Un vistazo a Java script

El mundo de la programación dirigida a objetos

Autor: Geuberth Johan Robles Santiago   
*Risaralda, Universidad Tecnológica de Pereira, Pereira, Colombia*

Correo-e: ejemplo@org.es

***Resumen*— Javascript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los**[**navegadores**](http://www.maestrosdelweb.com/editorial/%C2%BFcomo-elegir-un-navegador-web/)**son los encargados de interpretar estos códigos.**

**Muchos confunden el Javascript con el Java pero ambos lenguajes son diferentes y tienes sus características singulares. Javascript tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.**

**Java por su parte tiene como principal característica ser un lenguaje independiente de la plataforma. Se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado:**[**Linux**](http://www.maestrosdelweb.com/editorial/linux/)**,**[**Windows**](http://www.maestrosdelweb.com/editorial/microsoft/)**,**[**Apple**](http://www.maestrosdelweb.com/editorial/apple/)**, etc. Debido a sus características también es muy utilizado para internet.**

**Como síntesis se pude decir que Javascript es un lenguaje interpretado, basado en prototipos, mientras que Java es un lenguaje más orientado a objetos.(cit. Damián Pérez Valdés)**

***Palabras clave*—** [**lenguaje de programación**](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n)[**interpretado**](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica))**;** **Es el**[**lenguaje**](https://www.ecured.cu/Lenguaje)**cuyo código no necesita ser preprocesado mediante un**[**compilador**](https://www.ecured.cu/Compilador)**, eso significa que el ordenador es capaz de ejecutar la sucesión de instrucciones dadas por el programador sin necesidad de leer y traducir exhaustivamente todo el código.**

**Programación orientada a objetos: es un**[**paradigma de programación**](https://es.wikipedia.org/wiki/Paradigma_de_programaci%C3%B3n)**que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial**.

1. INTRODUCCIÓN

Hasta ahora hemos visto como crear una página web con [HTML](https://programarfacil.com/podcast/27-desarrollo-web-con-html5/) y como darle estilo con [CSS](https://programarfacil.com/podcast/29-desarrollo-web-con-css3/), con **JavaScript** nuestra página web va a tomar “movimiento”, vamos a poder interactuar con el navegador (abrir ventanas, ir adelante o atrás en el histórico…) e interactuar con los elementos de nuestra página web (verificar formularios, añadir/eliminar contenido, …).

JavaScript es el lenguaje de *script*de más amplio uso en la web y con las últimas tendencias de aplicaciones cliente se está convirtiendo en el lenguaje de programación más importante para el entorno web. En los últimos años han salido librerías como jQuery y framewoks como AngularJS o Node.js con lo que está adquiriendo un gran potencial.

Aunque su nombre lo sugiere, nada tiene que ver con Java. Su nombre es una táctica comercial de Netscape cuando allá por el año 1995 lo introdujo en su navegador. Por lo que ni es una versión reducida de Java ni es un lenguaje simple.

JavaScript es un lenguaje **orientado a objetos**, aunque con distinta filosofía que C++ y Java, no existen las clases y los objetos son colecciones de métodos y propiedades (más adelante veremos lo que es un objeto de este lenguaje).

Su sintaxis es similar a C++ o Java, aunque es menos restrictiva, no es obligatorio poner **;** al final de la sentencia (aunque recomendable), no es obligatorio declarar variables …

Su léxico diferencia entre mayúsculas y minúsculas, los comentarios son como C/C++, el ; es opcional si cada sentencia está en una línea (aunque nosotros recomendamos ponerlo siempre), las declaración de variables no es obligatoria, aunque aconsejable y estas no tienen tipo. Por lo que JavaScript es un lenguaje **no tipado**, es decir, si declaramos una variable esta puede almacenar tanto números como caracteres. Una gran diferencia con otros lenguajes de programación.

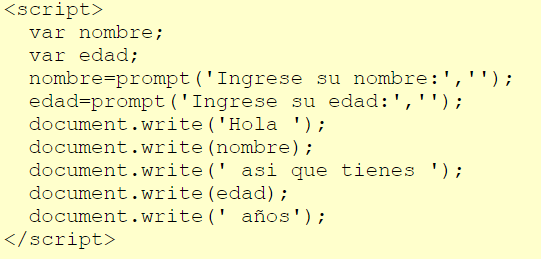
1. CONTENIDO

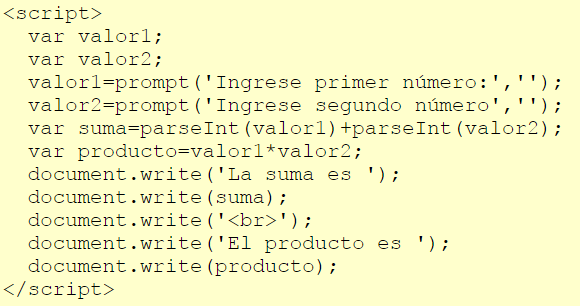
**Variables**: Una variable es un depósito donde hay un valor. Consta de un nombre y pertenece a un tipo (numérico, cadena de caracteres, etc.)

