ECMAScript

Última versión:

ES13 o ECMAScript 2022

Novedades:

1-Await a nivel superior

A partir de ahora en JS gracias al Ecmascript 2022 podremos usar el await sin necesidad de utilizarlo dentro de una función asíncrona con async.

```
[1,2,3,4,5].at(2)  // devuelve 3

[1,2,3,4,5].at(-1)   // devuelve 5
```

2-Object.hasOwn()

Este nuevo método comprueba si una propiedad está incluida dentro de un objeto o existe dentro del objeto.

```
const miObjeto = {
  titulo: "Master en React",
  descripcion: "Aprende React con Hooks desde cero",
  autor: "Víctor Robles",
  fecha: 2022
};

Object.hasOwn(miObjeto, "categoria"); //devuelve false
Object.hasOwn(miObjeto, "autor"); //devuelve true
```

3-Propiedades y métodos privados en ES2022

Ahora podremos crear atributos y métodos privados en nuestras clases de JavaScript ES2022 o ES13 usando la # delante del nombre de tu propiedad o método.

```
class Usuario{
  #nombre;

  constructor(nombre) {
    this.#nombre = nombre;
  }

  #apellidos(apellido) {
    return this.#nombre + ' ' + apellido;
  }

  mostrar(apellido) {
    console.log(this.#apellidos(apellido));
  }
}

const usuario = new Usuario("victor");
console.log(usuario.#nombre);           // Error
console.log(usuario.#apellidos("robles"));      // Error
usuario.mostrar("robles");                     // victor robles
```

4-.at()

El nuevo método at() para JavaScript permite seleccionar el índice de un array de manera flexible. Solo hay que pasarle como parámetro el índice del valor que queremos conseguir.

```
[1,2,3,4,5].at(2)  // devuelve 3

[1,2,3,4,5].at(-1)   // devuelve 5
```

5-error.cause

Esta novedad nos permite especificar que causó un error.

```
try {
    // tus instrucciones
} catch (error) {
    throw new Error(
      `Carga de datos fallida`,
      {cause: error}
    );
}
```