

# Los callbacks

“

Un **callback** es una **función** que se pasa como **parámetro** de otra **función**.

La función que lo recibe es quien se encarga de **ejecutarla** cuando sea necesario.



”

# El callback anónimo

En este caso, la función que pasamos como **callback** no tiene nombre. Es decir, es una **función anónima**.

Como las **funciones anónimas** no pueden ser llamadas por su nombre, debemos escribirla dentro de la función que se encargará de llamar al callback.

```
{ } setTimeout( function(){  
    console.log('Hola Mundo!')  
} , 1000)
```

# El callback definido

La función que pasamos como **callback** puede ser una función previamente **definida**. Al momento de pasarla como parámetro de otra función, nos referiremos a la misma por su nombre.

```
{}
```

```
let miCallback = () => console.log('Hola mundo!');  
setTimeout(miCallback, 1000);
```



Al escribir una función como parámetro lo hacemos **sin los paréntesis** para evitar que se ejecute. Será la función que la recibe quien se encargue de ejecutarla.

## {código}

```
function nombreCompleto(nombre, apellido) {  
    return nombre + ' ' + apellido;  
};
```

```
function saludar(nombre, apellido, callback) {  
    return '¡Hola ' + callback(nombre,  
apellido) + '!';  
};
```

```
saludar('Juanito', 'Sánchez', nombreCompleto);
```

## {código}

```
function nombreCompleto(nombre, apellido) {  
    return nombre + ' ' + apellido;  
};
```

```
function saludar(nombre, apellido, callback) {  
    return '¡Hola ' + callback(nombre,  
    apellido) + '!';  
};
```

```
saludar('Juanito', 'Sánchez', nombreCompleto);
```

Definimos la función **nombreCompleto()**.

Esta se encarga de unir con un espacio un nombre y un apellido.

Nos **devuelve** un **string**.

## {código}

```
function nombreCompleto(nombre, apellido) {  
    return nombre + ' ' + apellido;  
};
```

```
function saludar(nombre, apellido, callback) {  
    return '¡Hola ' + callback(nombre,  
    apellido) + '!' ;  
};
```

```
saludar('Juanito', 'Sánchez', nombreCompleto);
```

Definimos la función **saludar()**.

Esta recibe un nombre, un apellido y un **callback** como parámetros.

Este último será la función que vamos a querer ejecutar internamente.

## {código}

```
function nombreCompleto(nombre, apellido) {  
    return nombre + ' ' + apellido;  
};
```

```
function saludar(nombre, apellido, callback) {  
    return '¡Hola ' + callback(nombre,  
    apellido) + '!'  
};
```

```
saludar('Juanito', 'Sánchez', nombreCompleto);
```

Lo que queremos devolver es un string completo.

La primera parte la tenemos en el return: '¡Hola (...) !.

El resto (...) vendrá de lo que nos retorne el **callback** en el momento en el que se ejecute.



## {código}

```
function nombreCompleto(nombre, apellido) {  
    return nombre + ' ' + apellido;  
};  
  
function saludar(nombre, apellido, callback) {  
    return '¡Hola ' + callback(nombre,  
    apellido) + '!'  
};
```

```
saludar('Juanito', 'Sánchez', nombreCompleto);
```

Ejecutamos la función **saludar**, le pasamos como parámetros un nombre, un apellido y la función **nombreCompleto()**.

**Primero** se ejecutará el **callback**, que va a devolver el nombre completo y **luego** se ejecutará la función **saludar()**, que va a devolver el saludo completo.



La función **saludar()** ¿solo funciona si le pasamos como **callback** la función **nombre()**?

**¡No!**

Podemos pasarle cualquier otra función que devuelva un **string**, ya que en la estructura interna de **saludar()** definimos que opere con ese tipo de dato.



## {código}

```
function iniciales(nombre, apellido) {  
    return nombre[0] + apellido[0];  
};
```

Podríamos definir otra función que se encargue de devolvernos las iniciales del nombre y el apellido que nos pasen.

```
function saludar(nombre, apellido, callback) {  
    return '¡Hola ' + callback(nombre,  
    apellido) + '!';  
};
```

```
saludar('Juanito', 'Sánchez', iniciales);
```

## {código}

```
function iniciales(nombre, apellido) {  
    return nombre[0] + apellido[0];  
};
```

```
function saludar(nombre, apellido, callback) {  
    return '¡Hola ' + callback(nombre,  
    apellido) + '!';  
};
```

```
saludar('Juanito', 'Sánchez', iniciales);
```

Esta vez cuando ejecutamos la función **saludar()**, le pasamos la función **iniciales()** como callback.

Nuevamente se ejecutará el callback, esta vez va a devolver las iniciales y luego se ejecutará la función **saludar()**, que va a devolver el saludo completo.

**DigitalHouse** >  
Coding School