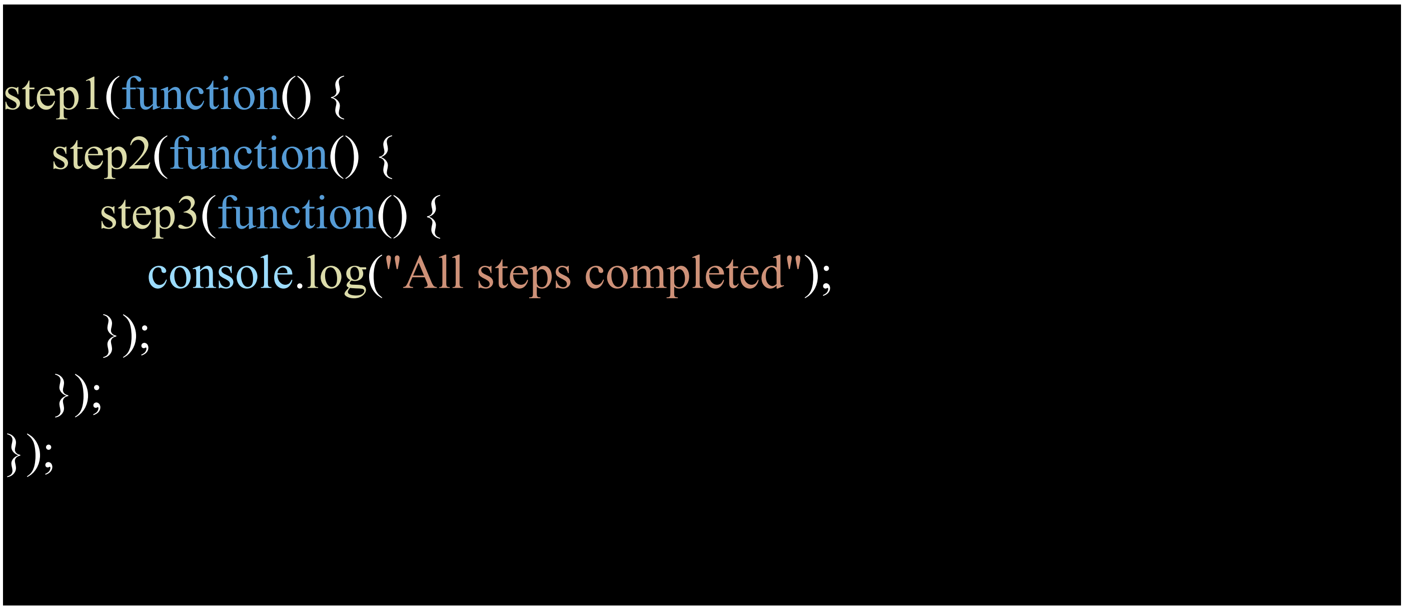
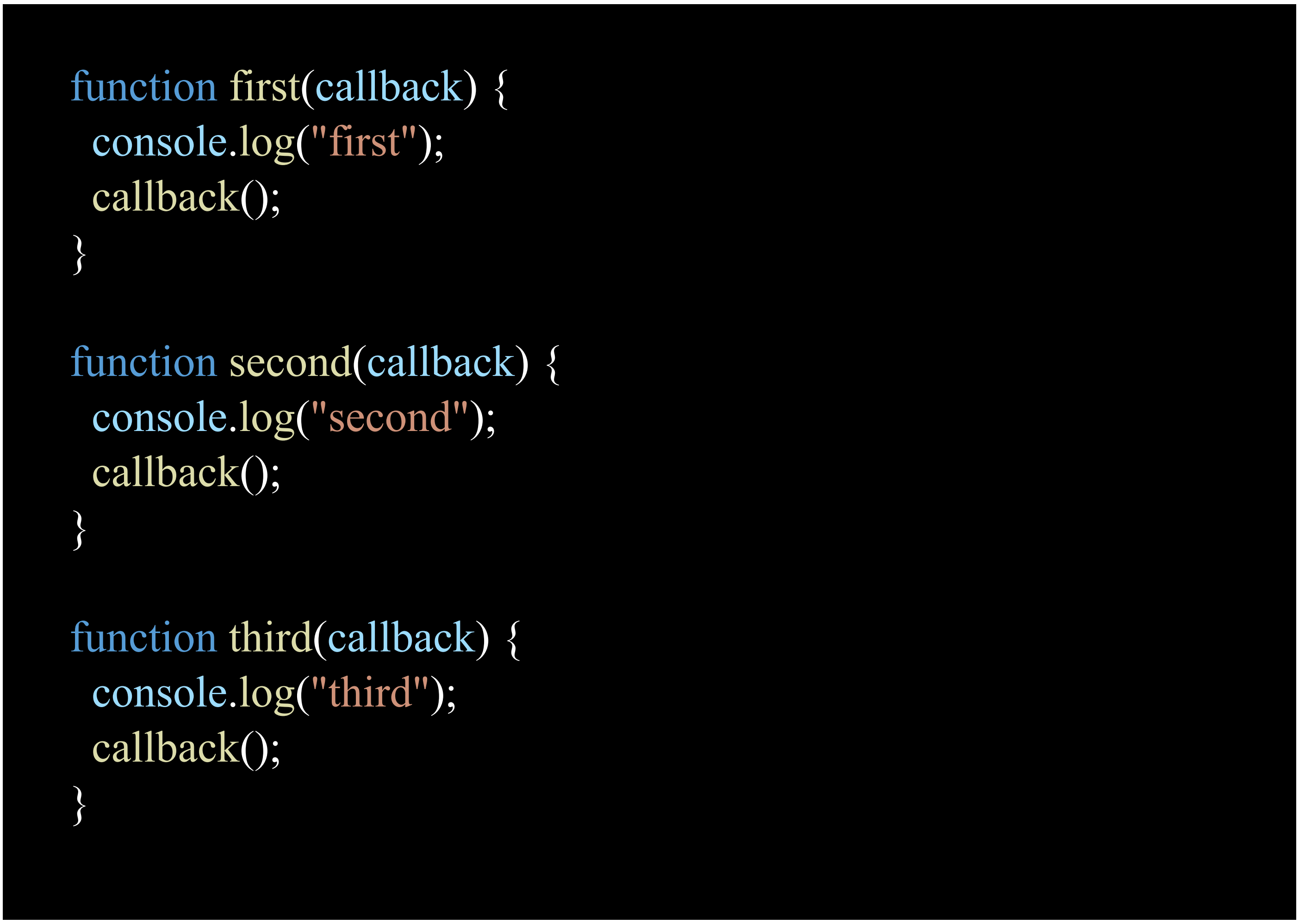
Callback hell

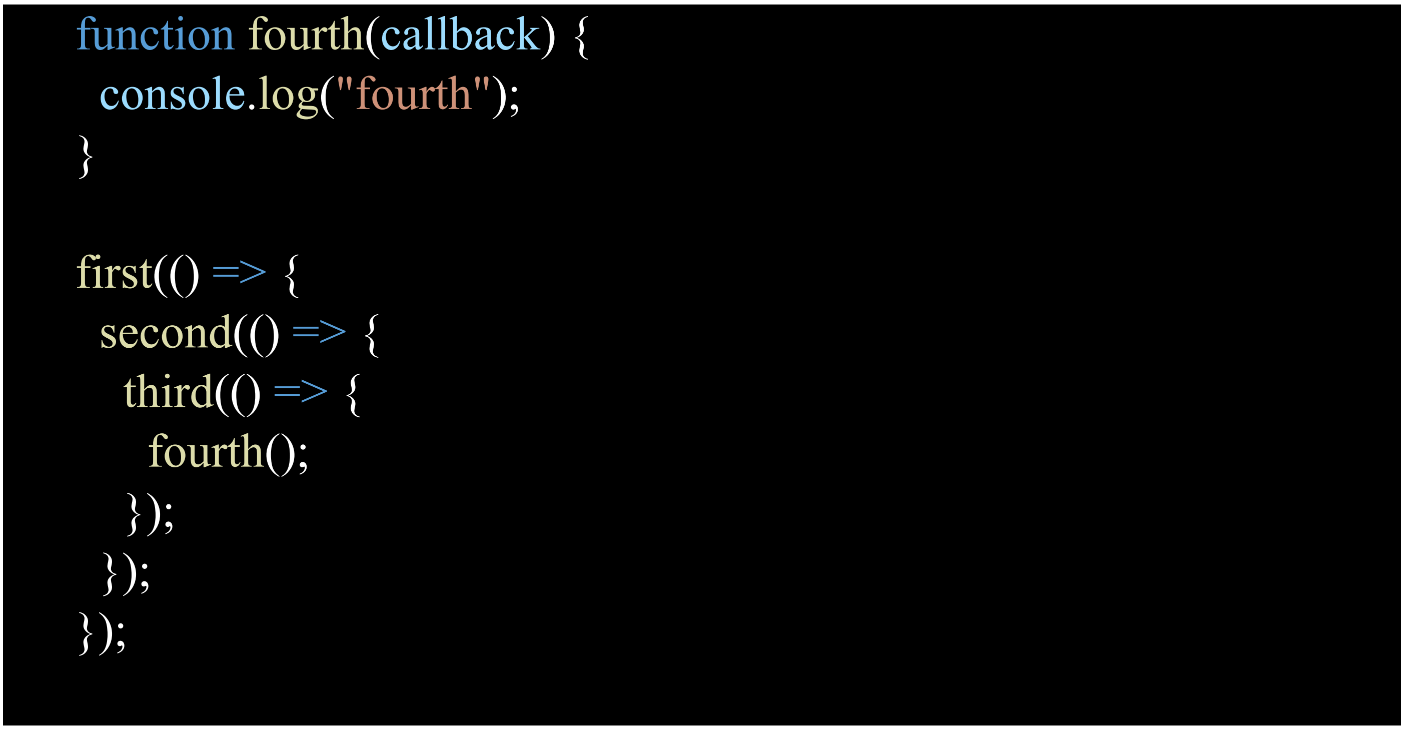
Callback hell, also known as "Pyramid of Doom," is a term used in JavaScript programming to describe a situation where multiple nested callbacks make the code difficult to read, understand, and maintain. This usually happens when dealing with asynchronous operations, such as making API requests or reading files.



