número
- **Math.ceil()**: Arredonix cap al valor superior més pròxim.
- **Math.floor()**: Arredonix cap al valor inferior més pròxim.
- **Math.round()**: Arredonix cap al valor superior si el número està a més de mig camí del següent valor i cap al valor inferior en cas contrari.

# L'objecte Math.

---

- **Math.pow(base,potència):** S'utilitza per a elevar un número a una determinada potència

```
var iNum = Math.pow(3,2);
```

**Math.sqrt():** Torna la l'arrel quadrada d'un determinat número.

```
var iNum = Math.sqrt(4);
```

○

**Math.random():** Torna un número aleatori entre 0 i 1, sense incloure 0 ni 1.

```
var iNum = Math.floor(Math.random() * 10 + 1)
// genera un número aleatori entre 1 i 10
```

**Math.PI:** emmagatzema el número Pi.

- **Math.E:** emmagatzema el número de Euler.

**Math.LN2 / Math.LN10 :** logaritme neperià de 2 / 10.

**Math.LOG2E / Math.LOG10E :** logaritme en base 2 / 10 del número d'Euler.

consultar la guia de referència per a més informació

# La classe Number.

---

- Guia de referència de la classe Number per a consultar
  - [https://www.w3schools.com/jsref/jsref\\_obj\\_number.asp](https://www.w3schools.com/jsref/jsref_obj_number.asp)
- S'utilitza per a emmagatzemar un número; en Javascript s'emmagatzema sempre en coma flotant de doble precisió (double)
- Es pot crear un objecte Number amb l'operador **new**  
*num = new Number(123);*
- **Mètodes i propietats**
  - **Infinity / -Infinity**
  - **Number.POSITIVE\_INFINITY / Number.NEGATIVE\_INFINITY**
    - Representa el número infinit



# La classe Number.

---

- **NaN / Number.NaN / isNaN()**      Not a Number (ja vist)
- **.toString(base)** : Convertix un número a una altra base
- **.toValue()** : Torna el valor primitiu
- **parseInt() / parseFloat / Number()** : (ja vist)
- **.toFixed(num)** : Torna una cadena de text del número amb el nombre de decimals especificats en num
- Representació 0xNNNNN : Notació hexadecimal  
Representació NN.NNNeEEE: Notació científica    NN.NNN\*10<sup>EEE</sup>

Es recomanable no utilitzar el object Number (lent) i utilitzar en el seu lloc el valor primitiu

consultar la guia de referència per a més informació

# Objectes BOM (Browser Object Model)

## Objecte Window

---

L'objecte Window és un objecte que representa a una finestra del navegador, té propietats i mètodes.

- Guia de referència de l'objecte Window per a consultar
  - [https://www.w3schools.com/jsref/obj\\_window.asp](https://www.w3schools.com/jsref/obj_window.asp)  
( s'ha de comprovar la compatibilitat entre navegadors )
- La resta d'objectes BOM són membres de l'objecte Window (navigator, screen, history, location), així com l'objecte document, però no és necessari posar window. si els volem utilitzar.

# Objectes BOM (Browser Object Model)

## Objecte Window

---

### **PROPIETATS**

**.name** : Nom de la finestra

**.outerHeight .outerWidth**

**.innerHeight .innerWidth** : Tamany de la finestra

**.pageXOffset .pageYOffset** : Longitud del scroll

**.screenX .screenY** :

Distància entre el cantó esquerra dalt i la finestra

**.parent** : Finestra pare

**.closed** : Indica si la finestra està tancada

# Objectes BOM (Browser Object Model)

## Objecte Window

---

### **MÈTODES**

**.open(url, name, info)** : Obri una nova finestra, torna l'id

**.close()** : Tanca la finestra

**.resizeBy(width, height);**

**.resizeTo(width, height)** : Redimensiona la finestra

**.moveBy(width, height)**

**.moveTo(width, height)** : Mou la finestra

**.scrollBy(width, height)**

**.scrollTo(width, height)** : Mou le scroll de la finestra

**.focus()** : Posa el focus en la finestra

**.print()** : Imprimix el contingut de la finestra

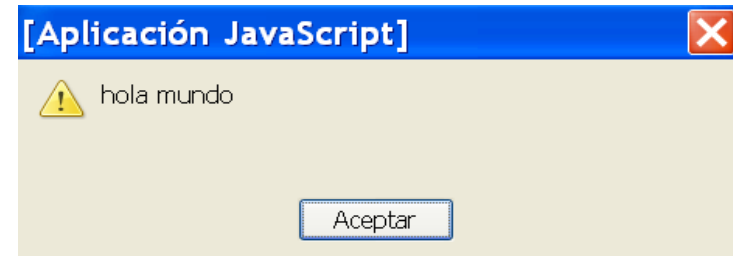
**.stop()** : Deté la càrrega i execució de la pàgina de la finestra

