



Callback & Callback Hell

What is a callback in JavaScript?

Callback is any function that is passed as an argument to another function, and then invoked from within that function.

Swipe for example



```
function greeting(name) {  
    alert('Hello ' + name);  
}
```

```
function processUserInput(callback) {  
    var name = prompt('Please enter your name.');
```

callback(name);

```
}
```

```
//Greeting IS A CALLBACK FUNCTION here which is  
//PASSED TO processUserInput FUNCTION  
//and then called from there  
processUserInput(greeting);
```

What is a callback hell in JavaScript?

When every new function invocation is placed inside previous callback and imagine we have to invoke 10 such callbacks one after other, such code is hard to read and maintain. This is called callback hell.

Swipe for example



```
function calculateSquare(number, callback)
{
  setTimeout(()=>{
    const result = number*number;
    callback(result);
  },1000)
}
```



Function responsible for calculating squares of the number passed

```
calculateSqaure(1, (result) => {
  console.log(result);
  calculateSquare(2,(result)=>{
    console.log(result);
    calculateSquare(3,(result)=>{
      console.log(result);
    })
  })
})
```

Here we are invoking calculateSquare function three times one after other

New invocation is placed inside previous callback, making the code complex, giving birth to

Callback hell

//Outputs 1,4 and 9 each after 1 second

So in order to keep code easy to read and maintainable, we need to avoid this callback hell situation. In order to avoid it, we use

PROMISES IN JS

Stay tuned for promises...



**Follow Rohit Sharma for more
such updates**