JAVASCRIPT

rortegag

Básico

En la página
```
<script type="text/javascript"> ... </script>
```
Incluir archivo JS externo
```
<script src="filename.js"></script>
```
Retraso - 1 segundo de espera
```
setTimeout(function () {

}, 1000);
```
Funciones
```
function addNumbers(a, b) {
  return a + b; ;
}
x = addNumbers(1, 2);
```
Editar elemento DOM
```
document.getElementById("elementID").innerHTML = "Hello World!";
```
Salida
```
console.log(a);
```
escribir en la consola del navegador.
```
document.write(a);
```
escribir en el HTML.
```
alert(a);
```
salida en una caja de alerta.
```
confirm("Really?");
```
diálogo sí/no, devuelve verdadero/falso dependiendo del clic del usuario.
```
prompt("Your age?","0");
```
diálogo de entrada. El segundo argumento es el valor inicial.

Comentarios
```
/* Línea múltiple
comentario */
// Una línea
```

Tipos de datos

```
var age = 18;
```
numero
```
var name = "Jane";
```
cadena de texto
```
var name = {first:"Jane", last:"Doe"};
```
objeto
```
var truth = false;
```
booleano
```
var sheets = ["HTML","CSS","JS"];
```
matriz
```
var a; typeof a;
```
indefinido
```
var a = null;
```
valor nulo

Objetos
```
var student = {        nombre del objeto
  firstName:"Jane",
  lastName:"Doe",
  age:18,
  height:170,    lista de propiedades y valores
  fullName : function() {     objeto de función
    return this.firstName + "" + this.lastName;
  }
};
student.age = 19;      valor de ajuste
student[age]++;      incremento
name = student.fullName();
```
llamar a la función del objeto

Variables

```
var a;
```
variable
```
var b = "init";
```
cadena de texto
```
var c = "Hi" + " " + "Joe";
```
= "Hi Joe"
```
var d = 1 + 2 + "3";
```
= "33"
```
var e = [2,3,5,8];
```
matriz
```
var f = false;
```
booleano
```
var g = /()/;
```
RegEx (expresión regular)
```
var h = function(){};
```
objeto de función
```
const PI = 3.14;
```
constante
```
var a = 1, b = 2, c = a + b;
```
una línea
```
let z = 'zzz';
```
variable local del ámbito de bloque

Modo estricto
```
"use strict";
```
Utiliza el modo estricto
para escribir código seguro
```
x = 1;
```
Lanza un error porque
la variable no está declarada

Valores
false, true
booleano
18, 3.14, 0b10011, 0xF6, NaN
numero
"flower", 'John'
cadena de texto
undefined, null , Infinity
especial

Operadores a nivel de bit
& AND
5 & 1 (0101 & 0001) 1 (1)
| OR
5 | 1 (0101 | 0001) 5 (101)
~ NOT
~ 5 (~0101) 10 (1010)
^ XOR
5 ^ 1 (0101 ^ 0001) 4 (100)
<< left shift
5 << 1 (0101 << 1) 10 (1010)
>> right shift
5 >> 1 (0101 >> 1) 2 (10)
>>> zero fill right shift
5 >>> 1 (0101 >>> 1) 2 (10)

MÁS

Para más información: https://developer.mozilla.org/es/docs/Web/JavaScript

JAVASCRIPT

rortegag

Variables

Operadores

```
a = b + c - d;
suma, resta
a = b * (c / d);
multiplicación, división
x = 100 % 48;
módulo   100 / 48 resto = 4
a++; b--;
incremento y decremento postfix
```

Aritmética

```
a * (b + c)
agrupación
person.age
miembro
person[age]
miembro
!(a == b)
no es lógico
a != b
no es igual
typeof a
tipo (número, objeto,
función...)
x << 2  x >> 3
cambio aritmético
a = b
asignación
a == b
es igual a
a != b
desigual a
a === b
estricta igualdad
a !== b
estricta desigualdad
a < b   a > b
menor y mayor que
a <= b  a >= b
menor o igual, mayor o igual
a += b
a = a + b (funciona con -, *, %...)
a && b
lógico y
a || b
lógico o
```

