Universidad CETEC Cuernavaca

VII Cuatrismestre
Ingeniería en Sistemas Computacionales

# PROGRAMACIÓN WEB

Profr. Miguel Ángel Piza Larios

Juan Hugo Reyes Pérez

## Resumen Unidad 5
## Programación del lado del cliente y del lado del servidor

# ÍNDICE



# ÍNDICE DE IMÁGENES

# 1. Programación del lado del cliente y del lado del servidor

## 1.1 Introducción al lenguaje del lado del cliente

Los lenguajes de programación del lado cliente se usan para su integración en páginas web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado, no compilado.
Para incorporar un fragmento de código script en una página HTML se introduce el script entre los tags <script> y </script>. Existen dos lenguajes script en la actualidad: el VBScript (derivado de Visual Basic) y el JavaScript (derivado de Java).

## 1.2 Elementos de programación del lado del cliente

Vbscript (Visual Basic Script)
Es un lenguaje de script, directamente derivado de Visual Basic. Para insertar código VBScript en una página HTML se añade al tag <script> el parámetro language=
"VBScript", que determina cuál de los lenguajes de script se utiliza. Decimos que los lenguajes de script se ejecutan interpretados, no compilados. Esto significa que un código escrito en un lenguaje de script no sufre ninguna transformación previa a su ejecución.

*Figura 1 Para incorporar un fragmento de código script en una página HTML se introduce el script entre los tags <script> y </script>*

Cada línea de código es traducida a lenguaje máquina justo antes de su ejecución. Después es ejecutada y la traducción no se conserva en ningún sistema de almacenamiento. Si es necesaria otra ejecución, el intérprete se verá abocado a realizar una nueva traducción de cada línea de código.
Sin embargo, el lenguaje Visual Basic, del cual deriva el VBScript, es un leguaje compilado. Esto significa que un código en Visual Basic sufre un proceso global de traducción a lenguaje máquina. Todo el código es traducido de una sola vez y el resultado de esa traducción se almacena en el disco con la extensión .exe. Cuando llega el momento de la ejecución, se ejecuta el código compilado, no el código original del programa, llamado código nativo o código fuente. Cada sistema tiene sus ventajas e inconvenientes (Editores, 2012).

**Las ventajas e inconvenientes de la interpretación son:**
El código es cómodo para depurar, ya que no es necesario volver a compilar tras un cambio.
No es necesario disponer de un compilador, ya que el intérprete (que forma parte del navegador) ejecuta el script.
El mantenimiento es fácil y rápido, por parte del autor o de otro programador.
La ejecución se ralentiza, al ser necesaria la interpretación línea a línea cada vez.
El código es visible y puede ser objeto de plagio por parte de otras personas.
El usuario tiene acceso al código y puede modificarlo, estropeando alguna operación.

**Las ventajas e inconvenientes de la compilación son:**

El código compilado se ejecuta muy rápido, al no ser necesaria una traducción cada vez.
El código compilado no puede ser "abierto" por otras personas. No es necesario transmitir el código fuente.
El código compilado puede estar, íntegramente, incluido en un solo fichero.
Es necesario disponer de un compilador-linkador para el proceso de la compilación.
El código compilado suele ocupar bastante en disco, ya que incorpora en el propio código algunas librerías del sistema.
Depurar un programa implica volver a compilar tras los cambios.

El código en VBScript puede, además, estar diseñado para su ejecución en el lado del cliente o en el del servidor. La diferencia es que un código que se ejecuta en el lado del servidor no es visible en el lado del cliente. Este recibe los resultados, pero no el código. El código que se debe de ejecutar en el lado del servidor estará incluido en la página web correspondiente entre los tags <% y %>. Además, habrá que renombrar la página para aplicarle la extensión .asp (Active Server Page. página activa en servidor) (Deitel et al., 2014). Un ejemplo de una página web sencilla que incluye código VBScript:


*Figura 2 El código en VBScript puede estar diseñado para su ejecución en el lado del cliente o en el del servidor.*

```
<html>
<head>
  <title>cuadro de mensaje</title>
</head>
<body>
  <script language = "vbscript"> msgbox ("ejemplo de mensaje")
  </script>
</body>
</html>
```

