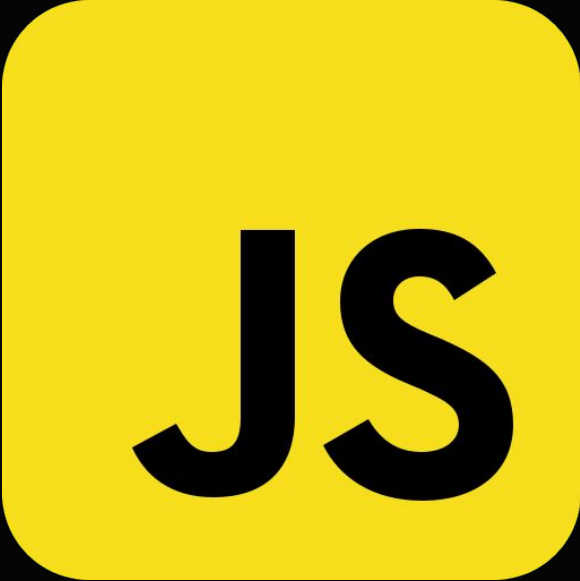




05 Callbacks

Yael Ruiz Pucheta

- Definición de Callbacks
- ¿Por qué son importantes?
- ¿Para que usan?
- Cómo pasar funciones a otras funciones
- Callback Hell



JS

Callbacks

Definición

Una función callback es aquella que es enviada como argumento a otra función para “llamarse de nuevo” en un momento posterior o en el cual sea necesaria.

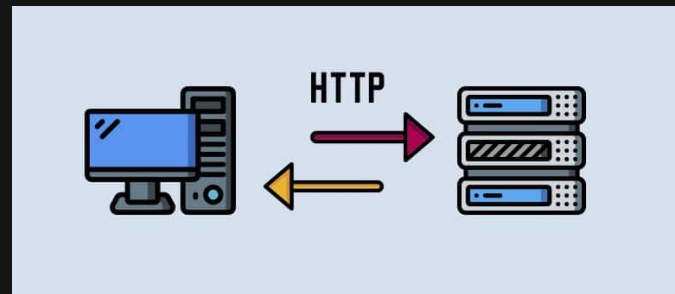
```
const primeraAccion = () = {  
  ...  
}  
  
const segundaAccion = (valor,callback) = {  
  ...  
}  
  
segundaAccion(1,primeraAccion)
```

Callbacks

¿Por qué son importantes?

Los callbacks tienen relevancia en Javascript ya que permiten “avisar” cuando una función termina de hacer algo.


Funciona bastante bien cuando se realizan procesos que **no se pueden resolver de forma secuencial** o se desconoce **el tiempo de espera** de resolución como solicitar información a un recurso externo.



Callbacks

Ejemplo

Hacer un contador de paso del tiempo



```
let contador = 1;

setInterval(() => {
  contador++;
  console.log(contador);
}, 1000);
```

Callbacks

Ejemplo

Hacer un contador de paso del tiempo (Variante)

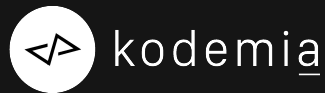


```
let contador = 1;

const aumentarContador = () => {
  contador = contador + 1;
  console.log(contador);
}

setInterval(aumentarContador, 1000);
```

Callbacks



Ejemplo

Simulación de petición a un servidor

```
const petitionServidor = (callback) => {  
  setTimeout(() => {  
    const calificaciones = [1, 2, 3, 4, 5];  
    callback(calificaciones)  
  }, 3000);  
}  
  
const mostrarCalificaciones = (calificaciones) => {  
  console.log(calificaciones);  
}  
  
petitionServidor(mostrarCalificaciones);
```

Callbacks

Ejemplo

Simulación de petición a un servidor (Variante)

```
const peticionServidor = (callback) => {  
  setTimeout(() => {  
    const calificaciones = [1, 2, 3, 4, 5];  
    callback(calificaciones)  
  }, 3000);  
}  
  
peticionServidor((calificaciones) => {  
  console.log(calificaciones);  
});
```


Callbacks

Ejercicio

Crear una función `find` que acepte un array y un callback y que:

- por cada elemento del array ejecute el callback pasándole dicho elemento como argumento
- devuelva el elemento pasado como argumento del primer callback que devuelva `true`
- si ningún callback devuelve `true`, devuelva `undefined`

Callbacks

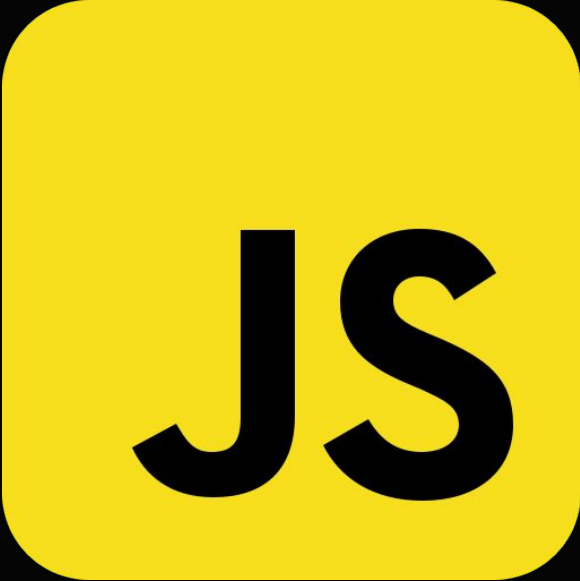
Callback Hell

Es el término dado a la **sobre explotación de la anidación de funciones**, generalmente esto es propiciado por malas prácticas de codificación.

Esta situación es tan conocida en el mundo de la programación que tiene su propia página web, callbackhell.com para conocer más sobre este fenómeno.



- Definición de Callbacks
- ¿Por qué son importantes?
- ¿Para que usan?
- Cómo pasar funciones a otras funciones
- Callback Hell



JS



05 Callbacks

Yael Ruiz Pucheta