Cadenas de texto

```
var abc = "abcdefghijklmnopqrstuvwxyz";
var esc = 'I don\'t \n know';
\n nueva línea
var len = abc.length;
longitud de la cadena de texto
abc.indexOf("lmno");
encontrar subcadena de texto,
-1 si no contiene
abc.lastIndexOf("lmno");
última ocurrencia
abc.slice(3, 6);
recorta "def", los valores negativos
cuentan por detrás
abc.replace("abc","123");
buscar y reemplazar, toma
expresiones regulares
abc.toUpperCase();
convertir a mayúsculas
abc.toLowerCase();
convertir a minúsculas
abc.concat(" ", str2);
abc + " " + str2
abc.charAt(2);
carácter en el índice: "c"
abc[2];
inseguro, abc[2] = "C" no funciona
abc.charCodeAt(2);
código de caracteres en el
índice: "c" -> 99
abc.split(",");
al dividir una cadena en comas
se obtiene un array
abc.split("");
división de caracteres
128.toString(16);
número a hex(16), octal (8) o binario (2)
```

Números y matemáticas

```
var pi = 3.141;
pi.toFixed(0);
devuelve 3
pi.toFixed(2);
devuelve 3.14
pi.toPrecision(2)
devuelve 3.1
pi.valueOf();
devuelve el número
Number(true);
convertir a número
Number(new Date())
número de milisegundos
desde 1970
parseInt("3 months");
devuelve el primer número: 3
parseFloat("3.5 days");
devuelve 3,5
```

Constantes

```
E, PI, SQRT2, SQRT1_2, LN2, LN10, LOG2E, Log10E
Number.MAX_VALUE
el mayor número posible de JS
Number.MIN_VALUE
el menor número posible de JS
Number.NEGATIVE_INFINITY
-Infinity
Number.POSITIVE_INFINITY
Infinity
```

Matemáticas

```
var pi = Math.PI;
3.141592653589793
Math.round(4.4);
= 4 - redondeado
Math.round(4.5);
= 5
Math.pow(2,8);
= 256 - 2 a la potencia de 8
Math.sqrt(49);
= 7 - raíz cuadrada
Math.abs(-3.14);
= 3,14 - absoluto,
valor positivo
Math.ceil(3.14);
= 4 - redondeado hacia arriba
Math.floor(Math.random() * 5) + 1;
número entero aleatorio, de 1 a 5
Math.floor(3.99);
= 3 - redondeado hacia abajo
Math.sin(0);
= 0 - seno
Math.cos(Math.PI);
OTROS: tan,atan,asin,acos,
Math.min(0, 3, -2, 2);
= -2, el valor más bajo
Math.max(0, 3, -2, 2);
= 3, el valor más alto
Math.log(1);
= 0 logaritmo natural
Math.exp(1);
= 2,7182pow(E,x)
Math.random();
número aleatorio entre
0 y 1
```

Para más información: https://developer.mozilla.org/es/docs/Web/JavaScript

JAVASCRIPT

rortegag

Matrizes

```
var dogs = ["Bulldog", "Beagle", "Labrador"];
var dogs = new Array("Bulldog", "Beagle", "Labrador");
declaración
alert(dogs[1]);
acceder al valor en el índice, siendo el primer elemento [0]
dogs[0] = "Bull Terier";
cambiar el primer elemento
for (var i = 0; i < dogs.length; i++) {
  console.log(dogs[i]);
}
recorrer matriz
```

Métodos

concat, copyWithin, every, fill, filter, find, findIndex, forEach, indexOf, isArray, join, lastIndexOf, map, pop, push, reduce, reduceRight, reverse, shift, slice, some, sort, splice, toString, unshift, valueOf