# Objectes BOM (Browser Object Model)

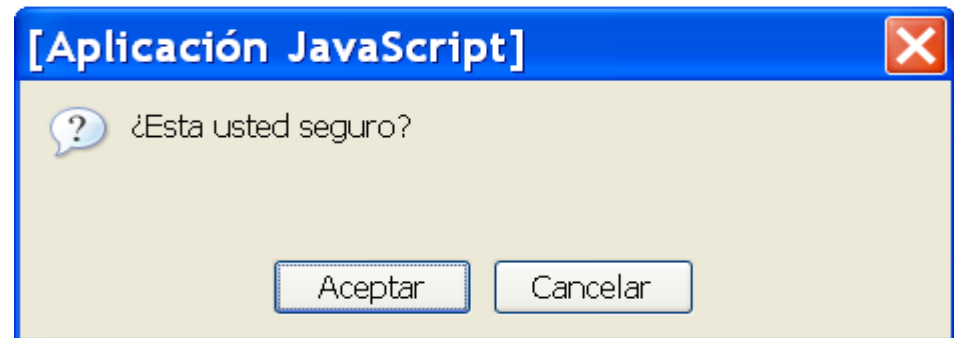
## Objecte Window

---

- **alert()**: El mètode mostra una finestra emergent a l'usuari i roman activa fins que l'usuari pressiona el botó d'Acceptar.



- **confirm()** : Mostra un missatge en una finestra emergent amb dos botons : Acceptar i Cancel·lar.  
En funció del botó que pressione l'usuari la funció **confirm()** tornarà true (Acceptar) o false (Cancel·lar)



# Objectes BOM (Browser Object Model)

## Objecte Window

---

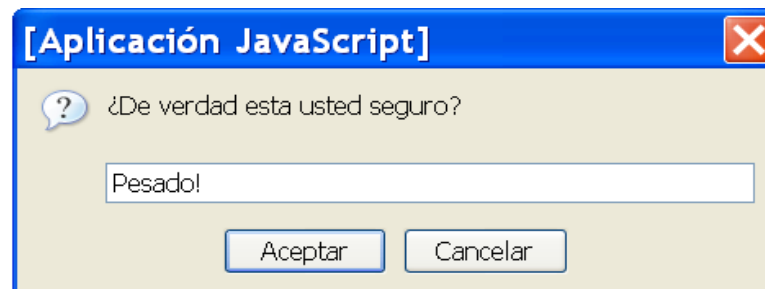
- **prompt():** S'utilitza per a demanar dades a l'usuari

Té dos arguments, el primer serà el missatge que es veurà en la pantalla i el segon és el valor per defecte que apareixerà en la caixa de text que es mostrarà a l'usuari.

Proporciona també dos botons d'*Acceptar* y *Cancel·lar*. Pressionant el botó *Acceptar* la funció torna el valor de la caixa de text.

Si polsem *Cancel·lar* la funció torna null.

```
prompt("¿De verdad esta usted seguro?", "Pesado!");
```



## L'objecte Window. Intervals i temps d'espera.

- Per a definir un temps d'espera s'utilitza el mètode **setTimeout()** de window, mètode qu'accepta dos arguments : El **códi a executar** i el nombre de **mil·lisegons a esperar** abans de la execució. El primer argument pot ser una cadena de còdi o un punter a una funció.

