

Asynchronous Functions



Dan Geabunea

Passionate Software Developer

@romaniancoder | www.dangeabunea.com

Overview

Brief introduction to promises

Async/await pattern

Course recap



Pre-requisites

Basic understanding of asynchronous programming in JavaScript

Basic understanding of Promises





Deep Dive: **JavaScript Promises and Async** **Programming**

Nate Taylor

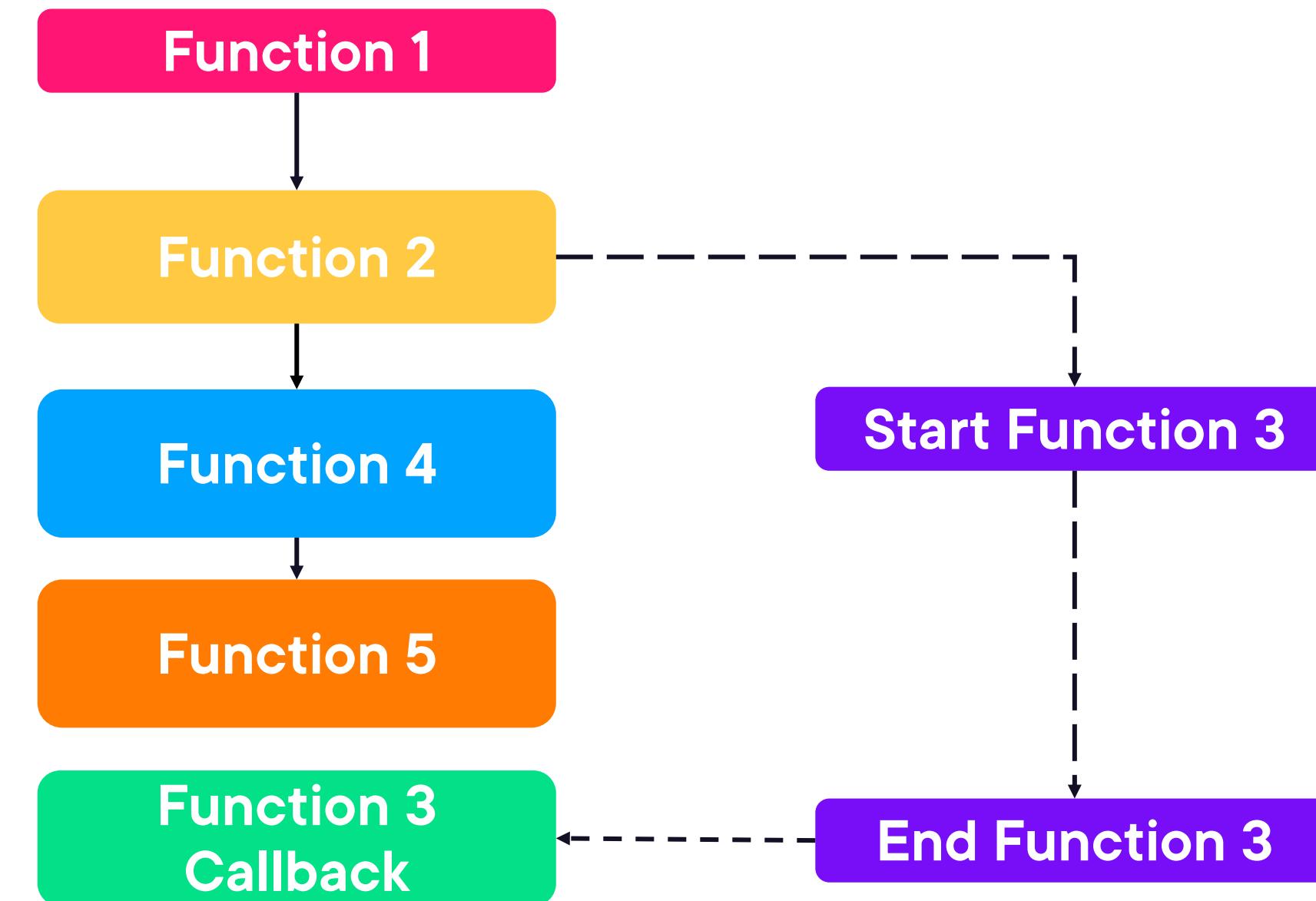


Asynchronous Programming

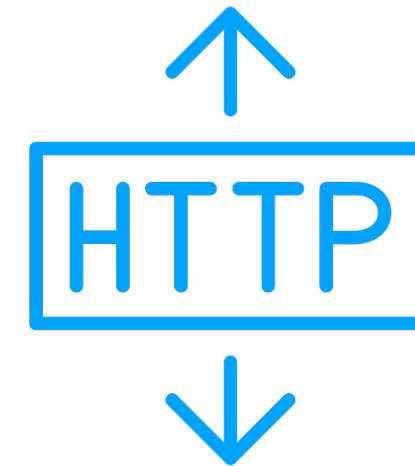
The ability of your program to start a potentially long-running task and still be responsive to other events while waiting for that task to complete



