

Introducción a JavaScript

Mauro Bender

Mayo 28, 2012

JavaScript

- Es un lenguaje de programación que se ejecuta del lado del cliente, es decir, se ejecuta en el navegador del usuario.
- Nos permite acceder al DOM (Document Object Model) del documento de forma “fácil”.
- Nos proporciona herramientas para poder manipular el DOM.
- Vamos a ver los aspectos básicos del lenguaje y herramientas que nos van a ser útiles al momento de hacer nuestras animaciones.

DOM

El DOM (Document Object Model) es un interfaz a través de la cual nos es posible acceder y modificar el contenido, estructura y estilo de los documentos HTML (es lo que nos permite acceder a todas esas cositas bonitas que queremos cambiar y animar).

Mas información:

http://es.wikipedia.org/wiki/Document_Object_Model.

Variables

En javascript las variables no son más que “contenedores” que nos permiten almacenar distintos tipos de valores.

Para declarar una variable usamos la palabra reservada **var** como sigue:

```
var nombre_variable;
```

Los nombres de variables pueden contener caracteres alfanuméricos y los símbolos \$ y _, con la salvedad de que no pueden comenzar con un número.

Las variables se pueden inicializar al declararlas o en cualquier momento posterior en el programa, para esto usamos el operador de =:

```
var numero = 5;
var nombre;
nombre = "Margarito Flores";
```

Tipos de variables

En javascript las variables pueden ser de diferentes tipos:

- **Numeric**: Variables numéricas como: 1, 4, 4.
- **String**: Cadenas de caracteres (texto) como: “Hola mundo!”, “Dieguín!!”.
- **Boolean**: Valores de verdad: **true** o **false**.
- **Array**: Es una lista de elementos indexada por un número positivo:


```
var colores = ["rojo", "azul", "verde"];
```
- **Object**: Objetos, pares de clave valor como por ejemplo:


```
var objeto = {"nombre": "mauro", "edad" : 24};
```

No es necesario especificar el tipo de valor que va a contener la variable, javascript lo determina al momento en que se le asigna un valor.

Tipo String y Array

Las variables de tipo String y Array proporcionan un conjunto de propiedades y métodos que son útiles cuando se está trabajando con ellas. De ellas una de las más importantes es "length" que nos devuelve el tamaño de la cadena o del arreglo.

```
var cadena = "Esto es una cadena ;-);";
var arreglo = ["esto", "es", "un", "arreglo", ":P"];

cadena.length; // 22
arreglo.length; // 5
```

Más información

Para saber que otras propiedades y/o métodos poseen los Arrays y los Strings pueden ir a

http://www.w3schools.com/jsref/jsref_obj_string.asp (para información sobre el tipo String) o a

http://www.w3schools.com/jsref/jsref_obj_array.asp (para información sobre el tipo Array).

Comentarios

Los comentarios son porciones de textos que podemos agregar a nuestro código para hacerlo más entendible. Éstos no afectan la ejecución del código ya que son ignorados por el navegador. Para escribir un comentario podemos usar // o /* ... */:

```
// Comentario de una línea

/* Comentario
de
varias
líneas
*/
```

Operadores - Operadores aritméticos

Existen diferentes tipos de operadores en JavaScript

- **+**: Efectúa la suma de dos números.

```
var suma = 5 + 4; // suma ahora vale 9
```

- **-**: Se utiliza para restar números.

```
var resta = 5 - 4; // resta ahora vale 1
```

- *****: Sirve para multiplicar dos números

```
var resultado = 2 * 3; // resultado ahora vale 6
```

- **/**: Se utiliza para dividir dos números.

```
var resultado = 6 / 3; // resultado ahora vale 2
```

- **%**: Se utiliza para obtener el módulo de una división (resto de la división entera).

```
var resultado = 7 % 3; // resultado ahora vale 1
```

Operadores - Operadores aritméticos unarios

- **++**: Incrementa en uno la variable afectada.

```
var numero = 1;
numero++; // numero ahora vale 2
```

Es lo mismo que ejecutar:

```
var numero = 1;
numero = numero + 1; // numero ahora vale 2
```

- **--**: Análogo a ++. Decrementa en uno la variable afectada.

```
var numero = 1;
numero--; // numero ahora vale 0
```

Es lo mismo que ejecutar:

```
var numero = 1;
numero = numero - 1; // numero ahora vale 0
```

Operadores - Operadores de comparación

- **==**: Devuelve **true** si los dos operandos son iguales. **false** en cualquier otro caso.

```
var num1 = 5, num2 = 3, num3 = 5;
num1 == num2; // false
num1 == num3; // true
```

- **!=**: Devuelve **true** si los dos operandos son distintos. **false** en cualquier otro caso.

```
var nombre1 = "javier", nombre2 = "adri", nombre3 = "javier";
nombre1 != nombre2; // false
nombre1 != nombre3; // true
```

- **!**: Niega la expresión afectada. Devuelve **true** si expresión es falsa y **false** si es verdadera.

```
var falso = !true; // falso es igual a false

var distintos = !("iguales" == "iguales"); // Es lo mismo que hacer "iguales" != "iguales"
```

- **===**: Compara por igualdad pero con un criterio más fuerte que **==**. Devuelve **true** si los dos operandos contienen el mismo valor y además son del mismo tipo. **false** en cualquier otro caso.

```
var num1 = 5, // Es del tipo Numeric
    num2 = "5", // Es del tipo String
    num3 = 5; // Es del tipo Numeric
num1 === num2; // false
num1 === num3; // true
```

