

ECMAScript

DEV.F
DESARROLLAMOS(PERSONAS);

dev

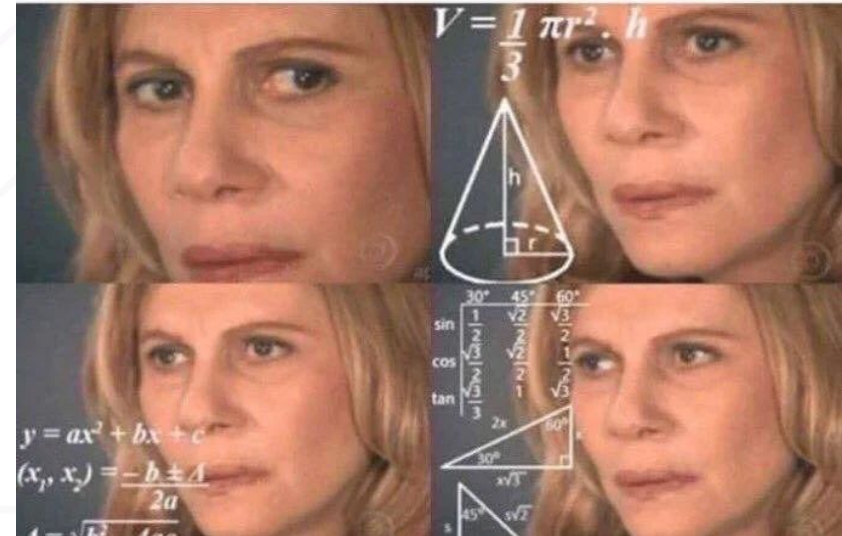
Repaso JS

DEV.F
DESARROLLAMOS(PERSONAS);

dev

¿Qué conocimientos previos necesito?

1. Tipos de datos (simples y complejos).
2. Operadores (asignación, aritméticos, comparación y lógicos).
3. Sentencias condicionales (if, else, switch).
4. Ciclos (for y while).
5. Objetos.
6. Arrays.
7. Funciones o métodos.



ECMAScript

DEV.F
DESARROLLAMOS(PERSONAS);

dev

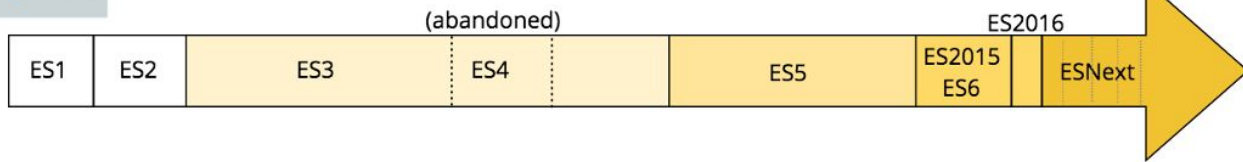
¿Qué es ECMA?

Ecma International es una organización internacional basada en membresías de estándares para la comunicación y la información. Adquirió el nombre Ecma International en 1994, cuando la European Computer Manufacturers Association (ECMA) cambió su nombre para expresar su alcance internacional.

Fue fundada en 1961 para estandarizar los sistemas informatizados en Europa.



ECMAScript



JS



ECMAScript

Es el estándar que la empresa del mismo nombre definió para JavaScript en el año 2015 (ES6) y encarga de regir como debe ser interpretado y cómo debe funcionar el lenguaje JavaScript.

Versión	Nombre	Publicación	Guía
ES11	ES2020	Junio del 2020	-
ES10	ES2019	Junio del 2019	-
ES9	ES2018	Junio del 2018	-
ES8	ES2017	Junio del 2017	-
ES7	ES2016	Junio del 2016	-
ES6	ES2015	Junio del 2015	-
ES5.1	ES5.1	Junio del 2011	-

Características más utilizadas de ECMA

- Let y const.
- Arrow functions (short and long form).
- For/of y for/in.
- Map vs forEach.
- Clases.
- Promises.
- Default parameters.
- Spread operator.
- Rest operator.

