

Bootcamp Java Developer

Fase 2 - Java Web Developer
Módulo 18

Funciones flecha

Funciones flecha

Las **funciones flecha** son una forma alternativa de escribir una función del ES6. Proveen una serie de ventajas que no las hacen una simple sintaxis alternativa para funciones.

- Permiten **ahorrar código**.
- Permiten **expresar funciones como expresiones**.
- Mejoran la gestión del ámbito de las **variables** (no cambian el **this***).



* : la palabra **this** es una herramienta para gestionar las variables en JavaScript. Si bien es una herramienta avanzada, la mencionamos aquí para que tengas esta referencia cuando vayas a estudiarla.

Ahorrar código

En este caso, la función flecha permite ahorrar código omitiendo la palabra reservada **function**.

Las funciones flecha son expresiones que **representan funciones**, por lo que, para ser utilizadas como una función, **deben asignarse a una variable**.

Su sintaxis es muy simple:

(parámetros) => proceso

```
1  function dividir(n1,n2) {
2      if(n2 > 0)
3          return n1 + n2;
4      else
5          return false;
6  }
7
8
9  // Con funciones flecha
10 const dividir = (n1, n2) => {
11     if(n2 > 0)
12         return n1 + n2;
13     else
14         return false;
15 }
16
```

Escribir funciones como expresiones

Las funciones flecha permiten ahorrar código en las funciones que sólo retornan un valor, que no tienen un proceso complejo dentro. Esto es especialmente útil cuando tengo una expresión (como un texto HTML) que se necesita parametrizar.

En este caso, **podemos omitir las llaves y la palabra reservada return en las funciones flecha.**

```
1  function itemLista(nombre) {  
2    |   return '<li>'+nombre+'</li>'  
3  }  
4  
5  
6  // Con funciones flecha  
7  const itemLista = (nombre) => '<li>'+nombre+'</li>'  
8
```

Estos dos códigos son equivalentes.

String template ES6

String template ES6

Las ***String Template*** (o Plantillas de texto, en español) son una forma mucho más simple mediante la cual podemos **concatenar texto con variables en ES6**.



Las ***String Templates*** sirven para:

- Insertar un código HTML parametrizado mediante JavaScript.
- Reutilizar un texto largo pero con pequeños cambios en cada implementación.
- Escribir un gran texto multilínea en JavaScript.

Podemos observar que es un texto común y corriente con tres salvedades.

- a. El **carácter delimitador** es diferente: ```
- b. En vez de **concatenar variables** cerrando el texto y volviéndolo a abrir, las vamos a concatenar mediante **`${variable o expresión}`**. De esta forma evitamos cerrar y volver a abrir el texto.
- c. Es el único texto que **soporta multilínea**.

Podemos combinar este tipo de texto con funciones *flecha* para optimizar su uso.

```
1  const itemList = (url, nombre) => `  
2  <li>  
3    <a href="${url}">${nombre}</a>  
4  </li>`  
5
```


**¡Sigamos
trabajando!**