

JavaScript



Índex

VERSIONS.....	3
A ON I COM INCLOURE'L.....	3
VARIABLES.....	4
OPERADORS ARITMÈTICS.....	5
OPERADORS D'ASSIGNACIÓ.....	6
OPERADOR CONDICIONAL.....	6
ESTRUCTURES DE CONTROL.....	7
if.....	7
if..else.....	7
if..else if..else.....	8
switch.....	9
while loop.....	10
do..while loop.....	10
for loop.....	11
for..in.....	11
FUNCIONS.....	12
Definició d'una funció.....	12
Funcions que retornen valors.....	12
FINESTRES EMERGENTS.....	13
Alerta.....	13
Confirmació.....	13
Demandar informació.....	14
EVENTS.....	15
FUNCIONS ESPECIALS.....	16

JavaScript és un llenguatge interpretat, és a dir, no necessita ser compilat, utilitzat principalment en pàgines web, amb una sintaxis semblant a la del llenguatge Java i el llenguatge C.

Al contrari que Java, JavaScript no és un llenguatge orientat a objectes pròpiament dit, ja que no disposa d'herència, és més aviat, un llenguatge basat en prototipus ja que les noves classes es generen clonant les classes base (prototipus) i estenent la seva funcionalitat. (*es.wikipedia.org*)

Els programes JavaScript van inclosos en els documents **HTML** o en tot aquell que acabi traduïnt-se en HTML en el navegador client ja sigui **ASP**, **PHP**,... i s'encarreguen de realitzar accions com pot ser demanar dades, confirmacions, mostrar missatges, crear animacions, comprovar camps, ...

VERSIONS

El programa que interpreta els programes JavaScript és el navegador, per tant, si el navegador no suporta JavaScript no es podran executar les funcions escrites en aquest llenguatge.

Per a evitar incompatibilitats entre els diferents navegadors existents al mercat, el **World Wide Web Consortium** va dissenyar el estàndard **Document Object Model** (DOM), que incorporen **Konqueror**, les versions 6 de **Internet Explorer** i **Netscape Navigator**, la versió 7 de **Opera** i **Mozilla** des de la seva primera versió.

A ON I COM INCLOURE'L

Existeixen diferents formes d'incloure llenguatge JavaScript en una pàgina.

La forma més freqüent de fer-ho és utilitzant la directiva **<script>** en un document HTML (es poden incloure tantes directives **<script>** com es vulgui en un document). El format és el següent:

```
<script type="text/javascript">  
// codi JavaScript;  
</script>
```

Si el que es vol és incloure un fitxer extern on està escrit el codi que es vol executar, s'utilitza el paràmetre **src**: **<script type="text/javascript" src="[URL]"></script>**

[URL]: és el camí absolut o relatiu que indica on es troba el fitxer amb codi JavaScript. El seu nom ha de tenir l'extensió **.js** i no ha de contenir l'etiqueta **<script>**

Es pot incloure també codi JavaScript com a resposta a algun event:

```
<input type="submit" onclick="alert('Acabes de fer click');return false;"  
value="Click">
```

Els scripts poden incloure's com a comentaris per assegurar que el seu codi no és "vist" per navegadors antics que no reconeixen JavaScript i d'aquesta forma evitar errors.

```
<script ....>
  <!--
    codi JavaScript;
  //-->
</script>
```

VARIABLES

JavaScript és case sensitive, per tant, diferencia majúscules de minúscules i això implica que la variable Hola no és la mateixa que la variable hola.

El nom de les variables ha de seguir les següent regles:

- Ha de començar per una **lletra** o el caràcter `_`
- Els caràcters següents poden ser lletres, números o `_`

var La Variable, 123Prova, \$variable, for; --> DEFINICIONS INCORRECTES
 var La_Variable, P123rova, _123, hoLaaa; --> DEFINICIONS CORRECTES

Les variables es poden declarar: **var nomvar = valor;**
 nomvar = valor;

Per assignar un valor a una variable: **var nomvar = "Hola";**
 nomvar = "Hola";

Quan es declara una variable dins d'una funció, la variable només pot ser referenciada dins d'aquesta funció. Quan es surt de la funció la variable és destruïda. Aquestes són les **variables locals**.