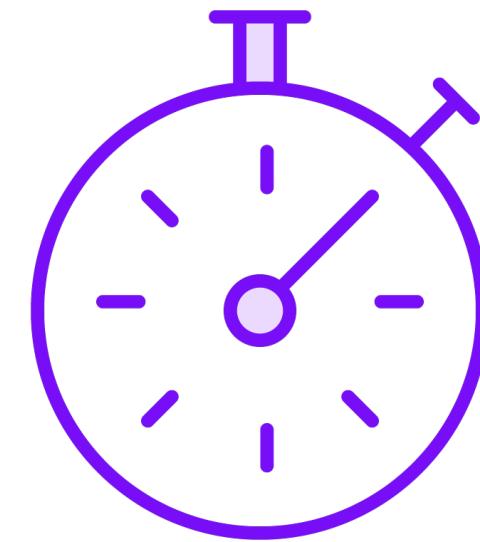
Asynchronous Code in JS



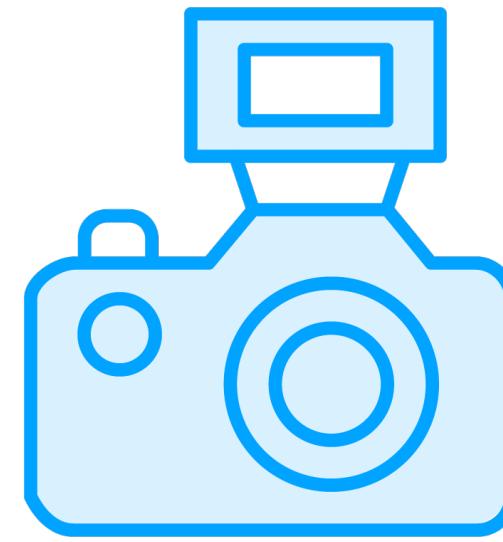
Actions That Might Take a Long Time



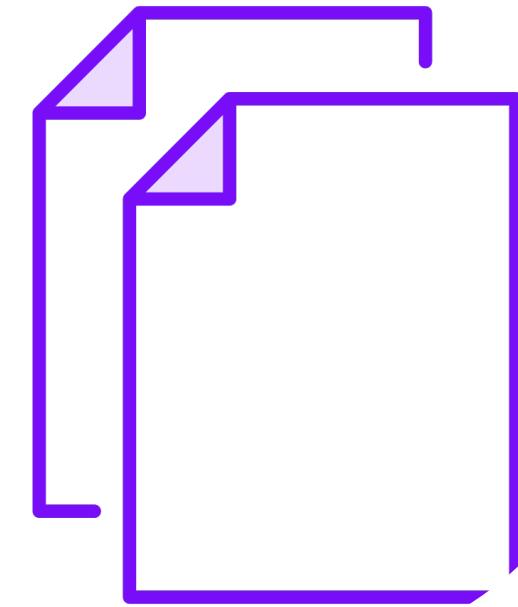
HTTP Calls



Time delays



**Camera /
microphone**



Selecting files



Promise

An object returned by an asynchronous function that represents its current state of execution



Promise

```
function printMediaDevices() {  
  navigator.mediaDevices  
    .enumerateDevices() // returns a Promise  
    .then((devices) => console.log(devices))  
    .catch((err) => console.log(err));  
}
```



Fetching Data Using Promises



Async / Await Pattern



Async / Await Pattern

An easier way to work with asynchronous, promise-based code



```
async function someFunction() {  
  return 'I am an async function';  
}
```

```
async function someFunction() {  
  return Promise.resolve('I am an async function');  
}
```

Async Functions

Always return a Promise

The value of the string is wrapped in a resolved promise automatically



```
async function someFunction() {  
  return 'I am an async function';  
}  
  
someFunction().then((res) => console.log(res)); // 'I am an async function'  
  
let value = await someFunction(); // Promise is resolved to its value  
  
console.log(value);
```

Calling Async Functions

Use the Promise object

Use await - it makes JavaScript wait until the promise settles and returns its result



**“await” suspends the
function execution until the
promise settles => JS can
run other jobs in the
meantime**



Refactoring to `async/await`



Summary

With asynchronous programming, JS can execute a long-running task and continue to run other tasks in parallel

In JavaScript, asynchronous operations are implemented with Promises

To simplify code, we can replace traditional Promise handling with the `async / await` pattern

An `async` function is a function that will always return a Promise

Inside an `async` function, we can use `await` to capture the result of Promise operations

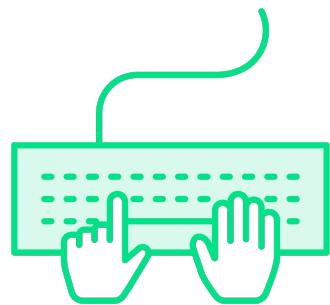


Course Recap

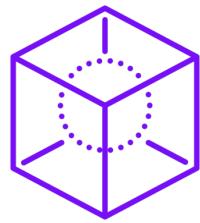


Congrats!

