

The logo consists of the text 'DEV.FL' in a bold, white, sans-serif font. The 'F' is stylized with three small squares at its base. The logo is centered within a dark blue diamond shape.

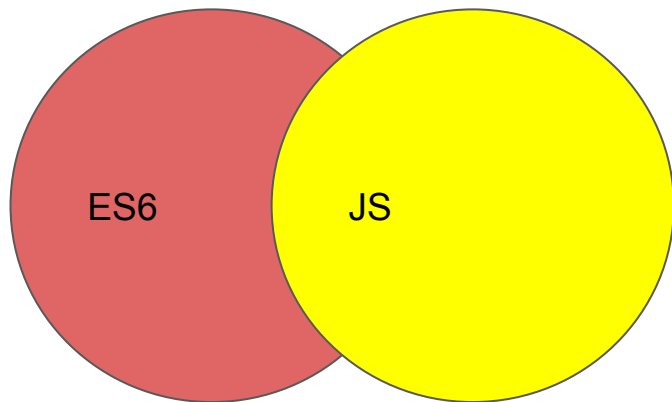
DEV.FL

ES6

¿JavaScript Moderno?

ECMAScript 6

dev



Cosas que faltan por implementar.

Cosas que los engines y navegadores agregan.

¿Qué es ECMAScript?

ECMAScript es el estándar y no el lenguaje.

Puede llegar a tomar mucho tiempo su implementación en un navegador.

European Computer Manufacturers Association



- Organización que se encarga de hacer estándares (CD fue una de ellas).
- Sede en Ginebra Suiza.

DEV.F

ECMAScript 6 (ES6) - ECMAScript 2015



Activa las notificaciones

Declaración de variables con let y const.

VAR vs LET vs CONST

	var	let	const
Reasignación	✓	✓	✗
Redeclaración	✓	✗	✗
Propiedad del objeto global (window)	✓	✗	✗
Function Scope	✓	✓	✓
Block Scope	✗	✓	✓
Hoisting	✓	✓	✓
TDZ (Temporal Dead Zone)	✗	✓	✓



Arrow functions (funciones de flecha) que proporcionan una sintaxis más concisa para definir funciones.

```
() => { }
```

```
(a) => { return a + a }
```

```
(a) => a + a
```

```
a => a + a
```

Template literals para
crear cadenas
de texto
interpoladas
de manera
más legible.

```
`string ${literal}`
```

Desestructuración de objetos y arreglos.



```
const {nombre, apellido, ciudad, gustos} =  
persona;  
console.log(nombre, apellido, ciudad)
```



```
const colores = ["Rojo", "Verde", "Azul"];  
let [r,g,b] = colores;  
console.log(`En la paleta RGB, R es ${r}, G es  
${g} y V es ${b}`)
```


Valores por defecto en los parámetros de las funciones.



```
JS

// Old Way
function beverage(drink) {
  drink = drink !== undefined ? drink : '☕'
}

// ✅ ES6 Way
function beverage(drink = '☕') {
}
```

Módulos para importar y exportar código entre archivos.

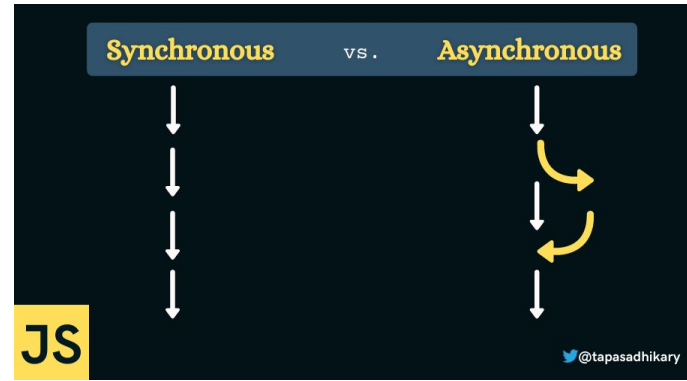
Las veremos más adelante <3



```
import {Perro, Oso} from './Animal';  
export class Zoologico {  
  constructor() {  
    const perro = new Perro();  
    const oso = new Oso();  
  }  
}
```

Promesas para trabajar con operaciones asincrónicas de manera más estructurada.

Las veremos más adelante <3



Clases y herencia basada en clases.