- **!==**: Análogo a **===**. Devuelve **true** si los operandos son de distinto tipo o si si contienen distintos valores. **false** en cualquier otro caso.

```
var num1 = 5, // Es del tipo Numeric
    num2 = "5", // Es del tipo String
    num3 = 3; // Es del tipo Numeric
num1 !== num2; // true
num1 !== num3; // true
```

- **<, <=, >, >=**: Devuelven **true** si el primer operando es menor, menor o igual, mayor o mayor o igual que el segundo operando respectivamente. **false** en otro caso.

```
var num1 = 5, num2 = 3, num3 = 5;
num1 < num2; // false
num1 <= num3; // true;
num1 > num3; // false
num1 >= num2; // true
```

Para tener en cuenta

El operador **+** también sirve para concatenar cadenas y arrays.

```
var nombre = "Mauro" + " " + "Bender";
// nombre ahora es "Mauro Bender"

var colores = ["azul", "rojo"] + ["blanco", "negro"];
// colores ahora es ["azul", "rojo", "blanco", "negro"]
```

Funciones

Las funciones son bloques de códigos que reciben parámetros y pueden devolver un valor. Para declarar una función usamos la palabra reservada **function** y encerramos el código de la función entre **{ }**.

```
function nombre_de_la_funcion (param1, param2, ...) {
    // codigo de la funcion
}
```

La palabra reservada **return** se utiliza para indicar que se debe terminar la ejecución de la función y devolver el valor que se indica a continuación. Si no se indica, la función termina al encontrar la **}** que cierra el bloque y no devuelve ningún valor.

```
function suma(numero1, numero2) {
    return numero1 + numero2;
}
```

Para llamar a una función sólo debemos escribir su nombre pasándole entre paréntesis los valores que queremos que reciba.

```
var res = suma(3, 6); // res es ahora 9
```

En javascript las funciones son interpretadas como valores, por lo que las podemos asignar a una variable.

```
var suma = function(numero1, numero2) {return numero1 + numero2;}
var res = suma(3, 6);
```

Por lo tanto, también es posible pasarlas como un parámetro a otra función.

```
function alertarResultado (funcion, param1, param2) {
var res = funcion(param1, param2);
alert(res);
}
function suma(num1, num2) {return num1 + num2;}
alertarResultado(suma, 3, 6);
```

Estructuras de control

En javascript disponemos también de ciertas estructuras que nos permiten controlar el flujo de ejecución de un programa.

If... Else...

Con la estructura **if... else...** podemos decirle al programa que ejecute una porción de código o no en base a una condición.

```
if(condicion) {
// hace algo si es condicion es verdadera
} else {
// hacer otra cosa en otra caso
}
```

El bloque correspondiente al **else** que sólo se ejecuta cuando la condición es false es opcional y se puede omitir si uno así lo desea.

En el siguiente ejemplo mostramos una alerta indicando si un número ingresado por el usuario es igual o no a 3, usando un if.

```
var numero = prompt("Ingrese un numero:");

if(numero == 3) {
alert("Es igual a 3 =.");
} else {
alert("No es igual a 3 =.");
}
```

prompt y alert

prompt y **alert** son dos “popups” que nos proporciona javascript. **prompt** sirve para pedirle al usuario que ingrese un valor mientras que **alert** se usa comúnmente para mostrarle información al usuario. Mas información: http://www.w3schools.com/js/js_popup.asp.

While

while sirve para repetir una porción de código mientras una condición siga siendo válida. Su sintaxis es:

```
while(condicion) {
// hace algo mientras condicion sea verdadera
}
```

Por ejemplo podemos tener un programa que vaya decrementando un número ingresado por el usuario en 1 hasta llegar a 0, alertando en cada paso el número que va quedando.

```
var numero = prompt("Ingrese un numero mayor a 0: ");

while(numero > 0) {
alert("Numero ahora todavia no es 0: " + numero);

numero--;
}

alert("Llegamos a cero");
```

For

for es también una estructura de repetición como **while**, pero es usualmente usado para ejecutar una porción de códigos una cantidad fija de veces.

```
for(inicializacion; condicion; incremento) {  
  // codigo  
}
```

Como se puede ver la estructura **for** se compone de cuatro partes:

- **inicializacion:** Aquí debemos, como su nombre lo indica, inicializar las variables que vamos a usar para controlar el loop.
- **condicion:** Acá debemos escribir una condición que se tiene que cumplir para que el bucle se siga ejecutando.
- **incremento:** Una porción de código que se va a ejecutar cada vez que se haya completado un ciclo del bucle, usualmente usado para incrementar o decrementar la variable que se usa para controlar el bucle.
- **codigo:** El código que se va a ejecutar en cada repetición.

Ejemplo:

```
var colores = ["rojo", "verde", "azul", "blanco"];  
  
for(var i = 0; i < colores.length; i++) {  
  alert("El color actual es: " + colores[i]);  
}
```

En el ejemplo creamos un array con diferentes colores y después lo recorremos y mostramos los colores que contiene usando un bucle **for**.

Repaso

¿Qué vimos?

- Variables
 - Como declarar variables.
 - Diferentes tipos de variables disponibles en Javascript.
- Operadores
 - Operadores aritméticos.
 - Operadores de comparación.
 - Operadores cuya significado depende del tipo de variables a la que se aplica ("+").
- Funciones
 - Diferentes formas de declarar funciones.
 - Funciones que reciben funciones como parámetros.
- Estructuras de control de flujo
 - If... Else...
 - While()
 - For(;;)

¿Qué vamos a ver el jueves?

- La librería JQuery: ¿Qué es? ¿Cómo se usa?
- Uso de los selectores para obtener los elementos de la página.
- Cómo modificar el css de un elemento desde javascript.
- Animaciones incorporadas en JQuery.
- Crear nuestras animaciones con el método "animate".

Fin =).