

js ES6+

Denis Pacheco



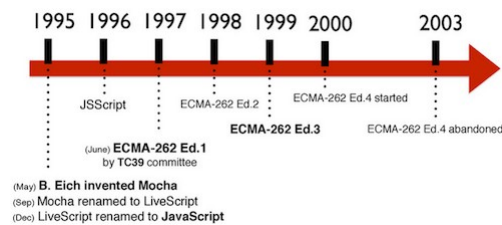
Que es ES6?

- **ECMAScript v6** (Abreviado como ES6 o ES2015) es el estándar que sigue JavaScript desde Junio de 2015. Hasta ese momento la versión de JS que estábamos usando en nuestros navegadores y Node.js, era la v5.



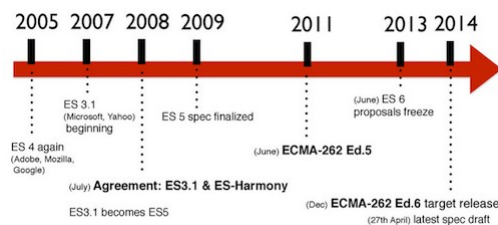
La evolución de JavaScript

- En 1997 se crea un comité (TC39) para estandarizar JavaScript por la European Computer Manufacturers' Association, ECMA. Se diseña el estándar del DOM (Document Object Model) para evitar incompatibilidades entre navegadores. A partir de entonces los estándares de JavaScript se rigen por ECMAScript.



La evolución de JavaScript

- En 1999 aparece la 3a versión del estándar ECMAScript, que se mantendría vigente hasta hace algunos años. Hubo pequeños intentos de escribir la versión 4, pero hasta 2011 no se aprobó y se estandarizó la versión 5 (ES5) que es la que usamos hoy en día.



La evolución de JavaScript

- ▶ Desde Junio 17, 2015, ECMA International publico la sexta versión principal de ECMAScript, que oficialmente se llama ECMAScript 2015, y que inicialmente se denominó ECMAScript 6 o ES6. Desde entonces, los estándares ECMAScript están en ciclos de lanzamiento anuales.



JS ES6

- ▶ A partir de la versión 6 de Javascript, se incluyeron muchas funcionalidades e instrucciones nuevas
- ▶ A esto podemos llamarlo “Javascript Moderno”
- ▶ Actualmente todos los navegadores soportan la versión ES6 de javascript, pero antiguamente (o actualmente si lo que se desea es retro compatibilidad) se usaba un pollyfill, que es un seistema que “traduce” el código actual, a uno compatible con versiones antiguas del lenguaje.



JS ES6

- ▶ Por ejemplo, como podríamos saber si una palabra ingresada comienza con la letra "a"?

- ▶ Con lo que actualmente sabemos:

```
if( palabra[0]=="a" | palabra[0]=="A"){
    console.log("Si empieza con A!")
}
```

- ▶ Con ES6 sería:

```
if(palabra.toUpperCase().startsWith("A")){
    console.log("Si empieza con A!")
}
```



Principales Novedades de ES6

- ▶ **Let y Const:** con estas declaraciones de variables (en lugar de var), podemos encapsular su uso para que no sea accesible más allá del ámbito que queramos:

```
if(true){
    var x=1;
    let y=2;
    console.log("valor de x:" + x);
    console.log("valor de y" + y);
}

console.log("valor de x:"+x);
console.log("valor de y"+y);
```

- ▶ La clave de **let** es que restringe su existencia al ámbito donde ha sido declarada (cualquier ámbito expresado con unas llaves), por lo que no "contaminará" el resto del código. (dejará de existir cuando se termine de ejecutar el bloque donde existe)



Principales Novedades de ES6

- **Const** nos permitirá declarar variables que NO podrán ser cambiadas mas adelante en el código. Su valor quedará fijo y solo puede asignado en la declaración.

```
const pi=3.14;  
pi=4;
```



