

Función Callback

Una función callback es una función que se pasa como parámetro a otra función. Esta función callback será llamada hasta que se complete alguna tarea.

```
const imprimeMensaje = function() { console.log('Hello World'); } //Asignamos una función a una constante

setTimeout(imprimeMensaje,5000); //Se la pasamos a la función SetTimeout()
                                //que la ejecutará hasta que pasen 5 segundos
```

También podemos pasar una función callback como función anónima o como función flecha:

```
setTimeout(function() { console.log('Hello World'); },5000); //callback pasado como función anónima a SetTimeout()

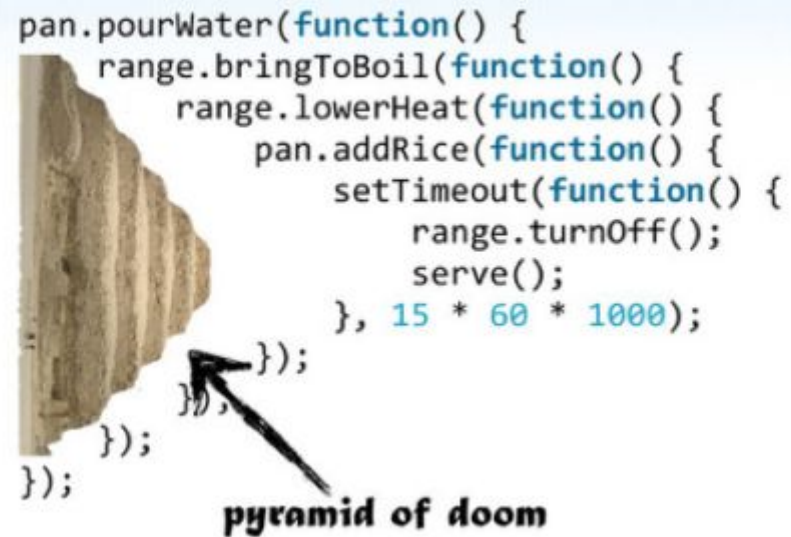
setTimeout( ()=>{ console.log('Hello World'); },5000);        //callback pasado como función flecha a SetTimeout()
```

Función callback en una declaración de eventos:

```
document.getElementById("#boton1").addEventListener("click", function() {console.log("Click en el boton 1");});
```

Callback Hell

El problema de los callbacks es que cuando una función depende del resultado de otra y empezamos a usar callbacks, podríamos terminar con lo que se conoce como Callback Hell:



```
pan.pourWater(function() {  
  range.bringToBoil(function() {  
    range.lowerHeat(function() {  
      pan.addRice(function() {  
        setTimeout(function() {  
          range.turnOff();  
          serve();  
        }, 15 * 60 * 1000);  
      });  
    });  
  });  
});
```

pyramid of doom

mozilla