

# Tema 2. Tipos, variables y operadores

---

- Sintaxis
- Tipos de datos
- Variables
- Operadores aritméticos
- Operadores relacionales
- Operadores lógicos
- Operadores de cadena
- Operadores de asignación.
- Operadores de manejo de bits

# Sintaxis

- La sintaxis de JavaScript es muy similar a la de otros lenguajes de programación como Java o C.
- Las normas básicas que definen la sintaxis de JavaScript son las siguientes:
  - No se tienen en cuenta los espacios en blanco y las nuevas líneas
  - Se distinguen las mayúsculas y minúsculas
  - No se define el tipo de las variables
  - El estándar indica que cada sentencia debe terminar con el carácter de punto y coma. No obstante, JavaScript acepta que no se ponga el punto y coma siempre que se cambie de línea. Pero este uso no es recomendable.
  - Se pueden incluir comentarios
- JavaScript define dos tipos de comentarios
  - Los comentarios de una sola línea: se definen añadiendo dos barras oblicuas (//) al principio de la línea.
  - Los comentarios multilínea se definen encerrando el texto del comentario entre los símbolos /\* y \*/

# Tipos de datos

- *Javascript* reconoce seis tipos de valores diferentes: numéricos, lógicos, cadenas, nulos, indefinidos y objetos.
- Es un lenguaje débilmente tipado, esto es, una variable puede cambiar de tipo durante su vida.
- Las variables no se declaran especificando el tipo de dato que contendrán, será el propio interprete el que le asignará el tipo apropiado.

Number	Enteros o coma flotante.
Boolean	true o false.
String	Los tipos de datos cadena deben ir delimitados por comillas simples o dobles.
Undefined	Corresponde a una variable que ha sido creada sin asignarle un valor.
Null	Nulo, vacío o referencia no existente. Su único valor es null
Object	Objetos nativos y definidos por el usuario

# Ejemplo 1

```
<script type="text/javascript">
var cadena="Hola Caracola";
var cadnew=new String("Hola amigos");
var enter=2; var flotant=2.35;
var nodef; var nulo=null;
var vect=new Array(); var vectdef=[1,2,3];
document.write("<p>cadena es ",typeof cadena,"</p>");
document.write("<p>cadnew es ",typeof cadnew,"</p>");
document.write("<p>enter es ",typeof enter,"</p>");
document.write("<p>flotant es ",typeof flotant,"</p>");
document.write("<p>nodef es ",typeof nodef,"</p>");
document.write("<p>nulo es ",typeof nulo,"</p>");
document.write("<p>vect es ",typeof vect,"</p>");
document.write("<p>>window es ",typeof window,"</p>");
document.write("<p>vectdef es ",typeof (vectdef) ,"</p>");
document.write("<p>parseInt es ", typeof(parseInt) ,"</p>");
</script>
```

# Variables (I)

- Las variables se referencian por su nombre:
  - Debe empezar con una letra o “\_”
  - Los caracteres siguientes pueden ser números (0-9), letras mayúsculas o letras minúsculas (sin la ñ).
  - No puede contener espacios.
  - Debe ser único.
- Para definir una variable se utiliza la palabra reservada var, aunque si se inicializa no es necesario.

```
var variable;  
var variable=valor;  
variable=valor;
```

# Ámbito de las variables

---

- Variable local
  - Cuando se define dentro de una función. Su alcance es el ámbito de la función.
- Variable global
  - Cualquier variable definida fuera del cuerpo de una función. Su alcance es todo el documento.
  - Adicionalmente, si una variable se define dentro de la función sin la palabra reservada *var*, se convierte en global, pero se debe ejecutar la función para poder utilizarla.

# Literales

- Valores que puede tomar una variable.
- Numéricos:
  - Decimal: Entero que no empiece por 0.
  - Octal: Entero se escribe empezando con el dígito 0.
  - Hexadecimal: Empieza por 0x o 0X.
  - Coma flotante: 211.45 o 2.1145e+2
- Cadenas:
  - "La cadena va entre \" o ' ' "
  - 'La cadena va entre \' o " ' '
- Caracteres especiales
  - \n \t \b , etc.
- Booleanos
  - true y false

# Operadores aritméticos

Operador	Nombre	Ejemplo	
+	Suma	5+3	
-	Resta	5-3	
*	Producto	5*3	
/	División	5/3	
%	Resto	5%3	
++	Pre/post-incremento	++a	a++
--	Pre/post-decremento	--a	a--
-	Cambio de signo	-a	



## Ejemplo 2

```
<html>
<head>
<title>Operadores</title>
</head>
<body>
<script type="text/javascript">
    a = 8;
    b = 3;
    document.write("<p>",a + b,"<br />");
    document.write(a - b,"<br />");
    document.write( a * b,"<br />");
    document.write(a / b,"<br />");
    document.write(++a,"<br />");
    document.write(b--,"<br />");
    document.write(b,"</p>");
    document.write("<h1 style=\"text-align:center\">Fin</h1>");
</script>
</body>
</html>
```

# Operadores relacionales

- Comparan sus operandos y devuelve un valor lógico: `true` o `false`.
- Los operando pueden ser numéricos o cadenas.

Operador	Nombre
<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
==	Igual que
!=	Distinto que
===	Estrictamente igual que
!==	Estrictamente distinto que

## Ejemplo 3

```
<html>
<body>
  <script type="text/javascript">
    a = 8; b = 3; c = 3;
    document.write("<p>",a == b,"<br />");
    document.write(a != b,"<br />");
    document.write(a < b,"<br />");
    document.write(a > b,"<br />");
    document.write(a >= c,"<br />");
    document.write(b <= c,"</p>");
    document.write("<p>",3 == "3","<br />");
    document.write(3 === "3","</p>");
  </script>
</body>
</html>
```

# Operadores lógicos

- Los operadores lógicos se utilizan para combinar múltiples comparaciones en una expresión lógica.
- Sus operandos son valores lógicos y devuelve también un valor lógico: `true` o `false`.

Operador	Nombre
&&	AND
	OR
!	NOT

# Operadores de cadena

- Los valores cadena pueden compararse usando los operadores relaciones.
- Adicionalmente, se puede concatenar cadenas usando el operador +.

```
<html>
<body>
<script type="text/javascript">
    Nombre = "Pepet"
    document.write( "Hola " + Nombre + "." );
</script>
</body>
</html>
```

# Operadores de Asignación.

Operadores de asignación	
Operador	Nombre
=	Asigna un valor a una variable.
+=, -=, *=, /=, %=	$a+=b \rightarrow a=a+b$
&=,  =, ^=	$a\&=b \rightarrow a=a\&b$

# Operadores de Manejo de bits.

Operadores de manejo de bits			
Op.	Nombre		Op.      Nombre
&	AND		^      OR exclusivo
	OR		<<      Desplazamiento izq.
~	NOT		>>      Desplazamiento der.

## Ejemplo 4

```
<html>
<body>
<script language="JavaScript">
  a = 5;
  b = -3;
  a += 3;
  document.write("<p>" + a + "<br />");
  a -= 2; document.write(a + "<br />");
  a &= 3; document.write(a + "<br />");
  a |= 3; document.write(a + "<br />");
  a =~ 3; document.write(a + "<br />");
  b=b<<3; document.write(b + "<br />");
  b=b>>3; document.write(b + "</p>");
</script>
</body>
</html>
```