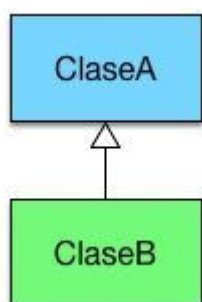


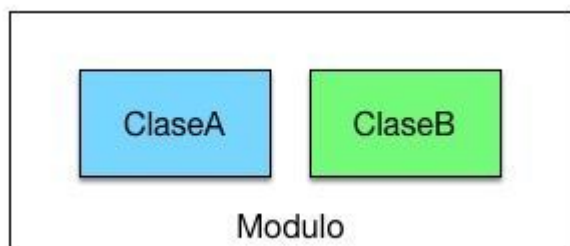
JavaScript ES6 es un cambio muy importante a nivel de programación. Hay muchas cosas que llegan al lenguaje y que realmente hacían mucha mucha falta. Vamos a revisar algunas de las que en mi opinión son las más destacables.

JavaScript ES6

Clases: Por fin tenemos **clases y herencia** de una forma similar a las que tenemos en los lenguajes más habituales.



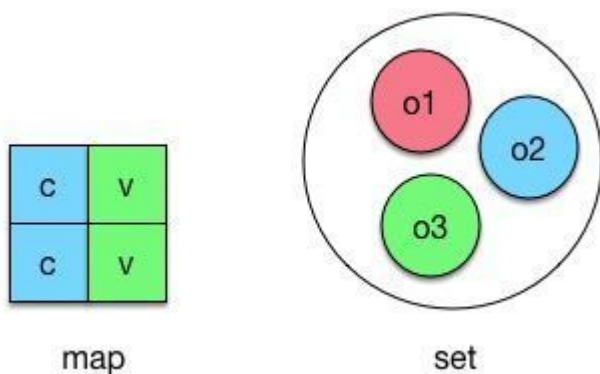
Módulos : La necesidad de **modularizar de una forma coherente** los diferentes componentes que construimos es algo que hace ya mucho tiempo que se echaba en falta. Su llegada es una de las más deseadas.



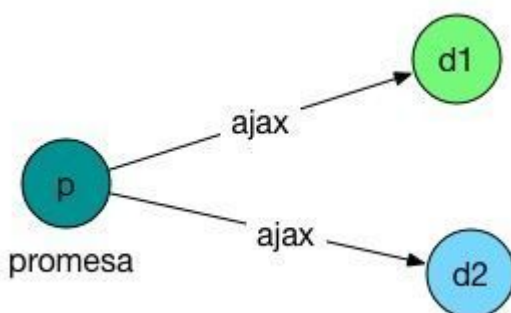
Arrow Functions : Las **funciones flecha** que se encargan de aportarnos una sintaxis lambda .Es algo que en un lenguaje funcional no podía faltar.

$$(a,b) \Rightarrow a+b$$

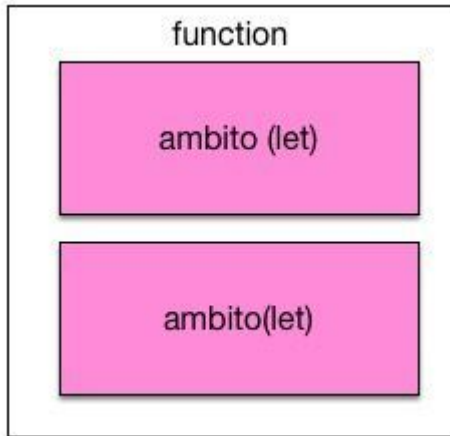
Las colecciones : El trabajar con arrays como único tipo abstracto de dato potente tiene sus limitaciones. Se echaba de menos la llegada de clases como **Map** y **Set**.



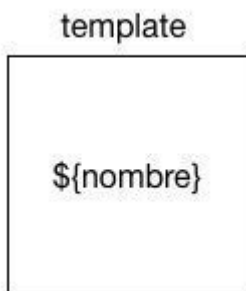
Promesas : La inclusión de un API de promesas a nivel del lenguaje se agradece mucho ya que casi todos los programas de JavaScript **son eminentemente asíncronos**.



Declaraciones let: La posibilidad de **declarar ámbitos de variables** sin tener que usar funciones , es algo que todos vamos a agradecer también.



Templates : El uso de **plantillas** es algo muy natural en todos los lenguajes de programación y que JavaScript lo soporte de forma nativa se agradece.



Constantes: Increíble o no hasta JavaScript ES6 el soporte de constantes no existía en JavaScript .Su inclusion era obligatoria.

JavaScript ES6 , un ejemplo

¿Cuánto nos supone la evolución de JavaScript para los desarrolladores en el día a día?. Esa es una pregunta interesante. Vamos a ver un ejemplo muy sencillo que ayuda a clarificar. Supongamos que tenemos un array al cual queremos eliminar un elemento, el siguiente código realiza la operación con ES5 y no es nada intuitivo.

```
var lista=[1,2,3,4];  
console.log(lista.indexOf(3));  
lista.splice(2,1);  
console.log(lista);
```

Con ES6 podríamos hacer algo muy parecido usando la clase Set y el resultado es mucho más natural.

```
var conjunto= new Set([1,2,3,4]);  
conjunto.delete(3);  
console.log(conjunto);
```

El uso de JavaScript ES6 y los transpiladores a ES5 se ira extendiendo de forma rápida ya que incluso en operaciones muy básicas ES6 es superior y aporta ventajas.

Otros artículos relacionados: [¿Qué es un JavaScript Bundle?](#) , [JavaScript Observers vs Arrays](#) , [JavaScript ES6 fetch API](#)