Deep Dive into Functions



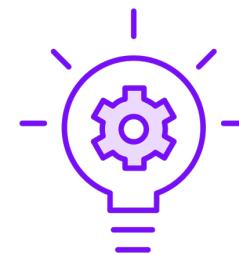
**Define and invoke
functions**



**Understand scope and
closure**



Pass data to functions



Demystify “this”



**Use methods, getters &
setters**

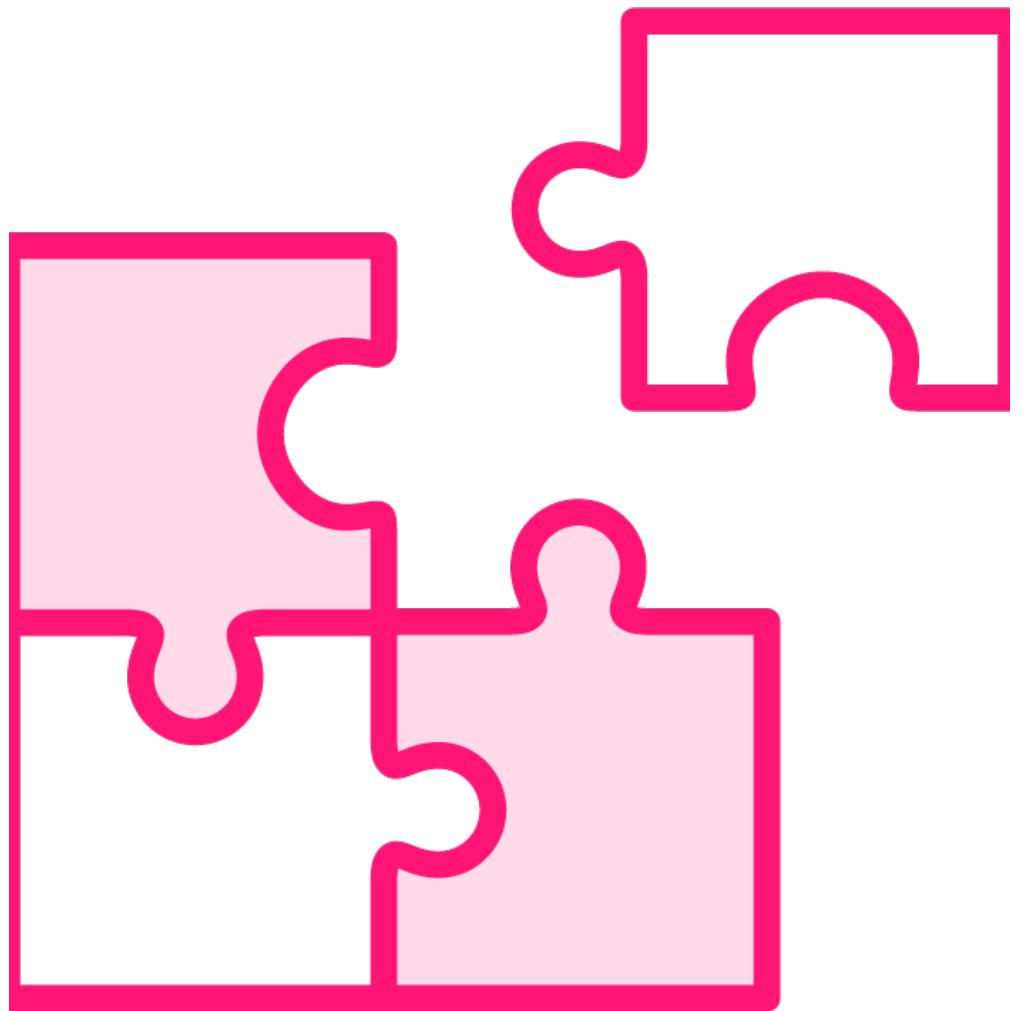


**Use asynchronous
functions**



You are now ready to
leverage the full power of
functions to create more
robust JS apps





**Functions are just one step on your road to
JavaScript mastery**





Dan Geabunea

@romaniancoder

www.dangeabunea.com

