# CÓDIGO JAVASCRIPT

console**.**log**(**`

----------------------------------------

TIPOS DE DATOS EN JAVASCRIPT

Primitivos: Se accede directamente al valor.

• string

• number

• boolean

• null

• undefined

• NaN

Compuestos: Se accede a la referencia del valor.

• object = {}

• array = []

• function () { }

• Class {}

• etc.

----------------------------------------

`**);**

//usar var es mala practica porque será global (hoisting)

**var** hola **=** "Hola Mundo"**;**

//usar let es buena prácica porque será de bloque

**let** hello **=** "Hello World"**;**

console**.**log**(**hola**);**

console**.**log**(**hello**);**

console**.**log**(window.**hola**);**

console**.**log**(window.**hello**);**

console**.**log**(**"---------DATOS PRIMITIVOS-----------"**);**

console**.**log**(**""**);**

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*var\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

**var** musica **=** "Rock"**;**

console**.**log**(**"Variable Música antes del Bloque"**,** musica**);**

//Esto es un bloque

**{**

**var** musica **=** "Pop"

console**.**log**(**"Variable Música dentro del Bloque"**,** musica**);**

**}**

console**.**log**(**"Variable Música después del Bloque"**,** musica**);**

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*let\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

**let** musica2 **=** "Rock"**;**

console**.**log**(**"Variable Música antes del Bloque"**,** musica2**);**

//Esto es un bloque

**{**

**let** musica2 **=** "Pop"

console**.**log**(**"Variable Música dentro del Bloque"**,** musica2**);**

**}**

console**.**log**(**"Variable Música después del Bloque"**,** musica2**);**

//CONSTANTES

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Const\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

**const** PI **=** 3.1416**;**

//Los const siempre se declaran con un valor inicial

//const NUEVE;

console**.**log**(**PI**);**

console**.**log**(**"\*\*\*\*\*\*\*\*Objeto y Array Const\*\*\*\*\*\*\*\*"**);**

/\*Los valores compuestos podremos modificarlos si son const, porque

los valores compuestos hacen referencia a una misma referencia

Las Funciones, Arreglos y Objetos usualmente son const\*/

**const** objeto **=** **{**

nombre**:** "Jon"**,**

edad**:** 35

**}**

**const** colores **=** **[**"blanco"**,** "negro"**,** "azul"**];**

console**.**log**(**objeto**);**

console**.**log**(**colores**);**

objeto**.**correo **=** "joinmircha@gmail.com"**;**

colores**.**push**(**"anaranjado"**);**

console**.**log**(**objeto**);**

console**.**log**(**colores**)**

//CADEMAS DE TEXTO AKA STRING

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*String\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

**let** nombre **=** "Jon"**;**

**let** apellido **=** "Mircha"**;**

**let** saludo **=** **new** String**(**"Hola Mundo"**);**

**let** lorem **=** "Lorem ipsum dolor sit amet consectetur adipisicing elit. Est voluptatum facilis cumque, exercitationem autem, iure suscipit harum hic aliquid inventore, minus beatae aspernatur debitis accusantium quidem incidunt blanditiis quaerat vero!"**;**

console**.**log**(**nombre**.**length**,**

apellido**.**length**,**

saludo**.**length**,**

nombre**.**toUpperCase**(),**

apellido**.**toLowerCase**(),**

lorem**.**includes**(**"amet"**),**

lorem**.**includes**(**"jon"**),**

lorem**.**trim**(),**

lorem**.**split**(**" "**)**

**);**

//TEMPLATE STRINGS

**let** nombre2 **=** "Jon"**;**

**let** apellido2 **=** "MirCha"**;**

//Concatenación

**let** saludo2 **=** "Hola mi nombre es " **+**

nombre2 **+** " " **+** apellido **+** "."**;**

console**.**log**(**saludo2**);**

//Interpolación de variables

//Template String

**let** saludo3 **=** `Hola mi nombre es ${nombre2} ${apellido2}.`**;**

console**.**log**(**saludo2**);**

//Template String es sensible a los saltos de línea

**let** ul **=** `

<ul>

<il>Primavera</il>

<il>Verano</il>

<il>Otoño</il>

<il>Invierno</il>

</ul>`**;**

console**.**log**(**ul**);**

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*NUMBERS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

//NÚMEROS (NUMBERS)

