

### Value Types

Number	Unicamente valores numéricos
String	Caracteres entre comillas dobles o simples
Boolean	True / False

### Escape Characters

\b	Backspace
\f	Form feed
\n	New Line
\r	Carriage return
\t	Tab
\'	Apostrophe (')
\"	Doble quote (")
\\	Backslash (\)
\ xxx	xxx is an octal number
\ xXX	XX is a hexadecimal number

### Escape Characters

\b	Backspace
\f	Form feed
\n	New Line
\r	Carriage return
\t	Tab
\'	Apostrophe (')
\"	Doble quote (")
\\	Backslash (\)
\ xxx	xxx is an octal number
\ xXX	XX is a hexadecimal number

### Variables

Declaración	<code>var variable;</code>
Type	<code>typeof</code>

Si no se inicializa su valor es null/undefined  
Empieza con la inicial del tipo de dato

### Operators

<code>x + y</code>	<b>Adds</b> x and y (numeric)
	<b>Concatinates</b> x and y (string)
<code>x - y</code>	<b>Subtracts</b> y from x
<code>x * y</code>	<b>Multiplies*</b> x and y
<code>x/y</code>	<b>Divides</b> x by y
<code>x % y</code>	<b>Modulus</b> (division remainder)
<code>x++, ++x</code>	<b>Adds one</b> to x
<code>x--, --x</code>	<b>Subtracts one</b> from x

### Assignments

<code>x = y</code>	Set x to the value of y
<code>x += y</code>	Same as <code>x = x + y</code>
<code>x -= y</code>	Same as <code>x = x - y</code>
<code>x *= y</code>	Same as <code>x = x * y</code>
<code>x /= y</code>	Same as <code>x = x / y</code>
<code>x %= y</code>	Same as <code>x = x % y</code>

### Comparisons

<code>x == y</code>	Returns true if x and y are <b>equals</b>
<code>x === y</code>	Returns true if x and y are <b>*identical</b> (same type)
<code>x != y</code>	Returns true if x and y are <b>not equal</b>
<code>x &gt; y</code>	Returns true if x is <b>greater than</b> y
<code>x &lt; y</code>	Returns true if x is <b>less than</b> y
<code>x &gt;= y</code>	Returns true if x is <b>greater than or equal to</b> y
<code>x &lt;= y</code>	Returns true if x is <b>less than or equal to</b> y
<code>x &amp;&amp; y</code>	Returns true if <b>both are true</b>
<code>x    y</code>	Returns true if <b>either are true</b>
<code>!x</code>	Return true if x is <b>false</b>

### Conversions

<b>Number()</b>	Valor a numero	<code>Number (true) Number (false) Number (new Date());</code>
<b>String()</b>	Valor a cadena	<code>new String ("12 3");</code>



### Conditional Statement

<b>if</b>	<code>if (cond) { ... }</code>
<b>if... else</b>	<code>if (cond) { ... }{ else{ ... }</code>
<b>if... else if</b>	<code>if (cond){ ... }{ else if(cond) ...}{ else {...</code>
<b>switch</b>	<code>`switch(...){ case n: .. case n-1: ... default: ...}`</code>

### Function

<b>.write</b>	Escribe cadena de texto en ese lugar	<code>docume nt.w ri te( "Hola mundo");</code>
<b>alert</b>	Muestra una ventana con el texto que pasamos	<code>alert( "El numero es; " + num);</code>
<b>prompt</b>	Muestra dialogo con mensaje opcional solicitando que se introduzca un texto	<code>prompt ("In troduce un numero ")</code>
<b>.getElementById</b>	Recupera único elemento que coincide con el id	<code>docume nte.ge tEl eme ntB yId(id)</code>
<b>.getElements-ByTag</b>	Recupera todo lo que coincide con la etiqueta	<code>elemen t.g etE lem ent sBy Tag Nam e (tagName)</code>
<b>.getElements-ByName</b>	Recupera todo lo que coincide con el valor del atributo name	<code>docume nt.g et Ele men tBy Nam e(n ame );</code>

### Function (cont)

<b>eval</b>	Opera un codillo representado como cadena de caracteres	<code>eval("x " *"y") + " &lt;br &gt;"</code>
<b>isFinite</b>	Comprueba si el parametro es numero finito, realiza conversión	<code>isFini te( "Ho la") is Finite (123);</code>
<b>isNaN</b>	Detecta numeros invalidos	<code>isNaN (num)</code>
<b>parseInt/ parseFloat</b>	Lectura de un numero desde cadena	<code>parseI nt(cad) parseF - loa t(Cad)</code>

### Conditional Loop Statement

<b>While</b>	<code>While (...) { ... }</code>
<b>Do... While</b>	<code>do { ... } while (...)</code>

### Iterative Loop Statement

<b>For</b>	<code>for (...){ ... }</code>
------------	-------------------------------

### Vectors

#### Declaration

```
var ... = new Array ("...", "...");
var ... = ["...", "..."]
```

#### Acces

```
vector[i]
```

#### Recorrido

```
for (var in vector){ ... }
```

#### Properties

<b>length()</b>		<code>vector.le ngth()</code>
<b>push()</b>	Añade al final	<code>vector.pu sh( "...")</code>
<b>unshift()</b>	Añade al inicio	<code>vector.un shi t("...", "...")</code>
<b>pop()</b>	Elimina ultimo y devuelve	<code>vector.pop()</code>
<b>shift()</b>	Elimina primero y devuelve	<code>vector.sh ift()</code>

**.getElements-ByClassName** Recupera todo lo que coincide con el valor del atributo class

```
document.getElementByIdByClassName('class');
```

**.querySelector** Devuelve primer elemento del documento que coincida con los selectores

```
document.querySelector('selector')
```

**.querySelectorAll** Devuelve array de elementos dentro del array que coincidan con los selectores

```
document.querySelectorAll('selector')
```



By **cbgclara**  
[cheatography.com/cbgclara/](http://cheatography.com/cbgclara/)

Not published yet.  
Last updated 22nd September, 2023.  
Page 2 of 3.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>