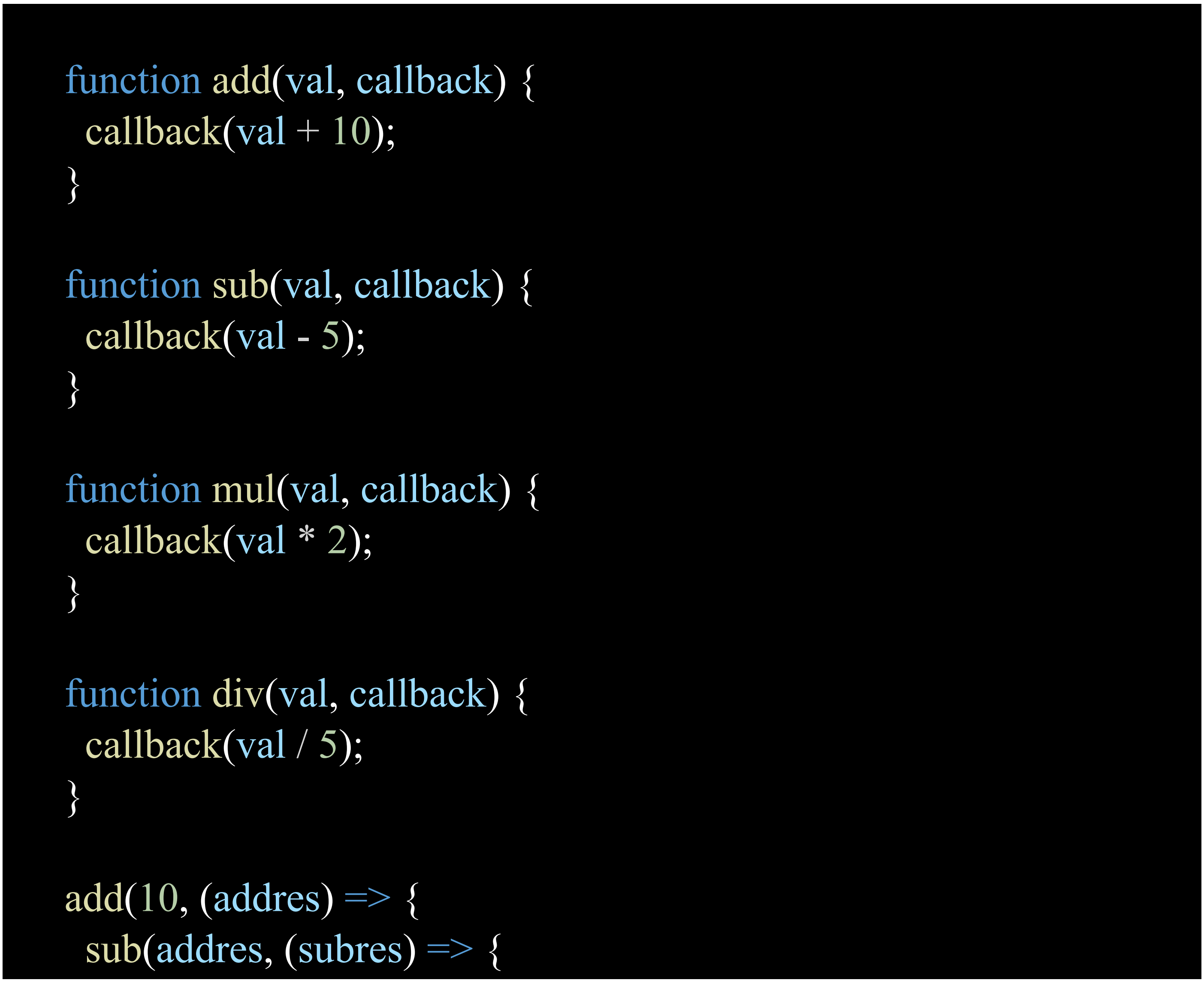
```
step1(function() {
    step2(function() {
        step3(function() {
            console.log("All steps completed");
        });
    });
});
```



```
function first(callback) {
  console.log("first");
  callback();
}

function second(callback) {
  console.log("second");
  callback();
}

function third(callback) {
  console.log("third");
  callback();
}
```



```
function fourth(callback) {
  console.log("fourth");
}

first(() => {
  second(() => {
    third(() => {
      fourth();
    });
  });
});
```



```
function add(val, callback) {
  callback(val + 10);
}

function sub(val, callback) {
  callback(val - 5);
}

function mul(val, callback) {
  callback(val * 2);
}

function div(val, callback) {
  callback(val / 5);
}

add(10, (addres) => {
  sub(addres, (subres) => {
      mul(subres, (mulres) => {
        div(mulres, (finalres) => {
          console.log(finalres);
        });
      });
    });
  });
```

To mitigate callback hell, several approaches have been developed:

1) Named functions
2) Promises
3) Async/await