**JavaScript**

JavaScript es un lenguaje de scripts compacto basado en objetos (y no orientado a objetos). Originariamente era denominado LiveScript, y fue desarrollado por Netscape para su navegador Netscape Navigator 2.0. Fue éste el primer cliente en incorporarlo. Se ejecuta sobre 16 plataformas diferentes, incluyendo los entornos de Microsoft.
Las diferentes versiones de JavaScript han sido finalmente integradas en un estándar denominado ECMAScript-262. Dicho estándar ha sido realizado por la organización ECMA dedicada a la estandarización de información y sistemas de comunicación. Las versiones actuales de los navegadores soportan este estándar.
JavaScript permite la realización de aplicaciones de propósito general a través de la Web y aunque no está diseñado para el desarrollo de grandes aplicaciones es suficiente para la implementación de aplicaciones web completas o interfaces web hacia otras más complejas. Por ejemplo, una aplicación escrita en JavaScript puede ser incrustada en un documento HTML proporcionando un mecanismo para la detección y tratamiento de eventos, como clics

del ratón o validación de entradas realizadas en formularios. Sin existir comunicación a través de la red una página HTML con JavaScript incrustado puede interpretar, y alertar al usuario con una ventana de diálogo, de que las entradas de los formularios no son válidas. O bien realizar algún tipo de acción como ejecutar un fichero de sonido, un applet de Java, etc.

```
<html>
        <head>
<script language=”JavaScript”> document.write(“Esto es JavaScript!”)
</script>
</head>
<body>
<br>Este es un documento HTML normal<br> </body>
</html>
```

Este primer programa se limita a escribir en pantalla un determinado texto para lo que se emplea el código document.write. En este código, document es un objeto creado por el sistema que hace referencia al propio documento y write es uno de los métodos que proporciona para interactuar con él. El resultado de cargar este documento en un visualizador que interprete JavaScript será la aparición de los dos textos, el escrito en JavaScript y el escrito en HTML, sin que el usuario sea consciente del proceso. Podemos hablar también de páginas dinámicas del servidor, que son reconocidas, interpretadas y ejecutadas por el propio servidor.

## 1.3 Manipulación de objetos del lado del cliente

JavaScript es un lenguaje interpretado que posee una característica que lo hace especialmente idóneo para trabajar en Web, ya que son los navegadores que utilizamos para viajar por el a los que interpretan y por tanto ejecutan los programas escritos en JavaScript. De esta forma, podemos enviar documentos a través de la Web que incorporan el código fuente de un programa, convirtiéndose de esta forma en documentos dinámicos, y dejando de ser simples fuentes de información estáticas.

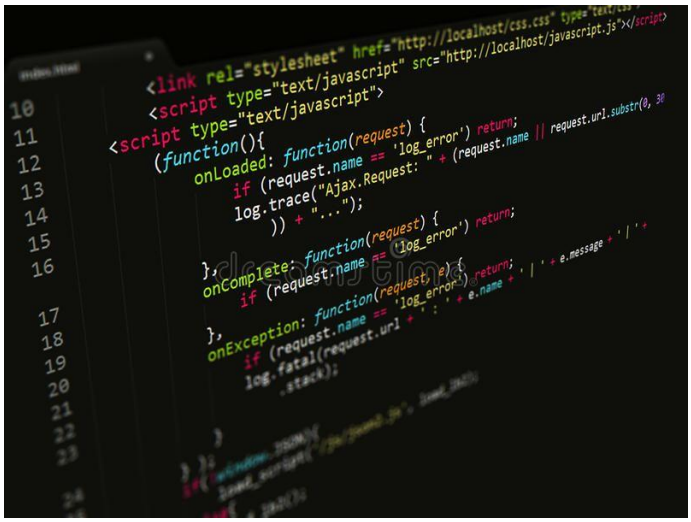


*Figura 3 JavaScript es especialmente idóneo para trabajar en Web*

Las dos principales características de JavaScript son, por un lado, que es un lenguaje basado en objetos y por otro que es un lenguaje orientado a eventos, debido al tipo de entornos en los que se utiliza. Esto implica que gran parte de la programación en JavaScript se centra en describir objetos y escribir funciones que respondan a movimientos del ratón, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos.

JavaScript no permite un control absoluto sobre los recursos de la computadora. Cada programa en JavaScript solo tiene acceso al documento HTML en el que va inmerso y, si

acaso, a las ventanas en las que se ejecuta el navegador dentro del cual se está ejecutando el programa en JavaScript.

