TYPES AND VARIABLES

FORMATO LÉXICO DEL LENGUAJE

VARIABLES Y TIPOS

OBJETOS

CONVERSIONES

FORMATO LÉXICO DEL LENGUAJE

Castellano

- puntos, comas, comillas, ñ, ll
- Ortografía

Javascript

- Diferencia mayúsculas y minúsculas
- Entiende tabulaciones y caracteres especiales (Unicode)
- Comentarios en Código: //, /**/
- Shortcuts (Linux/Windows + vscode)
- Punto y coma
- Identificadores de variables: letras, _ ó $

FORMATO LÉXICO DEL LENGUAJE

Javascript

- Literals: enteros, floats, strings, true, false, null, undefined
- Palabras reservadas

as	const	export	get	null	target	void
async	continue	extends	if	of	this	while
await	debugger	false	import	return	throw	with
break	default	finally	in	set	true	yield
case	delete	for	instanceof	static	try	
catch	do	from	let	super	typeof	
class	else	function	new	switch	var	

enum	implements	interface	package	private	protected	public

VARIABLES Y TIPOS

PRIMITIVOS: Números

- 64bit-floating-point ($\pm 1.7976931348623157 \times 10^{308}$ y $\pm 5 \times 10^{-324}$)
- Literals: (0, 2, 3)(0xff)(0b1010)(0o377)
- Float: (3.14) (.3774)(4.022e12)(99.9E-22))
- Aritmética: + - * /
- Nan, Infinity

NOTA!

Variables:

const

let/var

VARIABLES Y TIPOS

PRIMITIVOS: Números

- Library Math:

```
Math.pow(2,53)          // => 9007199254740992: 2 to the power 53
Math.round(.6)          // => 1.0: round to the nearest integer
Math.ceil(.6)           // => 1.0: round up to an integer
Math.floor(.6)          // => 0.0: round down to an integer
Math.abs(-5)            // => 5: absolute value
Math.max(x,y,z)         // Return the largest argument
Math.min(x,y,z)         // Return the smallest argument
Math.random()           // Pseudo-random number x where 0 <= x < 1.0
Math.PI                 // π: circumference of a circle / diameter
Math.E                  // e: The base of the natural logarithm
Math.sqrt(3)            // => 3**0.5: the square root of 3
Math.pow(3, 1/3)        // => 3**(1/3): the cube root of 3
Math.sin(0)             // Trigonometry: also Math.cos, Math.atan, etc.
Math.log(10)            // Natural logarithm of 10
Math.log(100)/Math.LN10 // Base 10 logarithm of 100
Math.log(512)/Math.LN2  // Base 2 logarithm of 512
Math.exp(3)             // Math.E cubed
```

NOTA!

Variables:

const

let/var

VARIABLES Y TIPOS

PRIMITIVOS: Strings

- Secuencia inmutable de valores de 16-bits
- Cero indexing
- Char -> string de tamaño 1
- UTF-16
- Se definen con: “, ‘, `
- Se pueden escaper caracteres especiales con \
- Operaciones: +, == or === or !=, < or >

VARIABLES Y TIPOS

PRIMITIVOS: Strings

- Separacion en líneas

```
// A string representing 2 lines written on one line:
'two\nlines'

// A one-line string written on 3 lines:
"one\
 long\
 line"

// A two-line string written on two lines:
`the newline character at the end of this line
is included literally in this string`
```

VARIABLES Y TIPOS

PRIMITIVOS: Strings

- Operaciones propias

```
let s = "Hello, world"; // Start with some text.

// Obtaining portions of a string
s.substring(1,4)        // => "ell": the 2nd, 3rd, and 4th characters.
s.slice(1,4)            // => "ell": same thing
s.slice(-3)             // => "rld": last 3 characters
s.split(", ")           // => ["Hello", "world"]: split at delimiter string

// Searching a string
s.indexOf("l")          // => 2: position of first letter l
s.indexOf("l", 3)       // => 3: position of first "l" at or after 3
s.indexOf("zz")         // => -1: s does not include the substring "zz"
s.lastIndexOf("l")      // => 10: position of last letter l

// Boolean searching functions in ES6 and later
s.startsWith("Hell")    // => true: the string starts with these
s.endsWith("!")         // => false: s does not end with that
s.includes("or")        // => true: s includes substring "or"

// Creating modified versions of a string
s.replace("llo", "ya")  // => "Heya, world"
s.toLowerCase()         // => "hello, world"
s.toUpperCase()         // => "HELLO, WORLD"
```

VARIABLES Y TIPOS

PRIMITIVOS: Strings

- Strings plantilla

let name = “maricarmen”;

let sayhello = `Hello ${ name }`;

VARIABLES Y TIPOS

PRIMITIVOS: booleans

- Verdadero o falso
- Muy útiles para expresiones condicionales
- Tabla de las leyes de Boole

CONDITION 1	CONDITION 2	AND	OR
FALSE	FALSE	FALSE	FALSE
FALSE	TRUE	FALSE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	TRUE	TRUE	TRUE

OBJETOS

- Todo lo demás
- Conjunto desordenado de primitivos
  - Arrays
  - Maps, sets, regex, dates, errors…

CONVERSIONES

Table 3-2. JavaScript type conversions

Value	to String	to Number	to Boolean
`undefined`	`"undefined"`	`NaN`	`false`
`null`	`"null"`	`0`	`false`
`true`	`"true"`	`1`	
`false`	`"false"`	`0`	
`""` (empty string)		`0`	`false`
`"1.2"` (nonempty, numeric)		`1.2`	`true`
`"one"` (nonempty, non-numeric)		`NaN`	`true`
`0`	`"0"`		`false`
`-0`	`"0"`		`false`

BIBLIOGRAFÍA

1. JavaScript: The Definitive Guide, 7th Edition
2. Eloquent JavaScript, 3th edition, Marijn Haverbeke
3. Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin Series)
4. https://www.guru99.com/v-model-software-testing.html, 20/03/2022
5. Modern c++ Programming with Test-Driven Development, Jeff Langr
6. Refactoring: Improve the design of existing Code, Martin Fowler
7. Game programming patterns, Robert Nystrom