**let** a **=** 2**;**

**let** b **=** **new** Number**(**2**);**

**let** c **=** 7.19**;**

**let** d **=** "5.6"**;**

//Devuelve tipo de variable

console**.**log**(typeof** c**,** **typeof** d**);**

console**.**log**(**a**,** b**);**

//Límitar decimales

console**.**log**(**c**.**toFixed**(**1**));**

console**.**log**(parseInt(**c**));**

console**.**log**(parseFloat(**c**));**

console**.**log**(**a **+** b**);**

//Parsear la información

console**.**log**(**c **+** Number**.parseInt(**d**));**

console**.**log**(**c **+** Number**.parseFloat(**d**));**

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*BOOLEANOS\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

//BOOLEANOS

**let** verdadero **=** **true;**

**let** falso **=** **false;**

**let** v **=** Boolean**(true);**

**let** f **=** Boolean**(**falso**);**

//Trythy para saber que valores tienen a true y false

console**.**log**(**verdadero**,** falso**,** v**,** f**);**

console**.**log**(typeof** verdadero**,** **typeof** falso**);**

console**.**log**(**Boolean**(**0**))**

console**.**log**(**Boolean**(-**7**))**

console**.**log**(**Boolean**(**""**))**

console**.**log**(**Boolean**(**"a"**))**

//VALORES PRIMITIVOS: undefined, null & NaN

/\*

undefined - variable no inicializada

null - ausencia intencial de un valor

NaN - resultados númericos no validos (Not a Number)

\*/

**let** indefinida**;**

console**.**log**(**indefinida**);**

**let** nulo **=** **null;**

console**.**log**(**nulo**);**

**let** noEsUnNumero **=** "hola" **\*** 3.7**;**

console**.**log**(**noEsUnNumero**);**

console**.**log**(**"----------DATOS COMPLEJOS-----------"**);**

console**.**log**(**""**);**

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCIONES\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

//FUNCIONES

//Función Declarada

**function** estoEsUnaFuncion**()** **{**

console**.**log**(**"Uno"**);**

console**.**log**(**"Dos"**);**

console**.**log**(**"Tres"**);**

**}**

**function** unaFuncionQueDevuelveValor**()** **{**

console**.**log**(**"Uno"**);**

console**.**log**(**"Dos"**);**

console**.**log**(**"Tres"**);**

**return** 19**;**

**return** "La función a retornado una Cadena de Texto"**;**

**}**

estoEsUnaFuncion**();**

**let** valorDeFuncion **=** unaFuncionQueDevuelveValor**();**

console**.**log**(**valorDeFuncion**);**

**function** saludar**(**nombre **=** "Desconocido"**,** edad **=** 0**)** **{**

console**.**log**(**`Hola mi nombre es ${nombre} y tengo ${edad} años.`**);**

**}**

saludar**(**"Marco"**,** 20**)**

saludar**()**

//Funciones declaradas VS Funciones expresadas

//hoisting de las funciones

funcionDeclarada**();**

**function** funcionDeclarada**()** **{**

console**.**log**(**"Esto es una función declarada, puede invocarse en

cualquier parte de nuestro código, incluso antes de que la función

sea declarada"**)**

**}**

funcionDeclarada**();**

//Función anónima - expresada

//FUNCIONEXPRESADA();

**const** FUNCIONEXPRESADA **=** **function** **()** **{**

console**.**log**(**"Esto es una función expresada, es decir; una función que

se le ha asignado como valor a una variable, si invocamos esta

función antes de su definición JS nos dirá -Cannot Access

'FUNCIONEXPRESADA' before initialization-"**)**

**}**

FUNCIONEXPRESADA**();**

console**.**log**(typeof** FUNCIONEXPRESADA**);**

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*ARREGLOS\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

//ARRGLOS (ARRAYS)

//[]

**const** arreglo1 **=** **[];**

**const** arreglo2 **=** **[**1**,** **true,** "Hola"**,** **[**"A"**,** "B"**,** "C"**,[**1**,** 2**,** 3**]]];**

console**.**log**(**arreglo1**);**

console**.**log**(**arreglo2**);**

console**.**log**(**arreglo2**.**length**);**

console**.**log**(**arreglo2**[**2**]);**

//llamar a un subArreglo

console**.**log**(**arreglo2**[**3**][**2**]);**

console**.**log**(**arreglo2**[**3**][**3**][**0**]);**

//Nuevo estándar

**const** arreglo3 **=** Array**.of(**"X"**,** "Y"**,** "Z"**,** 9**,** 8**,** 7**);**

console**.**log**(**arreglo3**);**

//Inicializar Array con algún tamaño

**const** arreglo4 **=** Array**(**100**).**fill**(false);**

console**.**log**(**arreglo4**);**

//formas antiguas de declarar un Array

//----------------------------------------

**const** arreglo5 **=** **new** Array**();**

console**.**log**(**arreglo5**);**

**const** arreglo6 **=** **new** Array**(**1**,** 2**,** 3**,** **true,** **false);**

console**.**log**(**arreglo6**);**

//----------------------------------------

**const** colores2 **=** **[**"Rojo"**,** "Verde"**,** "Azul"**];**

console**.**log**(**colores2**);**

colores2**.**push**(**"Negro"**);**

console**.**log**(**colores2**);**

colores2**.**pop**();**

console**.**log**(**colores2**);**

//----------------------------------------

//forEach para un arreglo

colores2**.**forEach**(function(**e**,** index**)** **{**

console**.**log**(**`<li id = "${index}">${e}</li>`**)**

**})**

console**.**log**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*OBJETOS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);**