## Elementos básicos de JavaScript

```
function getCountries() {
    return fetch('https://restcountries
}

function mostrarBanderas(countries) {
    contBanderas.innerHTML = '';
    countries.map((country, i) => {
        let bandera = document.createE
        bandera.src = country.flag;
        bandera.width = '20';
```

*Figura 4 En JavaScript no es necesario declarar una variable, pero cuando se hace es por medio de la palabra reservada var.*

***Variables***. En JavaScript no es necesario declarar una variable, pero cuando se hace es por medio de la palabra reservada var. Una variable, cuando no es declarada, tiene siempre ámbito global, mientras que en caso contrario será de ámbito global si está definida fuera de una función y local si está definida dentro.

***Asignación***: este tipo de operador se utiliza para asignar valores a las variables.

Comparación: en JavaScript, se pueden comparar variables de distinto tipo, pues es capaz de forzar conversiones:

== Devuelve true si son iguales. Forzar conversiones de tipo.
!= Devuelve true si son distintos. Fuerza conversiones de tipo.
=== Devuelve true si son iguales y del mismo tipo.
!== Devuelve true si son distintos o de distinto tipo.
> Devuelve true si la variable de la izquierda es mayor que la variable de la derecha.
< Devuelve true si la variable de la derecha es mayor que la variable de la izquierda.
>= Devuelve true si la variable de la izquierda es mayor o igual que la variable de la derecha.
<= Devuelve true si la variable de la izquierda es menor o igual que la variable de la derecha.

***Aritméticos***: los operadores aritméticos, a partir de varios operandos, devuelven un solo valor; resultado de la operación realizada con los anteriores operandos. En JavaScript existe notación postfija y prefija, por lo que variable++ y ++variable son dos formas distintas de incrementar una variable. En primer lugar, se procesa la variable, y luego se incrementa. Sin embargo, en el segundo caso, primero se incrementa la variable y después se procesa. Estos son los operadores aritméticos más importantes:

| | |
|---|---|
| * | Multiplicación. |
| / | División. |
| + | Suma. |
| – | Resta. |
| % | Resto de una división. |
| ++ | Incrementa el valor de la variable. |
| -- | Disminuye el valor de una variable. |

Lógicos. Estos operadores permiten realizar expresiones aritméticas. Es importante saber que si en la primer evaluación ya se conoce el resultado, no se evalúa la segunda expresión, es decir para el operador && si la primer expresión es falsa entonces automáticamente la evaluación es falsa y para el operador || si la primer expresión es verdadera entonces automáticamente la evaluación es verdadera.

&& Y
|| O
! Negación

Estructuras de Control. En todo lenguaje de programación existen estructuras que nos permiten variar el orden de ejecución dependiendo de ciertas condiciones. Estas estructuras se pueden clasificar en dos grandes grupos: bifurcaciones condicionales y bucles.
Un bucle es una estructura que permite repetir una tarea un número de veces, determinado por una condición. Para hacer bucles podemos utilizar las estructuras while y do…while. Su sintaxis es la siguiente:

```
while (condicion) {
acciones;
}

do {
acciones;
} while (condicion);
```

Estos bucles iteran indefinidamente mientras se cumpla una condición. La diferencia entre ellas es que la primera comprueba dicha condición antes de realizar cada iteración y la segunda lo hace después, es decir, el while comprueba primero si la condición se cumple y si no se cumple no se ejecuta el código que tiene dentro del bucle. En cambio, el do…while primero ejecuta el código y luego comprueba si la condición se cumple, en caso de que no se cumpla ya no continúa con el bucle. Con la sentencia break, se puede salir de una sentencia de bucle sin limitaciones. Con la sentencia continue, se termina el bucle actual y se comienza con el siguiente.

El código contenido en el bucle se ejecutará mientras la condición se cumpla. Antes de comenzar la primera iteración del bucle se ejecutará la sentencia inicio y en cada iteración se ejecutará la sentencia incremento.

