1. Create a function called runPromise()
2. Inside the function, declare a variable called aPromise that is a new Promise whose constructor has an arrow function that:
   1. Takes resolve and reject as arguments

let aPromise = new Promise((resolve, reject) => {  
  
}

* 1. Has a function body that:
     1. Declares a variable called delayedFunc that is set as follows:

…

let delayedFunc = setTimeout(() => {

//whether it resolves or rejects is unknown

let randomNumber = Math.random();

(randomNumber < 0.5) ? resolve(randomNumber) :   
 reject(randomNumber);

}, Math.random() \* 5000); //function will return sometime: 0-5s

…

The fact that we have used setTimeout here and the final argument Math.random() \* 5000 (which generates a random number between 0 and 1 and multiplies it by 5000) means that the arrow function will execute somewhere between 0ms and 5000ms. The arrow function itself generates a random number between 0 and 1 and the Promise is resolved if the number is less than 0.5 and rejects otherwise.

1. Call aPromise with a .then chain and set data to be the resolved value and log this out.
2. Add a catch block and set error to be the rejected value and log this out.
3. Save the file and refer to the console of your browser - don't forget that after each refresh you will need to wait up to 5 seconds for the result to show. Refresh the browser several times to satisfy that the Promise resolves and rejects randomly.