- **setTimeout()** torna un id.

```
function hola(){ alert("hola mundo!"); }  
var timeout = setTimeout(hola, 1000);
```

- Per a cancel·lar un temps d'espera pendent, s'ha d'utilitzar el mètode **clearTimeout()** i **passar l'id**

```
clearTimeout(timeout);
```

## L'objecte Window. Intervals i temps d'espera.

---

- Els intervals funcionen de la mateixa forma amb l'excepció que es repetix el codi indefinidament a intervals de temps concrets. Per a definir un interval, utilitzem ***setInterval()*** amb els mateixos arguments que ***setTimeout()*** : El còdig a executar i el nombre de mil·lisegons a esperar entre execucions.

- **setInterval() torna un id..**

```
function hola(){ alert("hola mundo!"); }  
setInterval(hola, 1000);
```

- Per a cancel·lar l'interval, s'ha d'utilitzar el mètode **clearInterval()** i passar l'id

```
clearInterval(timeout);
```

*consultar la guia de referència per a més informació*



# Objetos BOM (Browser Object Model)

## Objeto Navigator

---

L'objecte Navigator és un objecte que representa al navegador

- Guia de referència de l'objecte Navigator per a consultar
  - [https://www.w3schools.com/jsref/obj\\_navigator.asp](https://www.w3schools.com/jsref/obj_navigator.asp)  
( s'ha de comprovar la compatibilitat entre navegadors )

### **PROPIETATS**

- **navigator.appCodeName, navigator.appName, navigator.version, navigator.product**  
Informació sobre el navegador
- **navigator.language :**

# Objetos BOM (Browser Object Model)

## Objeto Navigator

---

- **navigator.platform** :  
Informació sobre la plataforma on està instal·lat el navegador
- **navigator.userAgent** : Capçalera del navegador que s'envia
- **navigator.online** : Saber si l'usuari està connectat a Internet
- **navigator.cookies** : Torna true o false
- **navigator.geolocation** :  
Torna un objecte de tipus "Geolocation" indicant on s'ubica l'usuari
- **navigator.javaEnabled()** : Torna true o false

consultar la guia de referència per a més informació

# Objetos BOM (Browser Object Model)

## Objeto Screen

---

L'objecte Screen és un objecte que representa la pantalla de l'usuari

- Guia de referència de l'objecte Screen per a consultar
  - [https://www.w3schools.com/jsref/obj\\_screen.asp](https://www.w3schools.com/jsref/obj_screen.asp)  
( s'ha de comprovar la compatibilitat entre navegadors )

### **PROPIETATS**

- **Screen.width, screen.height, screen.availWidth, screen.availHeight**  
Les dos últimes no tenen en compte la barra de tasques del sistema operatiu
- **navigator.colorDepth, navigator.pixelDepth**  
Profunditat i resolució de color

consultar la guia de referència per a més informació

# Objetos BOM (Browser Object Model)

## Objeto History

---

L'objecte History és un objecte que emmagatzema informació de les urls posades en una finestra del navegador

- Guia de referència de l'objecte History per a consultar
  - [https://www.w3schools.com/jsref/api\\_history.asp](https://www.w3schools.com/jsref/api_history.asp)  
( s'ha de comprovar la compatibilitat entre navegadors )

### **PROPIETATS i MÈTODES**

- **history.length**
- **history.back(), history.forward(), history.go(number), history.go(url)**
  - Anar cap a arrere, cap avant, “number” posicions o a una url
  - consultar la guia de referència per a més informació

# Objetos BOM (Browser Object Model)

## Objeto Location

---

L'objecte Location és un objecte que emmagatzema informació de l'url actual

- Guia de referència de l'objecte Location per a consultar
  - [https://www.w3schools.com/jsref/obj\\_location.asp](https://www.w3schools.com/jsref/obj_location.asp)  
( s'ha de comprovar la compatibilitat entre navegadors )

### **PROPIETATS i MÈTODES**

- **location.href**
- **location.hostname, location.pathname, location.protocol, location.host, location.origin**
  - location.host = location.hostname + location.pathname
  - location.origin = location.protocol + location.host

# Objetos BOM (Browser Object Model)

## Objeto Location

---

- **location.hash** : **protocol://hostname/pathname#hash**
  - **location.search** : Torna la querystring d'una url  
**protocol://hostname/pathname?querystring**
  - **location.assign(), location.replace()** :  
Substituïx la pàgina actual per una altra. **replace()** no emmagatzena la url anterior en l'objecte **history**
  - **location.reload()** : Recarrega la pàgina
- consultar la guia de referència per a més informació