//OBJETOS

//En javaScript todo es un objeto

**let** a2 **=** **new** String**(**"Hola"**);**

console**.**log**(**a**);**

//----------------------------------------

//Es adecuando usar const cuando estamos usando objetos

//Para impedir que se cambié la instancia del objeto

**const** b2 **=** **{}**

console**.**log**(**b2**);**

//también podemos utilizar esta estructura descontinuada

**const** c2 **=** **new** Object**();**

console**.**log**(**c2**);**

//Crear un objeto

**const** marco **=** **{**

nombre**:** "marco"**,**

apellido**:** "muralles"**,**

edad**:** 20**,**

pasatiempos**:** **[**"Correr"**,** "Hacer ejercicio"**,** "Dar clases"**],**

soltero**:** **true,**

contacto**:** **{**

email**:** "marco@gmial.com"**,**

twitter**:** "marco\_muralles"**,**

movil**:** "12345678"

**},**

saludar**:** **function(){**

console**.**log**(**"Hola"**)**

**},**

decirMiNombre**:** **function(){**

console**.**log**(**`Hola me llamo ${this.nombre} ${this.apellido} y

tengo ${this.edad} años. Y me puedes seguir como

${this.contacto.twitter} en twitter.`**);**

**}**

**}**

//formas de acceder a los atributos del objetos

console**.**log**(**marco**);**

marco**.**saludar**();**

console**.**log**(**marco**[**"nombre"**]);**

console**.**log**(**marco**.**apellido**);**

console**.**log**(**marco**.**pasatiempos**);**

console**.**log**(**marco**.**pasatiempos**[**1**]);**

console**.**log**(**marco**.**contacto**.**twitter**);**

marco**.**decirMiNombre**();**

//este metodo de Object que permite listar las llaves del objeto Marco

console**.**log**(**Object**.**keys**(**marco**));**

//ver los valores del objeto

console**.**log**(**Object**.**values**(**marco**));**

//valida si se encuentra el atributo

console**.**log**(**marco**.**hasOwnProperty**(**"nombre"**));**

# CONSOLE

**----------------------------------------**

TIPOS DE DATOS EN JAVASCRIPT

Primitivos**:** Se accede directamente al valor**.**

• string

• number

• **boolean**

• **null**

• undefined

• NaN

Compuestos**:** Se accede a la referencia del valor**.**

• object **=** **{}**

• array **=** **[]**

• **function** **()** **{** **}**

• Class **{}**

• etc**.**

**----------------------------------------**

39 Hola Mundo

40 Hello World

41 Hola Mundo

42 undefined

44 **---------**DATOS PRIMITIVOS**-----------**

45

46 **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*var\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

49 Variable Música antes del Bloque Rock

54 Variable Música dentro del Bloque Pop

57 Variable Música después del Bloque Pop

59 **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*let\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

62 Variable Música antes del Bloque Rock

67 Variable Música dentro del Bloque Pop

70 Variable Música después del Bloque Rock

73 **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***Const**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

79 3.1416

81 **\*\*\*\*\*\*\*\***Objeto y Array Const**\*\*\*\*\*\*\*\***

94 **{**nombre**:** 'Jon'**,** edad**:** 35**}**

95 **(**3**)** **[**'blanco'**,** 'negro'**,** 'azul'**]**

100 **{**nombre**:** 'Jon'**,** edad**:** 35**,** correo**:** 'joinmircha@gmail.com'**}**

101 **(**4**)** **[**'blanco'**,** 'negro'**,** 'azul'**,** 'anaranjado'**]**

104 **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***String**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

