

# Lenguaje *JavaScript*

# Objetivos

- **Conocer las características de los lenguajes de scripts en general y específicamente de JavaScript**
- **Conocer los Tipos de Páginas Web y cuando deben utilizarse.**
- **Conocer la sintaxis básica del lenguaje JavaScript**

# Objetivos

- **Conocer el modelo de objetos de JavaScript**
- **Conocer el modelo de eventos de JavaScript**
- **Conocer técnicas de programación para generar páginas dinámicas en cliente e interactuar con el usuario.**

# Características de Lenguajes de Scripts

- Son interpretados no compilados
- Los errores de sintaxis se detectan en tiempo de ejecución
- No requieren grandes cantidades de espacio al ser archivos de texto
- Pueden ser transportados a cualquier plataforma que los soporte
- Los cambios son más fáciles de hacer
- Se crean/modifican en cualquier editor de texto

# Lenguajes de Scripts en Clientes Web

- La forma más habitual para dotar de dinamismo(interacción con el usuario y páginas dinámicas) en cliente a una pagina web, es el uso de *Scripts* (guiones)

# Características de Lenguajes de Scripts en Cliente

- Las instrucciones se ejecutan en el momento de la carga de la página web
- Se pueden incluir tantos scripts como se considere necesario ya sea en el <HEAD> como en cualquier parte del cuerpo <BODY>
- Las funciones o scripts se pueden ejecutar como respuesta a un evento generado en el cliente

# **Características de Lenguajes de Scripts en Cliente**

- **Liberan al servidor de una gran carga de computo**
- **Disminuyen el trafico innecesario por la red**
- **Ofrece respuesta inmediata al usuario**
- **Permiten la creación dinámica de paginas en el cliente y la validación de datos en formularios**
- **Permiten utilizar recursos locales como CPU, RAM,etc.**

# Desventajas de Lenguajes de Scripts en Cliente

- No funcionan en cualquier browser
- Son tecnologías propietarias
- Interpretados no compilados
- Restricciones de acceso a los recursos cliente
- Existen diferentes versiones
- Son difíciles de depurar



# Lenguaje JavaScript

- Es un lenguaje de programación *interpretado y basado en objetos* que permite crear documentos web dinámicos y tiene la capacidad de capturar y responder a eventos o sucesos generados por el cliente



# Características: JavaScript

- Su sintaxis es similar a la de Java
- Débilmente tipado
- Es case sensitivo
- No existen las constantes
- Basado no orientado a objetos
- “Universalidad” de uso
- Seguridad de ejecución
- Páginas más ligeras de cargar

# Sintaxis Básica: JavaScript

- *Un lenguaje de programación esta formado por palabras básicas denominadas tokens que tienen un significado especial.*
- **JavaScript** clasifica los tokens en 5 grupos: identificadores, palabras reservadas, literales, operadores y separadores.

# Sintaxis Básica: Identificadores

- **Son nombres que representan una variable, un método o un objeto.**

Un identificador puede estar compuesto por:

- **Letras (A..Z, a..z)**
- **Subguiones o guión bajo ( \_ )**
- **Números (0..9)**

# Sintaxis Básica: Identificadores

## Restricciones que deben cumplir:

- Los identificadores no pueden coincidir con las palabras reservadas del lenguaje.
- Los identificadores no pueden incluir separadores
- Deben iniciar con una letra o un subguión

# Sintaxis Básica: Identificadores

## Ejemplos

### Correctos:

Variable1, V, v, Variable\_n

### Incorrectos:

1Variable, variable@, buenos dias,  
false, true, otra;variable

# Sintaxis Básica: Palabras Reservadas

- **Todo lenguaje se reserva determinados identificadores para usos específicos a los cuales se les denomina palabras reservadas y no pueden ser usadas como identificadores**

# Sintaxis Básica: Palabras Reservadas

abstract

boolean

break

byte

case

catch

char

class

const

continue

default

delete

do

double

else

extends

false

final

finally

float

for

function

goto

if

implements

import

in

InstanceOf

int

interface

labeled

long

native

new

null

package



# Sintaxis Básica: Palabras Reservadas

private

protected

public

return

short

static

super

switch

synchronized

this

throw

throws

transient

true

by

typeof

var

void

while

with

# Sintaxis Básica: Literales

- Un literal se utiliza para representar valores fijos y está compuesto por combinaciones de números o de caracteres.
- Un literal representa un valor que no varía durante la ejecución de un *Script*.