Características más utilizadas de ECMA

- Object entries y values.
- JS Modules.
- Literal templates / Template strings.
- Map y Set. => Estructuras de datos.

Características más utilizadas de ECMA

- Async / await.
- Asynchronous Iteration.
- Promise Finally.
- Object Rest Properties.
- New RegExp Features (<https://regexp.com/>).

Arrays

- Map y forEach(diferencias).
- Reduce.
- Push.
- Splice.
- Includes.
- Filter.
- Find.
- FindIndex.

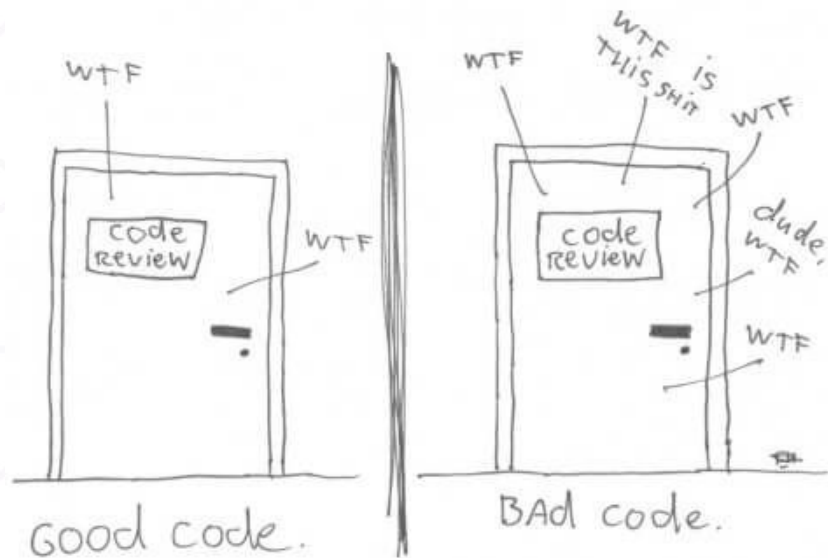
```
[1, 2, 3].push(4) // [1,2,3,4]
[1, 2, 3].pop() // [1,2]
[1, 2, 3].shift() // [2,3]
[1, 2, 3].unshift(0) // [0,1,2,3]
['a', 'b'].concat('c') // ['a','b','c']
['a', 'b', 'c'].join('-') // a-b-c
['a', 'b', 'c'].slice(1) // ['a','b']
['a', 'b', 'c'].indexOf('b') // 1
['a', 'b', 'c'].includes('c') // true
[3, 5, 6, 8].find((n) => n % 2 === 0) // 6
[2, 4, 3, 5].findIndex((n) => n % 2 !== 0) // 2
[3, 4, 8, 6].map((n) => n * 2) // [6,8,16,12]
[1, 4, 7, 8].filter((n) => n % 2 === 0) // [4,8]
[2, 4, 3, 7].reduce((acc, cur) => acc + cur) // 16
[2, 3, 4, 5].every((x) => x < 6) // true
[3, 5, 6, 8].some((n) => n > 6) // true
[1, 2, 3, 4].reverse() // [4,3,2,1]
[3, 5, 7, 8].at(-2) // 7
```

Bonus

- **Paso por referencia y valor.**
- Destructuring.
- Tipos de salida de una función.
- **Valor por defecto de una variable (||).**
- **Encadenamiento opcional (cortocircuito ?).**
- Nullish operator (??).
- Hoisting.
- Tablas de verdad y tipos de operadores.
- Tipos de funciones en JS.
- **Delete, type of, instanceof.**

Clean code

The only valid measurement
of code quality: WTFs/minute



(c) 2008 Focus Shift

Buenas prácticas

- Tener un código (honor) para escribir mi código :v => Eslint, Prettier.
- Nombrado de variables y funciones.
- Reutilización y definición de funciones.
- Inmutabilidad.
- Longitud de línea e indentación.
- Uso correcto de var, let y const.
- Documentación.
- Comparaciones de valores null, undefined, vacío (0 y emptyString), NaN, etc.
- Validación de JSON.
- Uso de dummies y constantes.
- Normalización de datos (patrón selector).