

2.2.2. Funciones flecha (*arrow*)

Como su nombre indica, son funciones definidas usando una flecha =>, pero su comportamiento es un poco diferente a las funciones tradicionales en varios aspectos.

Las funciones flecha tienen una sintaxis más corta que una expresión de función convencional. Además, son anónimas, no están relacionadas con métodos y no pueden ser usadas como constructores.

```
let nuevaFuncion = () => {  
  //código de la función  
}
```

Como se ve en el ejemplo, no solo se usa la flecha, sino que los paréntesis donde se colocarían los parámetros de la función también se mueven de lado, colocándolos antes de la flecha. La invocación se realizaría del mismo modo que conocemos:

```
nuevaFuncion();
```

El tratamiento de los parámetros se realiza como hasta ahora, simplemente se colocan entre los paréntesis:

```
let saludar = (nombre, tratamiento) => {  
  alert('Hola ' + tratamiento + ' ' + nombre)  
}  
  
//invocación  
saludar('Jana', 'srta.');
```

En caso de pasar solo un parámetro, la función se simplifica, puesto que nos podemos ahorrar los paréntesis como en el ejemplo siguiente:

```
let cuadrado = numero => {  
  return numero * numero;  
}
```

También nos podemos ahorrar algún carácter extra si solo tenemos una línea de código en nuestra función. En este caso no habría que poner las llaves de apertura y cierre de la función.

```
let saludar = (nombre, tratamiento) => alert('Hola ' + tratamiento + ' ' + nombre).
```

Adicionalmente, en caso de tener una única línea de código y que la función devuelva un valor, también nos podemos ahorrar la palabra *return* (además de las llaves, como ya hemos visto). La función del cuadrado quedaría así:

```
let cuadrado = numero => numero * numero;
```

La nueva forma es mucho más compacta que la que teníamos anteriormente. Lo veremos más claro comparando el último fragmento de código con la declaración de la función de la forma tradicional en ES5:

```
// ES5
var cuadrado = function(numero) {
  return numero * numero;
}
```

Si una función no tiene que recibir ningún valor por parámetro, entonces se dejan los paréntesis solo. Por ejemplo:

```
let getBienvenida = () => "Hola mundo";
```

Que sería lo mismo que hacer esto con ES5:

```
var getBienvenida = function () {
  return "Hola mundo";
};
```