Synchronous VS Asynchronous Programming

Example Code:

```
const name = 'Miriam';
const greeting = `Hello, my name is ${name}!`;
console.log(greeting);
// "Hello, my name is Miriam!"
```

This code does the following:

1. Declares a string called name.
2. Declares another string called greeting, which uses name variable.
3. Outputs the greeting to the JavaScript console.

In this example the browser waits for each line to finish before proceeding to run the next.
This is an Example of **Synchronous Programming**.

Cons of long running synchronous functions:

While one function is running our program is completely unresponsive

With **Asynchronous Functions** you are able to start a long running operation by calling a function. You can have that function start an operation and return immediately. Now our program can still be responsive to other events. We can also have it notify us when it eventually completes.

Async Functions are similar to **Event Handlers**. In actuality Event handlers are a form of asynchronous programming.

CALLBACKS
A Callback is a function that's passed into another function with the expectation that the callback will be called at a certain time. Callbacks used to be the main way async functions were implemented in JS.

For the sake of readability, modern APIs use the Promise for async programming.