Es poden tenir variables locals amb el mateix nom en diferents funcions ja que només són reconegudes dins de les funcions on són declarades.

Si es declara una variable fóra de les funcions, totes les funcions de la pàgina poden fer referència a ella, són les **variables globals**. El temps de vida d'aquestes variables és des de que són declarades fins que la pàgina es tanca.

JavaScript reconeix 6 tipus de valors diferents:

Numèrics	Enters o coma flotant.
Lògics (booleans)	True o False.
Cadenes	Han d'anar entre cometes senzilles o dobles.
Objectes	Obj = new Object();
Nuls	Null
Indefinits	Variable creada però no se li ha assignat cap valor.

Té la peculiaritat de ser un llenguatge dèbilment "tipat", això significa que una variable pot canviar de tipus durant la seva vida:

```
UnaVar = 4;
UnaVar = "Hola radiola";
```

OPERADORS ARITMÈTICS

Prenen valors numèrics (literals o variables) com a operands i retornen un sol valor numèric.

Operador	Descripció	Exemple	Resultat
+	Suma	x=2 y=2 x + y	4
-	Resta	x=5 y=2 x - y	3
*	Multiplicació	x=5 y=4 x * y	20
/	Divisió	15/5 5/2	3 2.5
%	Mòdul	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	6
--	Decrement	x=5 x--	4

Exemple:

```
<script language="JavaScript">
  a = 8;
  b = 3;
  c = 3;
  document.write( (a == b) && (c > b) );
  document.write("<br>");
  document.write( (a == b) || (b == c) );
  document.write("<br>");
  document.write( !(b <= c) );document.write("<br>");
</script>
```

OPERADORS D'ASSIGNACIÓ

Comparen els seus operands i retorna un valor lògic basat en si la comparació és certa o no. Els operands poden ser numèrics o cadenes.

Operador	Descripció	Exemple
==	és igual a	5==8 retorna false
===	és igual a (valida valor i tipus)	X=5 y="5" x==y retorna true x===y retorna false
!=	no és igual	5!=8 retorna true
>	és major que	5>8 retorna fals
<	és menor que	5<8 retorna true
>=	és major o igual que	5>=8 retorna false
<=	és menor o igual que	5<=8 retorna true

Exemple:

```
<script language="JavaScript">
  a = 8;
  b = 3;
  a += 3;
  document.write(a);document.write("<br>");
  a -= 2;
  document.write(a);document.write("<br>");
  b *= 2;
  document.write(b);
</script>
```

OPERADOR DE CADENES

Es poden concatenar cadenes mitjançant l'operador +:

Exemple:

```
txt1="Que tinguis";    -->  txt3="Que tinguis un bon dia"
txt2="un bon dia";
txt3=txt1+txt2;
```

OPERADOR CONDICIONAL

Assigna un valor a una variable en funció del resultat d'una condició:

```
nomvar=(condició)?valorCert:valorFals;
```

Exemple: benvinguda=(visitant=="PRES")?"Hola President":"Hola";

ESTRUCTURES DE CONTROL

if...