Ejemplos de métodos

```
dogs.toString();
convertir en cadena: resultados "Bulldog,Beagle,Labrador"
dogs.join(" * ");
unirse: "Bulldog * Beagle * Labrador"
dogs.pop();
eliminar el último elemento
dogs.push("Chihuahua");
añadir un nuevo elemento al final
dogs[dogs.length] = "Chihuahua";
lo mismo que push()
dogs.shift();
eliminar el primer elemento
dogs.unshift("Chihuahua");
añadir un nuevo elemento al principio
delete dogs[0];
cambiar el elemento a indefinido (no recomendado)
var animals = dogs.concat(cats,birds);
unir dos matrices (perros seguidos de gatos y pájaros)
dogs.splice(2, 0, "Pug", "Boxer");
añadir elementos (dónde, cuántos eliminar,
lista de elementos)
dogs.slice(1,4);
elementos de [1] a [4-1]
dogs.sort();
ordenar la cadena alfabéticamente
dogs.reverse();
ordenar la cadena en orden descendente
x.sort(function(a, b){return a - b});
clasificación numérica
x.sort(function(a, b){return b - a});
clasificación numérica descendente
highest = x[0];
el primer elemento de la matriz ordenada es el valor
más bajo (o más alto)
x.sort(function(a, b){return 0.5 - Math.random()});
clasificación por orden aleatorio
```

Fechas

```
var d = new Date();
Obtiene la fecha actual. Ejemplo de su salida:
Thu Sep 08 2022 13:02:08 GMT+0200 (hora de verano de
Europa central)
Number(d)
1662634928398 milisegundos transcurridos desde 1970
Date("2017-06-23");
declaración de la fecha
Date("2017");
se fija en el 01 de enero de 2017
Date("2017-06-23T12:00:00-09:45");
fecha - hora AAAA-MM-DDTHH:MM:SSZ
Date("June 23 2017");
formato de fecha largo
Date("Jun 23 2017 07:45:00 GMT+0100 (Tokyo Time)");
zona horaria
```

Obtener una fecha

```
var d = new Date();
a = d.getDay();
obtener el día de la semana
getDate();
el día como un número (1-31)
getDay();
el día de la semana como un
número (0-6)
getFullYear();
año de cuatro dígitos (aaaa)
getHours();
hora (0-23)
getMilliseconds();
milisegundos (0-999)
getMinutes();
minutos (0-59)
getMonth();
mes (0-11)
getSeconds();
segundos (0-59)
getTime();
milisegundos desde 1970
```

Establecer una fecha

```
var d = new Date();
d.setDate(d.getDate() + 7);
añade una semana a una
fecha
setDate();
el día como un número
(1-31)
setFullYear();
año (opcionalmente mes
y día)
setHours();
hora (0-23)
setMilliseconds();
milisegundos (0-999)
setMinutes();
minutos (0-59)
setMonth();
mes (0-11)
setSeconds();
segundos (0-59)
setTime();
milisegundos desde 1970)
```

Para más información: https://developer.mozilla.org/es/docs/Web/JavaScript

JAVASCRIPT

rortegag

Declaración If - Else

```
if ((age >= 14) && (age < 19)) {
condición lógica
  status = "Eligible.";
  se ejecuta si la condición es verdadera
} else {
El bloque else es opcional
  status = "Not eligible.";
  se ejecuta si la condición es falsa
}
```

Declaración Switch

```
switch (new Date().getDay()) {
la entrada es el día actual
case 6:
  si (día == 6)
  text = "Sábado";
  break;
case 0:
  si (día == 0)
  text = "Domingo";
  break;
default:
  si no...
  text = "Lo que sea";
}
```

Expresiones regulares

```
var a = str.search(/CheatSheet/i);
```

Modificadores

```
i     realizar una coincidencia sin
distinción de mayúsculas y minúsculas
g     realizar una coincidencia global
m     realizar una coincidencia multilínea
```