# Sintaxis Básica: Literales

En JavaScript existen 5 tipos de literales:

- **Enteros**
- **Reales**
- **Booleanos**
- **Strings(cadenas de caracteres)**
- **Caracteres especiales**

# Sintaxis Básica: Literales

## Entero:

- **representa un valor numérico entero expresado en base decimal, octal o hexadecimal.**
- **Literal decimal:** combinación de números del 0 al 9 (el primer dígito no debe ser 0)
- **Ejemplo:** 10, 245, 3459, 898

# Sintaxis Básica: Literales

## Entero:

- **Literal octal:** combinación de números del 0 al 7 (el primer dígito debe ser 0)
- **Ejemplo:** 07, 020, 0233

# Sintaxis Básica: Literales

## Entero:

- **Literal hexadecimal:** combinación de números del 0 al 9 y letras a..f ó A..F (debe iniciar con los caracteres 0x)
- Ejemplo: 0xF, 0Xf, 0X10, 0xf000

# Sintaxis Básica: Literales

## Real:

- Representa un valor numérico real y puede expresarse tanto en notación científica como estándar.

## Ejemplo Notación Normal:

var x=123.56;

y = 0.234;

var z = 400.25;

# Sintaxis Básica: Literales

## Real:

- **Notación científica:** se puede utilizar la letra e ó E para indicar el exponente y tanto la base como el exponente pueden ir precedidos por el signo + ó -

## Ejemplos:

var numero1=+234E-3;

var numero2=25E-2, numero3=-1e-6;



# Sintaxis Básica: Literales

## Booleano:

- Son las palabras reservadas `true` y `false` (en minúsculas) estos literales tienen asignado el valor 1 y 0 respectivamente.
- Las expresiones lógicas devuelven uno de estos valores: *true* o *false*

## Ejemplos:

```
var cierto=true, falso=false;
```

```
return(true);
```

# Sintaxis Básica: Literales

## String:

- Son combinaciones de caracteres (incluyendo la cadena vacía). Un literal de tipo string debe incluirse entre comillas dobles o simples

## Ejemplos

```
var x1=""Ejemplo1",x2='ejemplo2';
```

# Sintaxis Básica: Literales

## Caracteres especiales:

- Son caracteres que tienen un significado especial para JavaScript. Si deseamos que estos no se interpreten debemos incluir un carácter de escape.

# Sintaxis Básica: Literales

## Caracteres especiales:

- \b** borra el último carácter
- \f** genera alimentación de línea
- \n** genera el carácter línea nueva
- \r** genera el carácter retorno de carro
- \t** genera un tabulador
- \\** carácter \
- \'** carácter '
- \''** carácter ''

# Sintaxis Básica: Operadores

- Un operador es un símbolo que expresa la operación a realizar con los operandos.

## Operadores:

- Aritméticos: \*, /, +, -, %
- Lógicos: &&, ||, !
- Relacionales: <, >, <=, >=, ==, !=

# Sintaxis Básica: Separadores

- Los separadores son caracteres que permiten separar tokens o combinaciones de estos, como por ejemplo expresiones.
- Separadores: *espacio en blanco, la tabulación y el retorno de carro.*

*\*Estos no pueden formar parte de un identificador*

# Sintaxis Básica: Separadores

**Ejemplos:**

```
var x, a, b, c;
```

```
a = 1;
```

```
b = 5;
```

```
c = 6;
```

```
x = a + b * c;
```

# **Sintaxis Básica: VARIABLES**

- **Una variable es una posición de memoria a la que se le asigna un nombre o identificador y a la que se puede acceder para obtener o modificar el valor almacenado**



# Sintaxis Básica: VARIABLES

## Ambito de una variable

- **Global:** si no esta dentro de ninguna de las funciones.
  - **Local:** si se ha definido dentro de una función ó es un parámetro.
- \* Cuando se utilizan variables no declaradas JavaScript las declara como globales.