**Tipos de variable**:

* Valores Enteros (100, 260, etc.)
* Valores Reales (1.24, 2.90, 5.01, etc.)
* Cadenas de caracteres ('Juan', 'Compras', 'Listado', etc.)
* Valores lógicos (true, false)

**Entrada de Datos**: Para la entrada de datos por teclado tenemos la función prompt. Cada vez que necesitamos ingresar un dato con ésta función aparece una ventana donde cargamos el valor. Hay otras formas más sofisticadas para la entrada de datos en una página HTML, pero para el aprendizaje de los conceptos básicos de JavaScript nos resultará más práctica esta función.



**Estructura Secuencial**: Cuando en un problema sólo participan operaciones, entradas y salidas se la denomina estructura secuencial.  
Ejemplo:  


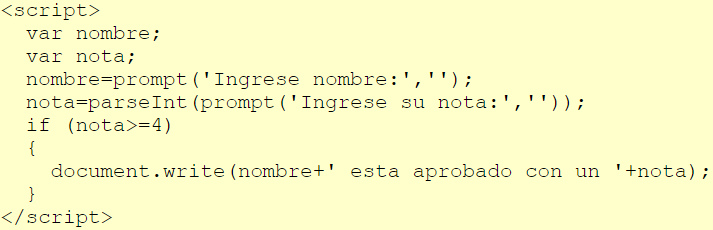
**Estructura Condicional simple:** No todos los problemas pueden resolverse empleando estructuras secuenciales. Cuando hay que tomar una decisión aparecen las estructuras condicionales.

En nuestra vida diaria se nos presentan situaciones donde debemos decidir.

¿Elijo la carrera A o la carrera B ?  
¿Me pongo este pantalón ?  
¿Entro al sitio A o al sitio B ?  
Para ir al trabajo, ¿elijo el camino A o el camino B ?  
Al cursar una carrera, ¿elijo el turno mañana, tarde o noche ?

Por supuesto que en un problema se combinan estructuras secuenciales y condicionales.  
Cuando se presenta la elección tenemos la opción de realizar una actividad o no realizarla.

En una estructura CONDICIONAL SIMPLE por el camino del verdadero hay actividades y por el camino del falso no hay actividades. Por el camino del verdadero pueden existir varias operaciones, entradas y salidas, inclusive ya veremos que puede haber otras estructuras condicionales.

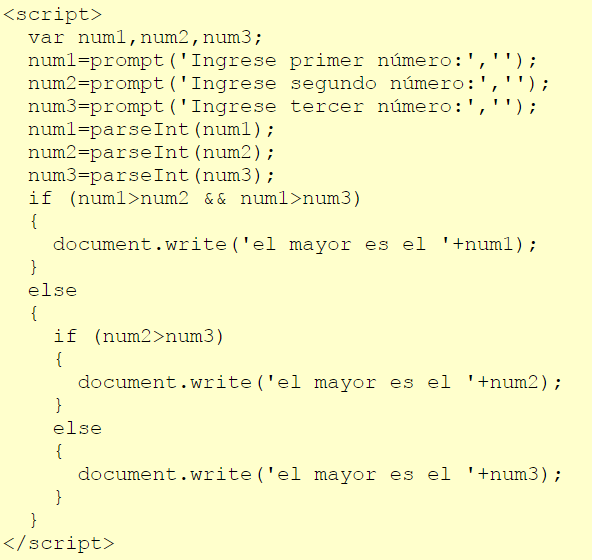


**Operadores lógicos && (y) en las estructuras condicionales**: El operador &&, traducido se lo lee como "Y". Se emplea cuando en una estructura condicional se dispone dos condiciones.

Cuando vinculamos dos o más condiciones con el operador "&&" las dos condiciones deben ser verdaderas para que el resultado de la condición compuesta de Verdadero y continúe por la rama del verdadero de la estructura condicional.

Recordemos que la condición debe ir entre paréntesis en forma obligatoria.

La utilización de operadores lógicos permiten en muchos casos, plantear algoritmos más cortos y comprensibles.



**Funciones**: En programación es muy frecuente que un determinado procedimiento de cálculo definido por un grupo de sentencias tenga que repetirse varias veces, ya sea en un mismo programa o en otros programas, lo cual implica que se tenga que escribir tantos grupos de aquellas sentencias como veces aparezca dicho proceso.

La herramienta más potente con que se cuenta para facilitar, reducir y dividir el trabajo en programación, es escribir aquellos grupos de sentencias una sola y única vez bajo la forma de una FUNCION.

Un programa es una cosa compleja de realizar y por lo tanto es importante que esté bien ESTRUCTURADO y también que sea inteligible para las personas. Si un grupo de sentencias realiza una tarea bien definida, entonces puede estar justificado el aislar estas sentencias formando una función, aunque resulte que sólo se le llame o use una vez.

Hasta ahora hemos visto cómo resolver un problema planteando un único algoritmo.

