Code Talks

Week 2 of Node.js

Instructions

Please review the questions/problems below. Please come prepared to explain the concept, or code out the question in front of the class.

Coding Problem 1

What is a promise? What are promises used for? What are pro/cons to using a promise? Please code some examples of promises to show to the class. Make sure to go into detail as you explain your code to the class.

A promise object is an object that is fulfilled after being processed. The asynchronous capabilities allows it to be defined and used in non chronological order processing. It can be in 3 States:

- Pending
- Fulfilled
- Rejected

```
// EXAMPLE ON GITHUB REPO
```

Coding Problem 2

What is a callback? What is callback hell? Why does callback hell occur and why should we avoid callback hell? Please code an example of callback hell to show to the class. Make sure to go into detail as you explain your code to the class.

```
//GIT HUB REPO
```

Coding Problem 3

The “this” keyword works differently in arrow functions vs regularly declared functions. Explain how the “this” keyword works differently in arrow functions vs normal functions. Write some code that shows that you understand the difference between how the “this” keyword is used. Make sure to go into detail as you explain your code to the class.

Coding Problem 4

Using only the **filter** method on the numbers array, create a function that returns a new array that only contains the numbers divisible by both 3 and 6 but NOT 9.

const numbers = [48, 20, 18, 7, 12,9, 8,10,12, 96, 3,36,51];

```
// EXAMPLE ON GITHUB REPO
```

Coding Problem 5

What is a async/await? Why is async/await useful? What are pro/cons to using async/await? Please code some examples of async/await to show to the class. Make sure to go into detail as you explain your code to the class.

async/await is a javascript syntax that allows you to declare a asynchronous function that. The await keyword pauses the execution of the function until it has been resolved or rejected. This makes it more akin to synchronous code and more readable, but some drawbacks include non compatibility with older environments predating 2017 and also more overhead tasks leaning causes longer wait times.

```
// EXAMPLE ON GITHUB REPO
```