

Arquitectura de Software

Tecnología - 2022

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

Presentación de equipo

Docentes de tecnología

- Pablo Geymonat
- Marcelo Caiafa
- Facundo Coto

Docente de teórico

- Gerardo Quintana
- Iván Echart

Introducción a la materia

- Hoja de Ruta
 - Recursos de estudio en aulas
 - Contacto (teams, aulas)
- Evaluaciones
 - 1 Obligatorio
 - 1 Parcial
 - Ejercicios
- Tecnologías
 - Node.js
 - Mongo
 - Redis con bull
 - MySQL

Introducción a Javascript

- ¿Qué es javascript?
 - Primera vez en 1995
 - Evolución
- Características del lenguaje
 - Interpretado (o compilado JIT)
 - De un solo hilo
 - Débilmente tipado
- ECMAScript
 - ¿Es lo mismo que JS?
 - ES6

ECMAScript

JS

Javascript: Tipos de datos

Tipos (2):

- Primitivos
- Objetos

Tipo de dato	Descripción	Ejemplo
String	texto	"hola mundo"
Number	integer o floating-point number	3, 3.234
Boolean	true o false	true, false
undefined	cuya variable no está inicializada	let variable;
null	valor nulo	let variable = null;
Symbol	cuya instancia es única e inmutable	let valor = Symbol('hola')
Object	colección de key-value pares	let persona = {};

undefined

vs

null

declared but not assigned `typeof name === 'object'` no value
runtime never sets null
programmer do it explicitly

```
let name;  
typeof name === 'undefined'
```

`let name = null;`

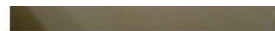
```
{  
  "name": undefined  
}  
{  
  "name": null  
}
```

Invalid JSON Valid JSON

`Boolean(undefined) => false`
`Boolean(null) => false`

`undefined == null => true`
`undefined === null => false`

undefined vs null



null



undefined



Javascript: Declaración de variables

- Sintaxis: Influenciada por C, C++ y Java.
- Case sensitive: x y X no son lo mismo para JS.

- ¿Qué es una variable?

Cómo declarar una variable:

var variable;
let variable;
const variable;

keyword	const	let	var
global scope	NO	NO	YES
function scope	YES	YES	YES
block scope	YES	YES	NO
can be reassigned	NO	YES	YES

Javascript: Operaciones

Asignación

```
let x = 1;
```

Operaciones aritméticas

- Suma (+)
- Resta (-)
- Multiplicación (*)
- Exponente (**)
- División (/)
- Módulo (%)
- Incrementar y decrementar (++ , --)

Operaciones de comparación

- Igualdad de variables: == & ===
- No igualdad: != & !==
- Mayor, menor: <, >, <=, >=
- Ternario: ?

Operaciones lógicas

- AND: &&
- OR: ||
- NOT: !

Javascript: Falsy & Truthy values

- Un valor puede traducirse a contextos booleanos, útiles para ahorrar código.
- Falsy se conoce como a valores que cuando entran en contexto booleanos, JS los convierte a valor false.
- No es necesario:
 - `if (variable != undefined).....`
 - `if (variable != null).....`
 - `if (variable != 0).....`
 - `if (variable != "").....`

Javascript: Type coercion

- Consecuencias del Tipado Blando(débilmente tipado)
- Explícito / implícito
- Se debe comprender como actúa en los siguientes casos

Ejemplo de Type de Coercion

```
[] == ![] // ✓ true
true == [] // ✗ false
!![] // ✓ true
true == ![] // ✗ false
false == [] // ✓ true
false == ![] // ✓ true
```

```
[] == 0 // ✓ true
[] == '' // ✓ true
!!"true" == !!"false" // ✓ true
!!"true" === !!"false" // ✓ true
NaN === NaN // ✗ false
Number.MIN_VALUE > 0 // ✓ true
```

TYPE COERCION

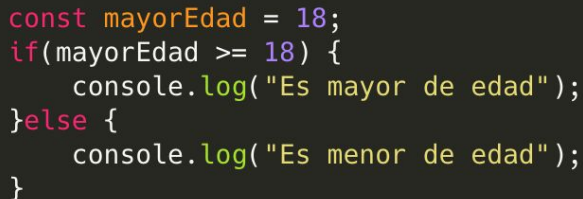
```
!!null // ✗ false
null == false // ✗ false
0 == false // ✓ true
'' == false // ✓ true
1 < 2 < 3 // ✓ true
3 > 2 > 1 // ✗ false
```

```
[] == '' // ✓ true
[0] == '' // ✗ false
[null] == '' // ✓ true
[null] == 0 // ✓ true
[undefined] == '' // ✓ true
[undefined] == 0 // ✓ true
```

<https://dorey.github.io/JavaScript-Equality-Table/>

Javascript: Estructuras de Control

- ¿Qué es una estructura de control?
- ¿Para qué sirven?



```
const mayorEdad = 18;  
if(mayorEdad >= 18) {  
  console.log("Es mayor de edad");  
}else {  
  console.log("Es menor de edad");  
}
```

Javascript: Funciones

¿Qué es una función?

- Bloque definido una vez y ejecutado varias veces.
- Puede recibir argumentos de los tipos ya vistos y... funciones!

- **Tipos de funciones**

- Comunes
 - `function (a,b){.....}`
- Anónimas
 - `const unaFuncion = function (a,b){.....}`
- Arrow functions
 - `let unaFuncion = (a, b)=>{.....}`
- Named functions
 - `const unaFuncion = unNombre function (a,b){.....}`