Patrones

```
\     carácter de escape
\d    encontrar un dígito
\s    encontrar un carácter de espacio
en blanco
\b    encontrar coincidencias al principio
o al final de una palabra
n+    contiene al menos un n
n*    contiene cero o más ocurrencias
de n
n?    contiene cero o una ocurrencia de n
^     inicio de la cadena de texto
$     fin de la cadena de texto
\uxxxx      encontrar un carácter Unicode
.     cualquier carácter simple
(a|b)     a o b
(...)     sección de grupos
[abc]     en el rango (a, b o c)
[0-9]     cualquiera de los dígitos entre
los corchetes
[^abc]    no está en el rango
\s    espacio en blanco
```

Bucles

Bucle For

```
for (var i = 0; i < 10; i++) {
  document.write(i + ": " + i*3 + "<br />");
}
var sum = 0;
for (var i = 0; i < a.length; i++) {
  sum + = a[i];
}
analizar una matriz
html = "";
for (var i of custOrder) {
  html += "<li>" + i + "</li>";
}
```

Bucle While

```
var i = 1;
inicializar
while (i < 100) {
entra en el ciclo si la declaración es
verdadera
  i *= 2;
  incrementa para evitar el bucle infinito
  document.write(i + ", ");
  salida
}
```

Bucle Do While

```
var i = 1;
inicializar
do {
entra en el ciclo al menos
una vez
  i *= 2;
  incrementa para evitar el
  bucle infinito
  document.write(i + ", ");
  salida
} while (i < 100)
repite el ciclo si la declaración
es verdadera al final
```

```
a?    cero o uno de a
a*    cero o más de a
a*?   cero o más, no codicioso
a+    uno o más de a
a+?   uno o más, no codicioso
a{2}     exactamente 2 de a
a{2,}    2 o más de a
a{,5}    hasta 5 de a
a{2,5}     2 a 5 de a
a{2,5}?    2 a 5 de a, no codicioso
[:punct:]    cualquier símbolo de
puntuación
[:space:]    cualquier carácter de espacio
[:blank:]    espacio o tabulación
```

Break

```
for (var i = 0; i < 10; i++) {
  if (i == 5) { break; }
  se detiene y sale del ciclo
  document.write(i + ", ");
  el último número de salida es
  el 4
}
```

Continue

```
for (var i = 0; i < 10; i++) {
  if (i == 5) { continue; }
  se salta el resto del ciclo
  document.write(i + ", ");
  se salta el 5
}
```

Func. globales

```
eval();
ejecuta una cadena de texto
como si fuera un código
de script
Cadena(23);
devuelve una cadena de texto
a partir de un número
(23).toString();
devolver cadena de texto a
partir de número
Number("23");
devuelve el número a partir de
la cadena
decodeURI(enc);
decodificar URI.
Resultado: "pagina.asp"
encodeURI(uri);
codificar URI.
Resultado: "pagina.asp":
"pagina.asp"
decodeURIComponent(enc);
decodificar un componente URI
encodeURIComponent(uri);
codificar un componente URI
isFinite();
es la variable un número finito
y legal
isNaN();
es la variable un número ilegal
parseFloat();
devuelve el número en coma
flotante de la cadena
parseInt();
analiza una cadena y devuelve
un número entero
```

Para más información: https://developer.mozilla.org/es/docs/Web/JavaScript

JAVASCRIPT

rortegag

Eventos

<button onclick="myFunction();">Click here</button>

Ratón
onclick, oncontextmenu, ondblclick, onmousedown, onmouseenter, onmouseleave, onmousemove, onmouseover, onmouseout, onmouseup

Teclado
onkeydown, onkeypress, onkeyup

Portapapeles
oncopy, oncut, onpaste

Medios de comunicación
onabort, oncanplay, oncanplaythrough, ondurationchange, onended, onerror, onloadeddata, onloadedmetadata, onloadstart, onpause, onplay, onplaying, onprogress, onratechange, onseeked, onseeking, onstalled, onsuspend, ontimeupdate, onvolumechange, onwaiting

