



# Características de ES6 en TypeScript




# Variables let y const

Dentro de TypeScript y en estándares de ES6 en adelante, se recomienda no utilizar `var` para declarar variables y mejor optar por `let` y `const`.

También podemos implementar buenas prácticas con variables `const` en caso de que su valor no vaya a cambiar y `let` para variables que en algún momento determinado vaya a cambiar.

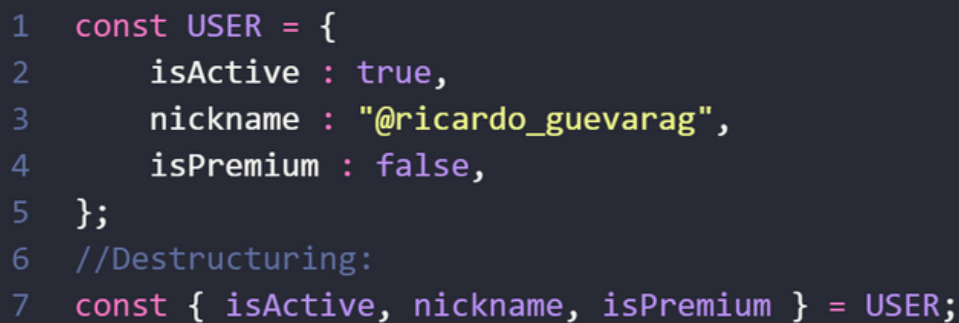
Para darle soporte a ES6 con TypeScript es necesario cambiar el target en el `tsconfig.json`



```
1  "target": "es6",
```

# Desestructuración de objetos

La desestructuración en ES6 es una característica de súper utilidad que nos permite acceder a propiedades de un objeto sin utilizar la notación por punto o corchetes.



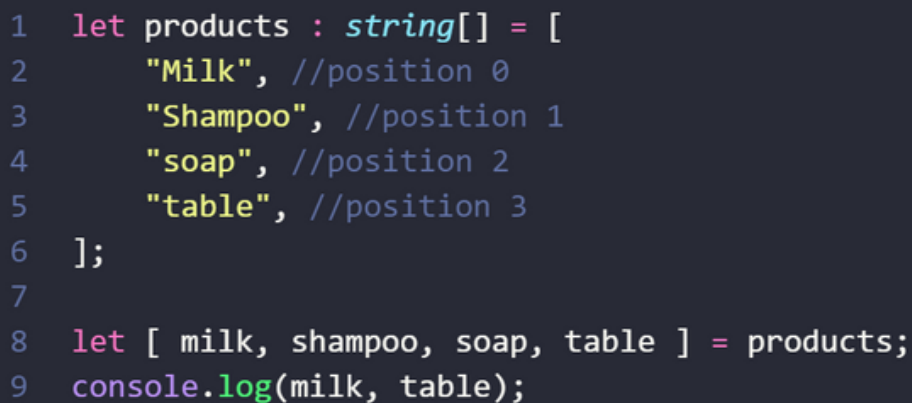
```
1  const USER = {  
2      isActive : true,  
3      nickname : "@ricardo_guevarag",  
4      isPremium : false,  
5  };  
6  //Destructuring:  
7  const { isActive, nickname, isPremium } = USER;
```

Declaramos otra constante y dentro de las llaves escribimos las propiedades del objeto, después la igualamos al nombre del objeto.

Otra ventaja que nos brinda la desestructuración es a la hora de recibir parámetros en las funciones, puedo indicar mediante desestructuración la propiedad que quiero recibir en mi función.

# Desestructuración de arreglos

Para desestructurar un arreglo basta con declarar una variable con corchetes e igualarla a nuestro arreglo, dentro de los corchetes le indico las posiciones que quiero obtener de mi arreglo, recordando que empiezan de la posición 0.




```
1  let products : string[] = [  
2      "Milk", //position 0  
3      "Shampoo", //position 1  
4      "soap", //position 2  
5      "table", //position 3  
6  ];  
7  
8  let [ milk, shampoo, soap, table ] = products;  
9  console.log(milk, table);
```

# El ciclo for of


Nos sirve para trabajar con arreglos que tienen un límite específico, nos ayuda a recorrer el arreglo de una manera más sencilla que un ciclo for tradicional.

Tenemos el siguiente arreglo:




```
1 let cars : string[] = ["Ferrari", "Mustang", "Camaro"];
```

Para recorrerlo e imprimir cada uno de sus elementos con un ciclo for tradicional sería de la siguiente manera:



```
1 for (let i = 0; i < cars.length; i++) {  
2     console.log(cars[i]);  
3 };
```

Con un ciclo for of nos podemos ahorrar un par de líneas, también el código se vuelve más legible y fácil de entender:



```
1  for (let car of cars) {  
2      console.log(car);  
3  }
```