110 3 6 10 'JON' 'mircha' **true** **false** 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Est voluptatum facilis cumque, exercitationem autem, iure suscipit harum hic aliquid inventore, minus beatae aspernatur debitis accusantium quidem incidunt blanditiis quaerat vero!' **(**30**)** **[**'Lorem'**,** 'ipsum'**,** 'dolor'**,** 'sit'**,** 'amet'**,** 'consectetur'**,** 'adipisicing'**,** 'elit.'**,** 'Est'**,** 'voluptatum'**,** 'facilis'**,** 'cumque,'**,** 'exercitationem'**,** 'autem,'**,** 'iure'**,** 'suscipit'**,** 'harum'**,** 'hic'**,** 'aliquid'**,** 'inventore,'**,** 'minus'**,** 'beatae'**,** 'aspernatur'**,** 'debitis'**,** 'accusantium'**,** 'quidem'**,** 'incidunt'**,** 'blanditiis'**,** 'quaerat'**,** 'vero!'**]**

128 Hola mi nombre es Jon Mircha**.**

133 Hola mi nombre es Jon Mircha**.**

144

mavera**</**il**>**

ano**</**il**>**

ño**</**il**>**

ierno**</**il**>**

146 **\*\*\*\*\*\*\*\*\*\*\*\*\*\***NUMBERS**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

154 number string

155 2 Number **{**2**}**

157 7.2

158 7

159 7.19

160 4

162 12.190000000000001

163 12.79

165 **\*\*\*\*\*\*\*\*\*\*\*\*\***BOOLEANOS**\*\*\*\*\*\*\*\*\*\*\*\*\*\***

173 **true** **false** **true** **false**

174 **boolean** **boolean**

175 **false**

176 **true**

177 **false**

178 **true**

188 undefined

191 **null**

194 NaN

196 **----------**DATOS COMPLEJOS**-----------**

197

198 **\*\*\*\*\*\*\*\*\*\*\*\*\***FUNCIONES**\*\*\*\*\*\*\*\*\*\*\*\*\*\***

203 Uno

204 Dos

205 Tres

209 Uno

210 Dos

211 Tres

219 19

222 Hola mi nombre es Marco y tengo 20 años**.**

222 Hola mi nombre es Desconocido y tengo 0 años**.**

234 Esto es una función declarada**,** puede invocarse en cualquier parte de nuestro código**,** incluso antes de que la función sea declarada

234 Esto es una función declarada**,** puede invocarse en cualquier parte de nuestro código**,** incluso antes de que la función sea declarada

244 Esto es una función expresada**,** es decir**;** una función que se le ha asignado como valor a una variable**,** si invocamos esta función antes de su definición JS nos dirá **-**Cannot access 'FUNCIONEXPRESADA' before initialization**-**

248 **function**

250 **\*\*\*\*\*\*\*\*\*\*\*\*\*\***ARREGLOS**\*\*\*\*\*\*\*\*\*\*\*\*\*\***

258 **[]**

259 **(**4**)** **[**1**,** **true,** 'Hola'**,** Array**(**4**)]**

260 4

261 Hola

263 C

264 1

268 **(**6**)** **[**'X'**,** 'Y'**,** 'Z'**,** 9**,** 8**,** 7**]**

272 **(**100**)** **[false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false,** **false]**

277 **[]**

280 **(**5**)** **[**1**,** 2**,** 3**,** **true,** **false]**

284 **(**3**)** **[**'Rojo'**,** 'Verde'**,** 'Azul'**]**

287 **(**4**)** **[**'Rojo'**,** 'Verde'**,** 'Azul'**,** 'Negro'**]**

290 **(**3**)** **[**'Rojo'**,** 'Verde'**,** 'Azul'**]**

295 **<**li id **=** "0"**>**Rojo**</**li**>**

295 **<**li id **=** "1"**>**Verde**</**li**>**

295 **<**li id **=** "2"**>**Azul**</**li**>**

298 **\*\*\*\*\*\*\*\*\*\*\*\*\*\***OBJETOS**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

303 2

309 **{}**

313 **{}**

336 **{**nombre**:** 'marco'**,** apellido**:** 'muralles'**,** edad**:** 20**,** pasatiempos**:** Array**(**3**),** soltero**:** **true,** …**}**

328 Hola

338 marco

339 muralles

340 **(**3**)** **[**'Correr'**,** 'Hacer ejercicio'**,** 'Dar clases'**]**

341 Hacer ejercicio

342 marco\_muralles

331 Hola me llamo marco muralles y tengo 20 años**.** Y me puedes seguir como marco\_muralles en twitter**.**

346 **(**8**)** **[**'nombre'**,** 'apellido'**,** 'edad'**,** 'pasatiempos'**,** 'soltero'**,** 'contacto'**,** 'saludar'**,** 'decirMiNombre'**]**

348 **(**8**)** **[**'marco'**,** 'muralles'**,** 20**,** Array**(**3**),** **true,** **{**…**},** ƒ**,** ƒ**]**

350 **true**