## 1.4 Introducción al lenguaje del lado del servidor

Los lenguajes de programación del lado del servidor son especialmente útiles en trabajos que se tiene que acceder a información centralizada, situada en una base de datos en el servidor, y cuando por razones de seguridad los cálculos no se pueden realizar en la computadora del usuario.
Es importante destacar que los lenguajes de programación del lado del servidor son necesarios porque para hacer la mayoría de las aplicaciones web se debe tener acceso a muchos recursos externos a la computadora del cliente, principalmente bases de datos

alojadas en servidores de Internet. Un caso claro es un banco: no tiene ningún sentido que el cliente tenga acceso a toda la base de datos, sólo a la información que le concierne.
Las páginas dinámicas del servidor se suelen escribir en el mismo archivo HTML, mezclado con el código HTML, al igual que ocurría en las páginas del cliente. Cuando una página es solicitada por parte de un cliente, el servidor ejecuta los scripts y se genera una página resultado, que solamente contiene código HTML. Este resultado final es el que se envía al cliente y puede ser interpretado sin lugar a errores ni incompatibilidades, puesto que sólo contiene HTML.

## 1.5 Elementos de programación del lado del servidor

Para escribir páginas dinámicas de servidor existen varios lenguajes.

**Practical Extraction and Report Language (PERL)**
Es un lenguaje de programación desarrollado por Larry Wal inspirado en otras herramientas de UNIX. PERL es el lenguaje más utilizado para la creación de programas CGI en los servidores web. Es más rápido que los programas shell script de UNIX, puede leer y escribir ficheros binarios, PERL no necesita ser recompilado, es un lenguaje interpretado.

**Active Server Pages (ASP)**
El estándar ASP permite utilizar cualquier lenguaje para la programación. En teoría y mediante la programación, es posible adaptar versiones de lenguajes como Pascal, C y otros, para utilizarlos dentro de páginas ASP.

**Java Server Pages (JSP)**
Es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje Java. La tecnología JSP, o de JavaServer Pages, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página web.

**Hipertext Preprocesor (PHP)**
PHP (acrónimo recursivo de “PHP: Hypertext Preprocessor”, originado inicialmente del nombre PHP Tools, o Personal Home Page Tools) es un lenguaje de programación interpretado. Aunque fue concebido en el tercer trimestre de 1994 por Rasmus Lerdorf no fue hasta el día 8 de junio de 1995 que fue lanzada la versión 1.0. Se utiliza entre otras cosas para la programación de páginas web activas, y se destaca por su capacidad de mezclarse con el código HTML.

## 1.6 Aplicación del lado del servidor

Una de las aplicaciones más importantes de PHP es su integración con diversos motores de base de datos. El PHP está construido para generar en forma sencilla páginas web dinámicas a partir de información almacenada en bases de datos (Pavón Puertas, 2007). Conexión a la base
$db_link=mysql_connect(hostname, user, password);

Ejemplo:

$db=mysql_connect(“localhost”,”root”,”secreto”);

La función realiza la conexión a la base Mysql y devuelve “false” si hubo algún error en la conexión o un link a la conexión a la base en caso de que la conexión sea exitosa. El link es un número que indica la sesión dentro del MySql. Para finalizar la conexión se debe utilizar la función mysql_close(). Es muy importante cerrar la conexión a la base de datos una vez finalizadas las transacciones para evitar la sobrecarga en el motor de la base de datos (Pavón Puertas, 2007).

Selección de la base de datos a utilizar mysql_select_db(database_name, db_link);

Ejemplo: mysql_select_db(“test”,$db);

Esta función configura cual es la base de datos que se utilizara por omisión. En este caso el link a utilizar en esta función es el link que se obtuvo al ejecutar la función mysql_connect. La función mysql_select_db devuelve el valor “false” en caso de que se encuentre algún error, como por ejemplo la inexistencia de la base de datos. En este punto cabe aclarar que la denominación de las bases de datos de MySql es casesensitive, por lo que debemos mantener un standard a la hora de elegir los nombres de las distintas bases de datos.

Queries a la base de datos
Nuevamente el link que se debe usar es el que se obtiene al conectarse a la base, mysql_query devuelve falso en caso de que el query no pueda ejecutarse (error de SQL) o bien un result set en los casos que devuelva algún tipo de datos como por ejemplo en un select.

Es muy importante fijarse con quÉ usuario se realizó la conexión a la base de datos a la hora de ejecutar el mysql_connect, ya que la gran mayoría de los errores producidos en esta instancia son el resultado de la falta de permisos para realizar la consulta.