Si es vol executar un codi només si la condició indicada és certa:

```
if (condició)
{
    codi que s'executa si la condició és certa
}
```

Exemple:

```
<script type="text/javascript">
    //Escriu "Hora de dinar!" si són les 13 hores
    var d=new Date()
    var hora=d.getHours()

    if (hora==13)
    {
        document.write("<b>Hora de dinar!</b>")
    }
</script>
```

if...else

Si es vol executar un codi si la condició indicada és certa o un codi alternatiu en cas de que sigui falsa:

```
if (condició)
{
    codi que s'executa si la condició és certa
}
else
{
    codi que s'executa si la condició és falsa
}
```

Exemple:

```
<script type="text/javascript">
    //Escriu "Bon dia!" si és abans de les 12 hores i "Bona tarda!"
    //en cas contrari
    var d=new Date()
    var hora=d.getHours()

    if (hora<12)
    {
        document.write("<b>Bon dia!</b>")
    }
}
```

```
    else
    {
        document.write("<b>Bona tarda!</b>")
    }
</script>
```

if...else if...else

Si es vol executar un codi en funció de la condició que és certa:

```
if (condició1)
{
    codi que s'executa si la condició1 és certa
}
else if (condició2)
{
    codi que s'executa si la condició2 és certa
}
else
{
    codi que s'executa si la condició1 i la condició2 no són certes
}
```

Exemple:

```
<script type="text/javascript">
    var d = new Date()
    var hora= d.getHours()

    if (hora<12)
    {
        document.write("<b>Bon dia!</b>")
    }
    else if (hora>12 && hora<20)
    {
        document.write("<b>Bona tarda!</b>")
    }
    else
    {
        document.write("<b>Bona nit!</b>")
    }
</script>
```


switch

Si es vol executar un codi en funció dels possibles valors d'una expressió (normalment una variable):

```
switch(expressió)
{
case 1:
    executa codi bloc 1
    break
case 2:
    executa codi bloc 2
    break
default:
    codi que s'executarà si la expressió és diferent de 1 i 2
}
```

El valor de l'expressió s'avalua una vegada i llavors es compara amb el valor de cadascuna de les opcions case de l'estructura. Si hi ha correspondència s'executa el codi corresponent al bloc.

S'utilitza **break** per a que no s'executi codi que no pertany a la condició que coincideix amb el valor de l'expressió.

Exemple:

```
<script type="text/javascript">
    //Escriu el nom del dia que correspon
    //Compte que Diumenge = 0, Dilluns = 1,...

    d=new Date()
    elDia=d.getDay()
    switch (elDia)
    {
    case 5:
        document.write("Finalment Divendres")
        break
    case 6:
        document.write("Super Dissabte")
        break
    case 0:
        document.write("Tranquil Diumenge")
        break
    default:
        document.write("Estic desitjant que arribi el cap de setmana!")
    }
</script>
```

while loop

S'utilitza quan es vol executar un codi mentre es compleixi una condició:

```
while(condició)
{
    codi que s'executa mentre la condició sigui certa
}
```

Exemple:

```
<script type="text/javascript">
    var i=0
    while (i<=10)
    {
        document.write("El nombre és " + i)
        document.write("<br />")
        i=i+1
    }
</script>
```

do...while loop

S'utilitza quan es vol executar un codi com a mínim una vegada abans de verificar si la condició és certa. En cas de que sigui certa s'anirà executant mentre es compleixi la condició:

```
do
{
    codi que s'executa
}
while(condició)
```

Exemple:

```
<script type="text/javascript">
    var i=0
    do
    {
        document.write("El número és " + i)
        document.write("<br />")
        i=i+1
    }
    while (i<0)
</script>
```

for loop

S'utilitza quan es vol executar un codi un nombre conegut de vegades:

```
for (var=valorinicial;var<=valorfinal;var=var+increment)
{
    codi que s'executa
}
```

Exemple:

```
<script type="text/javascript">
    var i=0
    for (i=0;i<=10;i++)
    {
        document.write("El nombre és " + i)
        document.write("<br />")
    }
</script>
```

for...in

S'utilitza quan recòrrer els elements d'una taula o les propietats d'un objecte:

```
for (variable in objecte)
{
    codi que s'executa
}
```

Exemple:

```
<script type="text/javascript">
    var x
    var cotxes = new Array()
    cotxes[0] = "Saab"
    cotxes[1] = "Volvo"
    cotxes[2] = "BMW"

    for (x in cotxes)
    {
        document.write(cotxes[x] + "<br />")
    }
</script>
```

FUNCIONS

Una funció conté codi que s'executa quan es fa una crida a la funció. La crida es pot fer directament o mitjançant un event d'un objecte de la pàgina.