JS **ES6 CLASS**



```
var Meal = function(food) {  
  this.food = food  
}  
  
Meal.prototype.eat = function() {  
  return '😋'  
}
```

✗ OLD

```
class Meal {  
  constructor (food) {  
    this.food = food  
  }  
  
  eat() {  
    return '😋'  
  }  
}
```

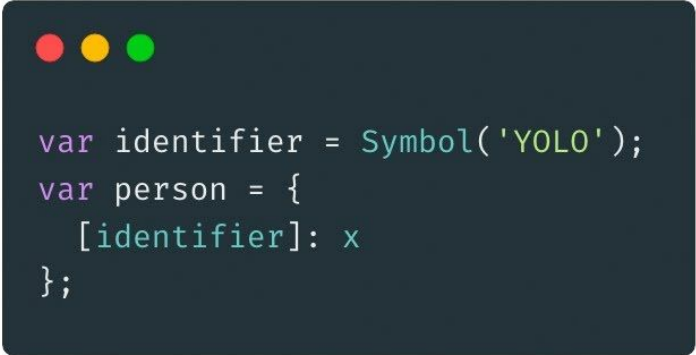
✓ NEW

 samantha_ming samanthaming.com  samanthaming

Las veremos en este módulo <3

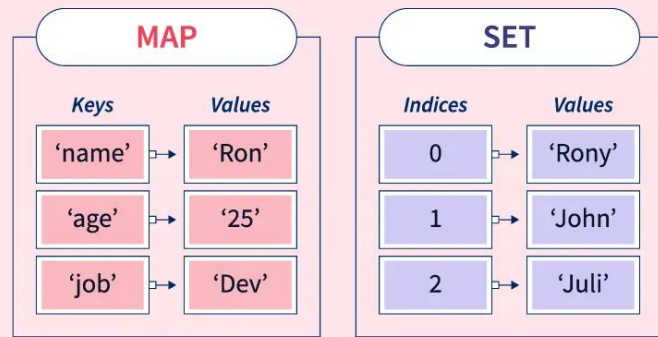
Symbol, un nuevo tipo de dato primitivo para crear identificadores únicos.

Symbol()



```
var identifier = Symbol('YOLO');  
var person = {  
  [identifier]: x  
};
```

Mapas (Map) y conjuntos (Set) para estructuras de datos más eficientes.



The logo for DEV.FL, featuring the text "DEV.FL" in a bold, white, sans-serif font. The "F" is stylized with a grid pattern. The logo is set against a dark blue diamond-shaped background.

DEV.FL

ECMAScript 7 (ES7) - ECMAScript 2016

**Inclusión del
operador de
exponenciación
(**) para cálculos
matemáticos.**

Funciones asincrónicas (async/await) para escribir código asincrónico de forma más legible y estructurada.

```
async function fetchMovies() {  
  const response = await fetch('/movies');  
  return await response.json();  
}
```

Las veremos más adelante <3

The logo for DEV.FL, featuring the text "DEV.FL" in a bold, white, sans-serif font. The "F" is stylized with a grid pattern. The logo is set against a dark blue diamond-shaped background.

DEV.FL

ECMAScript 8 (ES8) - ECMAScript 2017

**Object.values() y
Object.entries() para
obtener los valores y
pares clave-valor de un
objeto.**

**String padding methods
para agregar relleno a
cadenas de texto
(padStart() y padEnd()).**

Object.getOwnPropertyDescriptors()
para obtener todas las
propiedades de un
objeto, incluyendo
descriptores.

DEV.F

ECMAScript 9 (ES9) - ECMAScript 2018

**Operador de
propagación (...) en
objetos literales.**

Promise.prototype.finally()
para ejecutar una lógica
específica sin importar si
la promesa se resuelve o
se rechaza.

The logo for DEV.FL, featuring the text "DEV.FL" in a bold, white, sans-serif font. The "F" is stylized with a grid pattern. The logo is positioned inside a dark blue diamond shape on the left side of the slide.

DEV.FL

ECMAScript 2020 (ES2020)

**Nullish Coalescing Operator (??)
para proporcionar un
valor predeterminado
cuando un valor es null o
undefined.**

**Optional Chaining Operator (?)
para acceder a propiedades
de objetos anidados sin
lanzar errores si una
propiedad intermedia es
null o undefined.**

The logo for DEV.FL, featuring the text "DEV.FL" in a bold, white, sans-serif font. The "F" is stylized with a grid pattern. The logo is positioned inside a dark blue diamond shape on the left side of the slide.

DEV.FL

ECMAScript 2021 (ES2021)

**Operadores lógicos
encadenados (||= y &&=)
para asignación
condicional.**

Gracias