

Full Stack Development II – Lab 10

- ES6 Promises & Error Exercises

Developer Note:

- You may use the JS playground for development
<https://repl.it/languages/nodejs>
- Create separate files for each exercise

Exercise 1:

- Write a function **appendMyArray** that takes an array and value as parameters
- The function will have the following requirements
 - Use the array push method to append the value to the array
 - Implement a try..catch block to capture any errors and log the error message in console
 - Implement a finally statement that will output the final value of the array

```
appendMyArray([53, 54], 55)
appendMyArray('xyz', 100)
appendMyArray({'make': 'Tesla', model: 'S'}, 5)
```

The following output is expected:

```
exiting..myArray:53,54,55
myArray.push is not a function
exiting..myArray:xyz
myArray.push is not a function
exiting..myArray:[object Object]
```

Exercise 2:

- Write a function **tryCatchFinally** that takes no parameters
- The function will have the following requirements
 - Implement a try...catch block to capture any errors and log the error message in console
 - Create an error object with the message **“throwing an error”** and use the **throw** keyword to throw the message within the body of the try..catch block
 - Implement a finally statement that output the message **“runs finally.”**

Invoke the function:

```
tryCatchFinally();
```

The following output is expected:

```
Throwing an error
Runs finally
```

Exercise 3:

- Write a function **compareNumToTen** that takes a number as an argument and returns a **Promise** that tests and rejects if the value is less than or resolves if greater than the value 10.

Promise Reference

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

```
compareNumToTen(15)
  .then(result => console.log(result))
  .catch(error => console.log(error))
```

Output

```
15 is greater than 10, success!
8 is less than 10, error!
```

Exercise 4:

- Write two functions that use Promises that you can chain! The first function, **makeUpperCase()**, will take in an array of words and capitalize them, and then the second function, **sortWords()**, will sort the words in alphabetical order. If the array contains anything but strings, it should throw an error.

Promise Chaining Reference

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise

```
const arrayOfNames = ['jaxx', 'tiny', 'clay']
const mixedArray = ['anarchy', 99, true]
```

```
makeUpperCase(arrayOfNames)
  .then(sortWords)
  .then((result) => console.log(result))
  .catch(error => console.log(error))

makeUpperCase(mixedArray)
  .then(sortWords)
  .then((result) => console.log(result))
  .catch(error => console.log(error))
```

Output

```
[ 'CLAY', 'JAXX', 'TINY' ]  
Error: Not all items in the array are strings!
```

Exercise 5:

- Use the **Promise.all** syntax to resolve all three promises below and return a new promise. Call the function and use the thenable `.then()` keyword to output the result.

Promise.all() Reference

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/all

```
var promise1 = 101  
var promise2 = "follow the white"  
var promise3 = new Promise(function(resolve, reject) {  
  |   resolve("rabbit.. neo");  
});
```

Output

```
🔍 [ 101, 'follow the white', 'rabbit.. neo' ]
```