

CALLBACKS

- Callbacks is simply a function passed as an argument to another function which will then use it (call it back).
- A callback function is called at a completion of a given task.
- Callbacks are called when we don't know when something will be done.

```
var fs = require('fs');  
  
fs.readFile("./sample.txt.txt", function(error, file) {  
  if (error) {  
    console.log('Error: ', error);  
  }  
  else {  
    console.log(file.toString());  
  }  
});  
  
console.log('Program Ended');
```

Callback function takes 2 arguments error and file. By convention, first argument of a callback function is an error.

CALLBACK HELL

Callback hell is a phenomenon where a javascript developer tries to execute multiple asynchronous operations one after the other.

```
const createUser = function(username, password, picture, callback) {
  dataBase.createUser(username, password, (error, userID) => {
    if (error) {
      callback(error)
    } else {
      cloudinary.uploadPicture(picture, (error, path) => {
        if (error) {
          callback(error)
        } else {
          dataBase.updatePicture(userID, path, (error) => {
            if (error) {
              callback(error);
            } else {
              callback(null);
            }
          })
        }
      })
    }
  })
}
};
```