Marco
onabort, onbeforeunload, onerror, onhashchange, onload, onpageshow, onpagehide, onresize, onscroll, onunload

Formulario
onblur, onchange, onfocus, onfocusin, onfocusout, oninput, oninvalid, onreset, onsearch, onselect, onsubmit

Arrastrar
ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop

Animación
animationend, animationiteration, animationstart

Miscelánea
transitionend, onmessage, onmousewheel, ononline, onoffline, onpopstate, onshow, onstorage, ontoggle, onwheel, ontouchcancel, ontouchend, ontouchmove, ontouchstart

Promesas

```
function sum (a, b) {
  return Promise(function (resolve, reject) {
    setTimeout(function () {
    enviar la respuesta después de 1 segundo
      if (typeof a !== "number" || typeof b !== "number") {
      comprobación de los tipos de entrada
        return reject(new TypeError("Inputs must be numbers"));
      }
      resolve(a + b);
    }, 1000);
  });
}
var myPromise = sum(10, 5);
myPromsise.then(function (result) {
  document.write(" 10 + 5: ", result);
  return sum(null, "foo");
  Datos no válidos y devuelven otra promesa
}).then(function () {
No se llamará por el error
}).catch(function (err) {
En su lugar, se llama al manejador de captura, después de otro segundo
  console.error(err);
  => Por favor, proporcione dos números para sumar.
});
```

Estados
pending, fulfilled, rejected

Propiedades
Promise.length, Promise.prototype

Métodos
Promise.all(iterable), Promise.race(iterable), Promise.reject(reason), Promise.resolve(value)

Para más información: https://developer.mozilla.org/es/docs/Web/JavaScript

JAVASCRIPT

rortegag

Errores

```
try {      bloque de código a probar
  undefinedFunction();
}
catch(err) {    bloque para manejar los errores
  console.log(err.message);
}
```

Valores del nombre del error

```
RangeError
Un número está "fuera de rango"
ReferenceError
Se ha producido una referencia ilegal
SyntaxError
Se ha producido un error de sintaxis
TypeError
Se ha producido un error de tipo
URIError
Se ha producido un error de encodeURI()
```

Lanzar error

```
throw "My error message";
lanzar un texto
```

Validación de entrada

```
var x = document.getElementById("mynum").value;
obtener el valor de entrada
try {
  if(x == "")  throw "empty";
  if(isNaN(x)) throw "not a number";
  x = Number(x);
  if(x > 10)   throw "too high";
  casos de error
}
catch(err) {    si se produce un error
  document.write("Input is " + err);     escribir el error a la salida
  console.error(err);     escribir el error en la consola
}
finally {
  document.write("</br />Done");
  se ejecuta independientemente del resultado de try / catch
}
```

JSON

```
var str = '{"names":[' +
crear objeto JSON
'{"first":"Hakuna","lastN":"Matata" },' +
'{"first":"Jane","lastN":"Doe" },' +
'{"first":"Air","last":"Jordan" }]}';
obj = JSON.parse(str);
analiza
document.write(obj.names[1].first);
acceso
```

Enviar

```
var myObj = { "name":"Jane", "age":18, "city":"Chicago" };
crear objeto
var myJSON = JSON.stringify(myObj);
encadenar texto
window.location = "demo.php?x=" + myJSON;
enviar a un archivo php
```

Almacenamiento y recuperación

```
myObj = { "name":"Jane", "age":18, "city":"Chicago" };
myJSON = JSON.stringify(myObj);
almacenamiento de datos
localStorage.setItem("testJSON", myJSON);
text = localStorage.getItem("testJSON");
recuperación de datos
obj = JSON.parse(text);
document.write(obj.name);
```

Para más información: https://developer.mozilla.org/es/docs/Web/JavaScript