Practice tasks for Callbacks:

1. Write a function that returns true or false for a given number.
2. Write a function that returns the given string with a concatenated exclamation ! at the end.
3. Write a function that duplicates each char in a string. If I pass 'abc' to the function, it should return 'aabbcc'.
4. Write a function called atLeastTwo that receives an array and a callback as its arguments. If at least two elements from the array return a truthy value when passed as an argument to the callback, atLeastTwo should return true. If there aren't at least two elements from the array that return a truthy value when passed as arguments to the callback, atLeastTwo should return false.
5. Write a groupBy function that groups elements from an array by the returned value from the callback when an element from the array is passed as an argument.

Practice tasks for Promises:-

1. Using setTimeout, print the string 'Hello!' after 1000ms.
2. Create a promise. Have it resolve with a value of 'Resolved!' in resolve after a delay of 1000ms, using setTimeout. Print the contents of the promise after it has been resolved by passing console.log to .then
3. Create another promise. Now have it reject with a value of "Rejected!" without using setTimeout. Print the contents of the promise after it has been rejected by passing console.log to .catch
4. Promises are asynchronous and we're now going to prove that they indeed are! Create a promise and have it resolve with the value of "Promise has been resolved!" Then uncomment the code at bottom of Challenge 4. What order do we expect "Promise has been resolved!" and "I'm not the promise!" to print? Why?

5. Write a function delay that returns a promise. And that promise should return a setTimeout that calls resolve after 1000ms
6. This challenge we'll chain promises together using ".then" Create two variables: firstPromise and secondPromise Set secondPromise to be a promise that resolves to "Second!" Set firstPromise to be a promise that resolves to secondPromise Call the firstPromise with a ".then", which will return the secondPromise> promise. Then print the contents of the promise after it has been resolved by passing console.log to .then
7. Using Promise.all, print the message once all the promises are resolved

1. What would this code revert?

```
function job() {
    return new Promise(function(resolve, reject) {
        reject();
    });
}

let promise = job();

promise

.then(function() {
    console.log('Success 1');
})

.then(function() {
    console.log('Success 2');
})

.then(function() {
    console.log('Success 3');
})

.catch(function() {
    console.log('Error 1');
})
```

```
.then(function() {
    console.log('Success 4');
});
```