# Sintaxis Básica: VARIABLES

## Tipos de Variables JavaScript:

<u>Tipo</u>	<u>Ejemplo</u>	
number	3.1416	200
boolean	true	false
string	"abcd"	'cadena'
function	typeof	isNaN
object	document	window

# Sintaxis Básica: VARIABLES

## Tipos de Variables JavaScript:

- Las variables de tipo number pueden contener *enteros* ó *reales*
- Las variables de tipo boolean pueden contener el valor *true* ó *false*
- Las variables de tipo string contienen cadenas de caracteres

# Sintaxis Básica: VARIABLES

## Tipos de Variables JavaScript:

- Las variables de tipo function pueden hacer *referencia* a funciones propias del lenguaje JavaScript o a funciones definidas por el usuario.
- Las variables de tipo object contienen objetos(propiedades y métodos)

# Sintaxis Básica: CONSTANTES

- Una constante es una variable cuyo valor no sufre modificaciones durante la ejecución del programa
- JavaScript no soporta el concepto de constante
- Es responsabilidad del programador no modificar los valores constantes

# Sintaxis Básica: EXPRESIONES

- Una expresión es una sentencia o conjunto de sentencias que devuelven un único valor. Una expresión puede devolver uno de los cinco tipos de datos que reconoce JavaScript.

# Sintaxis Básica: EXPRESIONES

- Ejemplos

```
x = 110;
```

```
fin = false;
```

```
(x > 15) && (fin==false);
```

```
document.lastModified;
```

# Sintaxis Básica: COMENTARIOS

- *Cuando se desarrolla un programa es recomendable documentarlo por medio de comentarios para facilitar su comprensión.*
- *JavaScript permite insertar comentarios de una sola línea con el carácter // y de varias líneas iniciando con /\* y finalizando con \*/*



# Sintaxis Básica: COMENTARIOS

## Ejemplo:

// Comentario de una sola línea

/\* Comentario que ocupa más  
de una línea\*/

# Sintaxis Básica: FUNCIONES

- *Una función es una secuencia de instrucciones caracterizada por un nombre y que puede ejecutarse tantas veces como sea necesario simplemente llamando a la función por el nombre que la identifica*

# Sintaxis Básica: FUNCIONES

## *Objetivos de las funciones JavaScript*

- Estructurar el código de la aplicación
- Evitar repeticiones de segmentos de código.
- Permitir que un *script* se ejecute ante determinados eventos.

# Sintaxis Básica: FUNCIONES

## Ejemplo:

```
<SCRIPT Language="JavaScript">
var x=10;
function cuadrado(valor) {
    valor = valor * valor;
    return(valor);
}
cuadrado(x);
document.write(x);
</SCRIPT>
```

# OPERADORES: JavaScript

- *En JavaScript existe una variedad de operadores que actúan sobre los distintos tipos de datos. Estos operadores se clasifican en unarios y binarios*

*Operador de asignación =, ejemplos:*

*$a = 3 * 4;$*

*$b = b + 1;$*

*$c = a + b;$*

# OPERADORES: JavaScript

## *Operadores Aritméticos*

	<u>Significado</u>	<u>Ejemplo</u>
+	Suma	$a + b$
-	Resta	$a - b$
-	Niega	$-a$
*	Producto	$a * b$
/	Cociente	$a / b$
%	Módulo	$a \% b$

# OPERADORES: JavaScript

## *Operadores Aritméticos abreviados*

$+=$              $b += 3$              $b = b + 3$

$-=$              $b -= 3$              $b = b - 3$

$*=$              $b *= 3$              $b = b * 3$

$/=$              $b /= 3$              $b = b / 3$

$\%=$              $b \% = 3$              $b = b \% 3$

$++$              $++b, b++$              $b = b + 1$

$--$              $--b, b--$              $b = b - 1$

# OPERADORES: JavaScript

## *Operadores de Comparación*

==	igual	a == b
!=	distinto	a != b
>	mayor que	a > b
<	menor que	a < b
>=	mayor o igual que	a >= b
<=	menor o igual que	a <= b
===	estrictamente igual	a===b
!==	estrictamente distinto	a!==b



# OPERADORES: JavaScript

## *Operadores Lógicos*

&&	AND ('y' lógica)	(a>0) && (b>0)
	OR ('o' lógica)	(a>0)    (b>1)
!	NOT ('no' lógica)	!(a>0)