Principales Novedades de ES6

- **this**: con esta palabra clave, se puede referenciar directamente un método o variable (objeto) del ámbito donde estamos trabajando.
- Esto se utiliza cuando tengamos declaraciones de distintas funciones pero con el mismo nombre en distintas clases u objeto.
- Ejemplo:

```
function foo(){  
  console.log("fuera del objeto")  
}  
  
var obj = {  
  foo : function() {  
    console.log("dentro del objeto")  
  },  
  bar : function() {  
    document.addEventListener("click", (e) => this.foo());  
    document.addEventListener("click", (e) => foo());  
  }  
}  
  
obj.bar();
```



Principales Novedades de ES6

- **Funciones Arrow** (flecha): con esta característica, se puede crear una sintaxis mas corta para crear funciones en javascript.
- Ejemplo:
- Versión tradicional:

```
function sumar100v1(a){
  return a+100;
}
```

Versión ES6:

```
let sumar100v2 = a=>a+100;
```

testeando

```
console.log(sumar100v1(50));
console.log(sumar100v2(50));
```



Principales Novedades de ES6

- ejemplo2, si quisiéramos un arreglo y luego enviarlo a pantalla multiplicando cada elemento por 2:

```
function escribirNumerosv5(){
  var lista=[0,1,2,3,4];
  for(var i=0;i<lista.length;i++){
    console.log( lista[i]*2+"\n");
  }
}

escribirNumerosv5();
```

- Con ES6:

```
var escribirNumerosv6=[0,1,2,3,4].map((numero)=>console.log(numero*2)+"\n");
```

Pueden ver mas información de funciones Arrow en :
https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Funciones/Arrow_functions



Principales Novedades de ES6

- **Spread** (operador de propagación)
- Permite que una expresión sea expandida en situaciones donde se esperan múltiples elementos
- Por ejemplo, si quisiéramos unir dos arreglos:

```
let primero=[1,2,3,4];
let segundo=[5,6,7];
//unir ambos arreglos
primero.push(segundo);
console.log(primero);
```

```
(5) [1, 2, 3, 4, Array(3)]
0: 1
1: 2
2: 3
3: 4
4: (3) [5, 6, 7]
  length: 5
  __proto__: Array(0)
```

► Lo que está mal

- Con ES6, spread:

```
primero.push(...segundo);
console.log(primero);
```

```
(7) [1, 2, 3, 4, 5, 6, 7]
0: 1
1: 2
2: 3
3: 4
4: 5
5: 6
6: 7
  length: 7
  __proto__: Array(0)
```

► Lo que está bien



Principales Novedades de ES6

- Otro ejemplo:

```
let numeros=[1,2,3];
function sumar(a,b,c){
  return a+b+c;
}
console.log(sumar(...numeros));
```

- En este caso, tenemos una función que suma 3 números, pero estos números los tenemos en un arreglo, y sin embargo, al llamar a la función, no le pasamos los tres parámetros `sumar(1,2,3)`, sino que le pasamos el arreglo pero a través del operador spread (los tres puntos...). Esto hace que automáticamente la función tome los tres números almacenados dentro del arreglo.



Principales Novedades de ES6

- ▶ **Desestructuración:** supongamos que necesitamos guardar cada elemento de un arreglo en una variable individual.
- ▶ Con la forma clásica:

```
// Método antiguo
const myArray = ['apple', 'pear', 'orange', 'banana'];
let fruit1 = myArray[0];
let fruit2 = myArray[1];
let fruit3 = myArray[2];
let fruit4 = myArray[3];
```

- ▶ Con ES6:

```
let [fruit5, fruit6, fruit7, fruit8] = myArray; // mucho mejor, ¿no?
```

- ▶ Testeando, nos da el mismo resultado:

```
console.log(fruit2);
console.log(fruit6);
```

```
pear
pear
>
```



ES6

- ▶ Para una mayor referencia, pueden visitar el sitio:
- ▶ <http://es6-features.org/> (en inglés, específico de ES6. sugerido si lo que necesitan es ver ejemplo concretos sin mucha explicación)
- ▶ <https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia>
- ▶ (en español, pero es genérico, no específico de ES6)
- ▶ <https://desarrolloweb.com/manuales/manual-de-ecmascript-6.html>
- ▶ Para un manual un poco mas explicativo de algunas (otras) funciones