Les funcions es poden definir en el **<HEAD>** o a la secció del **<BODY>** d'un document HTML. No obstant, si es vol assegurar que la funció ha estat carregada pel navegador abans de que es cridi per primera vegada, s'ha de posar al **<HEAD>**.

Exemple:

```
<html>
<head>
  <script type="text/javascript">
    function mostraMissatge()
    {
      alert("Hola Món!")
    }
  </script>
</head>
<body>
  <form>
    <input type="button" value="Fes-me clic!" onclick="mostraMissatge()" >
  </form>
</body>
</html>
```

Definició d'una funció

La sintaxis per definir una funció és:

```
function nomFunció ([var1, var2, ..., varn])
{
  codi a executar
}
```

Funcions que retornen valors

Les funcions que retornen valors utilitzen la instrucció return:

Exemple:

```
function prod(a,b)
{
  x=a*b
  return x
}
```

Quan es fa la crida de la funció s'han d'indicar els valors dels paràmetres:

Exemple: producte = prod(4,5)

FINESTRES EMERGENTS

Es poden crear 3 classes de finestres emergents: d'**alerta**, de **confirmació** i per **demanar informació**.

Alerta

S'utilitzen quan es vol tenir l'assegurança de que l'usuari està informat d'algun fet. L'usuari només pot acceptar la informació. `alert("text")`

Exemple:

```
<html>
<head>
  <script type="text/javascript">
    function mostra_alerta()
    {
      alert("Aquest missatge està" + '\n' + "dividit en diferents línies !")
    }
  </script>
</head>
<body>
  <input type="button" onclick="mostra_alerta()" value="Mostra Alerta" >
</body>
</html>
```

Confirmació

S'utilitzen quan es vol tenir la confirmació per part de l'usuari. Se li donen 2 opcions, Acceptar o Cancel·lar. Si l'usuari Accepta la finestra retorna true, en cas contrari, retorna false. `confirm("text")`

Exemple:

```
<html>
<head>
  <script type="text/javascript">
    function mostra_confirmacio()
    {
      if (r==true)
      {
        document.write("Has pitjat Acceptar!")
      }
      else
      {
        document.write("Has pitjat Cancel·lar!")
      }
    }
  </script>
</head>
```

```
<body>
<input type="button" onclick="mostra_confirmacio()" value="Mostra Confirmació">

</body>
</html>
```

Demandar informació

S'utilitzen quan es vol demandar informació a un usuari abans d'accedir a una pàgina. Se li donen 2 opcions, **Acceptar** o **Cancel·lar**. Si l'usuari **Accepta** la finestra retorna el **valor introduït** per ell, en cas contrari, retorna **null**. `prompt("text", "valor per defecte")`

Exemple:

```
<html>
<head>
  <script type="text/javascript">
    function mostra_prompt()
    {
      var nom=prompt("Siusplau indica el teu nom", "Harry Potter")
      if (nom!=null && nom!="")
      {
        document.write("Hola " + nom + "! Com estàs avui?")
      }
    }
  </script>
</head>
<body>
  <input type="button" onclick="mostra_prompt()" value="Mostra Prompt" >
</body>
</html>
```

EVENTS

Utilitzant JavaScript es poden fer pàgines web dinàmiques. Un event és una acció que pot ser detectada per JavaScript.

Tot element d'un web té certes accions que poden accionar funcions de JavaScript.

*Pe. Es pot utilitzar l'acció **onClick** d'un element botó per indicar que una funció s'executarà quan un usuari premi el botó.*

Les accions es declaren a les etiquetes dels elements HTML.