Con funciones podemos segmentar un programa en varias partes.  
Frente a un problema, planteamos un algoritmo, éste puede constar de pequeños algoritmos.

Una función es un conjunto de instrucciones que resuelven una parte del problema y que puede ser utilizado (llamado) desde diferentes partes de un programa.

Consta de un nombre y parámetros.  
Con el nombre llamamos a la función, es decir, hacemos referencia a la misma.  
Los parámetros son valores que se envían y son indispensables para la resolución del mismo.  
La función realizará alguna operación con los parámetros que le enviamos. Podemos cargar una variable, consultarla, modificarla, imprimirla, etc.

Incluso los programas más sencillos tienen la necesidad de fragmentarse. Las funciones son los únicos tipos de subprogramas que acepta JavaScript. Tienen la siguiente estructura:

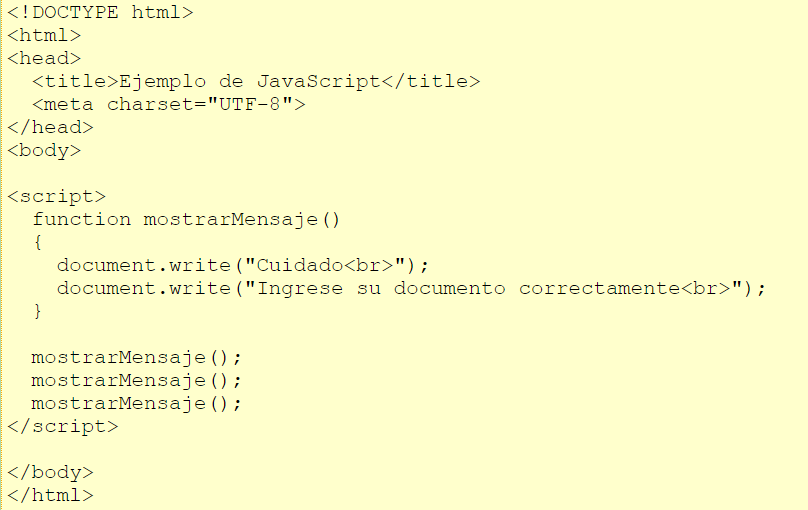
**function <nombre de función>(argumento1, argumento2, ..., argumento n)**

**{**

**<código de la función>**

**}**

**Ejemplo:**



**Programación dirigida a Objetos**: Un objeto es una estructura que contiene tanto las variables (llamadas propiedades) como las funciones que manipulan dichas variables (llamadas métodos). A partir de esta estructura se ha creado un nuevo modelo de programación (la programación orientada a objetos) que atribuye a los mismos propiedades como herencia o polimorfismo. Como veremos, JavaScript simplifica en algo este modelo y hace una programación híbrida entre la programación estructurada y la programación orientada a objetos.

El modelo de la programación orientada a objetos normal y corriente separa los mismos en dos: clases e instancias (objetos). Las primeras son entes más abstractos que definen un conjunto determinado de objetos. Las segundas son miembros de una clase, poseyendo las mismas propiedades que la clase a la cual pertenecen.

Propiedades y métodos.

Para acceder a los métodos y propiedades de un objeto debemos utilizar la siguiente sintaxis:

objeto.propiedad  
objeto.metodo(parametros)

### Conceptos Básicos.

Objetos: Son todas las cosas con identidad propia. Se relacionan entre si. Poseen características (atributos) y tienen responsabilidades (funciones, métodos) que deben cumplir.  
Son ejemplares (instancias) de una clase y conocen a la clase a la cual pertenecen.

Atributos o propiedades: Son las características, cualidades distintivas de cada objeto. Deben ser mínimos para poder realizar todas las operaciones que requiere la aplicación.

**Ejemplos de objetos del mundo real**:

- Casa:

Atributos: tamaño, precio, cantidad de habitaciones, etc.;

Responsabilidades: comodidad, seguridad, etc.

- Mesa:

Atributos: altura, largo, ancho, etc.;

Responsabilidades: contener elementos.

- Ventana:

Atributos: tamaño, color, etc.;

Responsabilidades: abrirse, cerrarse, etc.

Ejemplos de objetos del mundo de la programación:

- Ventana:

Atributos: tamaño, color, etc.;

Responsabilidades: mostrar título, achicarse

etc.

1. CONCLUSIONES

JavaScript es una ayuda indispensable para realizar páginas en las que queremos una alta interacción con el usuario, este nos proporciona una serie de eventos que reconoce, unas librerías de funciones y métodos y todo lo queramos hacer con un lenguaje: bucles de condición, Variables…. Todo con la ventaja de que se ejecuta en el cliente y no tiene que estar llamando continuamente al servidor por lo que descarga el trabajo de este y hace que la navegación sea de mayor velocidad y eficacia en si misma

BIBLIOGRAFIA

**(Cit. Damián Pérez Valdés):** <http://www.maestrosdelweb.com/que-es-javascript/>

**Contenido:**

<https://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=10>

**Conclusiones**:

Geuberth Johan Robles Santiago.