# OPERADORES: JavaScript

## *Operadores para strings*

<u>sign</u>	<u>significado</u>	<u>ejemplo</u>
+	concatenar	s1 + s2
==	¿son iguales?	s1 == s2
!=	¿son distintos?	s1 != s2
>	¿s1 mayor que s2?	s1 > s2
>=	¿s1 mayor o igual que s2?	s1 >= s2
<	¿s1 menor que s2?	s1 < s2
<=	¿s1 mayor que s2?	s1 <= s2

# OPERADORES: JavaScript

## *Operador typeof*

- Se utiliza para conocer el tipo de dato con el que se está trabajando. Los distintos tipos de dato que puede devolver el operador typeof son las cadenas: ‘number’, ‘string’, ‘boolean’, ‘function’, ‘object’ y ‘undefined’

Sintaxis:

```
tipo = typeof variable;
```

Ejemplo:

```
(typeof dato === 'number')? ++dato: "no es número";
```

# OPERADORES: JavaScript

## *Operadores a nivel de bits*

&	AND bit a bit	$2 \& 3 \Rightarrow 2$
	OR bit a bit	$2   3 \Rightarrow 3$
^	XOR bit a bit	$2 \wedge 3 \Rightarrow 1$
~	NOT bit a bit	$\sim 2 \Rightarrow 1$
>>	rotación a derecha	$2 \ll 2 \Rightarrow 8$
<<	rotación a izquierda	$134 \gg 2 \Rightarrow 33$

# Estructuras de Control

- **JavaScript cuenta con las siguientes estructuras de control**

Condicionales: if...else...

Bucles:

for, for...in, while, do...while, break y continue

Sentencia etiquetada

Sentencia with

Sentencia switch

# Estructuras de Control

## Sentencias condicionales

### Sintaxis:

```
if (condición) {  
    //instrucciones  
} [else {  
    //instrucciones  
}]
```

# Estructuras de Control

**Bucles: for**

**Sintaxis:**

```
for ([valor_inicial];[condición];[incremento])  
{  
    //instrucciones  
}
```

# Estructuras de Control

**Bucles: for...in**

**Sintaxis:**

```
for ( variable in [objeto|matriz])  
{  
    //instrucciones  
}
```



# Estructuras de Control

**Bucles: while**

**Sintaxis:**

```
while (condición)
```

```
{
```

```
    //instrucciones
```

```
}
```

# Estructuras de Control

**Bucles: do..while**

**Sintaxis:**

```
do  
{  
    //instrucciones  
} while(condición)
```

# Estructuras de Control

**Bucles: with**

**Sintaxis:**

```
with(objeto)
```

```
{
```

```
    //instrucciones
```

```
}
```

# Estructuras de Control

**Bucles: switch**

**Sintaxis:**

```
switch(expresión)
```

```
{
```

```
    case etiqueta: //instrucciones;[break;]
```

```
    case etiqueta: //instrucciones;[break;]
```

```
    [default]: //instrucciones;
```

```
}
```

# Técnicas para insertar Scripts

- En la etiqueta HTML
- En el encabezado ó en el cuerpo de la una página Web
- En un Archivo .js

# Técnicas para insertar Scripts

## En la etiqueta HTML

- ESTA SUJETA A LA OCURRENCIA DE UN EVENTO Y SU SINTAXIS ES:

```
<etiqueta  
  evento1="instruccion1;instrucciónN;"  
  evento2="instruccion1;instrucciónN;">  
</etiqueta>
```

# Técnicas para insertar Scripts

- En el encabezado de la página Web

```
<script language="JavaScript">  
//DECLARAR VARIABLES GLOBALES  
//DECLARAR FUNCIONES  
//INSTRUCCIONES JAVASCRIPT  
</script>
```

# Técnicas para insertar Scripts

## En un Archivo .JS

a) Se debe crear un archivo con el código JavaScript.

b) Se debe enlazar las paginas donde se usará el código con el archivo .js por medio de la etiqueta script. La *sintaxis* es:

```
<script language="JavaScript"  
  src="URL_SCRIPT.JS">  
</script>
```