Alguns exemples d'accions:

- Un click de ratolí
- La càrrega d'una pàgina web o una imatge
- Moure el ratolí sobre un punt concret de la pàgina
- Seleccionar un camp d'entrada de dades en un formulari
- Enviar les dades d'un formulari
- Pitjar qualsevol tecla

FF: Firefox, **N:** Netscape, **IE:** Internet Explorer

Atribut	L'event succeeix quan...	FF	N	IE
onabort	La càrrega d'una imatge és interrompuda	1	3	4
onblur	Un element perd el focus	1	2	3
onchange	L'usuari canvia el contingut d'un camp	1	2	3
onclick	Es fa un clic de mouse sobre un objecte	1	2	3
ondblclick	Es fa un doble-clic de mouse sobre un objecte	1	4	4
onerror	Succeeix un error mentre s'està carregant un document o una imatge	1	3	4
onfocus	Un element té el focus	1	2	3
onkeydown	Es pitja una tecla	1	4	3
onkeypress	Es pitja o es manté pitjada una tecla	1	4	3
onkeyup	Es deixa anar una tecla	1	4	3
onload	Una pàgina o imatge s'ha acabat de carregar	1	2	3
onmousedown	Es pitja un botó del ratolí	1	4	4
onmousemove	Es mou el ratolí	1	6	3
onmouseout	El ratolí marxa d'un element	1	4	4
onmouseover	El ratolí entra en un element	1	2	3
onmouseup	Es deixa anar un botó del ratolí	1	4	4
onreset	Es pitja el botó de reset	1	3	4
onresize	Es redimensiona una finestra o un frame	1	4	4
onselect	Es selecciona un text	1	2	3
onsubmit	Es pitja el botó de submit	1	2	3
onunload	L'usuari surt de la pàgina	1	2	3

FUNCIONS ESPECIALS

Existeixen una sèrie de funcions que es poden utilitzar en tots els objectes de JavaScript.

Funció	Descripció	FF	N	IE
eval()	Avalua una cadena i l'executa com si fos el codi d'un script	1	2	3
isFinite()	Valida si un valor és un nombre finit	1	4	4
isNaN()	Valida si un valor no és un nombre	1	2	3
Number()	Converteix el valor d'un objecte en un nombre	1		
parseFloat()	Converteix un string en un nombre en coma flotant	1	2	3
parseInt()	Converteix un string en un enter	1	2	3
String()	Converteix el valor d'un objecte en un string	1		

Exemple:

```
<script type="text/javascript">
  eval("x=10;y=20;document.write(x*y)")
  document.write("<br />")
  document.write(eval("2+2"))
  var x=10
  document.write(eval(x+17))
  document.write("<br />")

  document.write(isNaN(123)+ "<br>")
  document.write(isNaN(-1.23)+ "<br>")
  document.write(isNaN(5-2)+ "<br>")
  document.write(isNaN(0)+ "<br>")
  document.write(isNaN("Hello")+ "<br>")

  document.write(eval("3 + 3 +" + parseInt("4")))
  document.write(parseInt("10") + "<br>")
  document.write(parseFloat("10") + "<br>")
  document.write(parseInt("10.00") + "<br>")
  document.write(parseFloat("10.00") + "<br>")
  document.write(parseInt("10.33") + "<br>")
  document.write(parseFloat("10.33") + "<br>")
  document.write(parseInt("34 45 66") + "<br>")
  document.write(parseFloat("34 45 66") + "<br>")
  document.write(parseInt(" 60 ") + "<br>")
  document.write(parseFloat(" 60 ") + "<br>")
  document.write(parseInt("40 years") + "<br>")
  document.write(parseFloat("40 years") + "<br>")
  document.write(parseInt("He was 40") + "<br>")
  document.write(parseFloat("He was 40") + "<br>")
</script>
```