

Promises

GIRLSSCRIPT EDUCATION OUTREACH PROGRAM

HELLO!

I am Shubham

You can find me at @shugo111

1.

What is promises?

promises

- ◆ **A promise is an object which has then and catch methods on it. One of this method gets called when the promise returns a value or an error. Let's understand how to create a promise.**
- ◆ **A promise object is created from Promise constructor/class which needs a callback function AKA executor function (either in ES5 syntax or a fat arrow function). This callback function receives the resolve and reject function arguments, either of which we must envoke with an optional payload**

Promises(then and catch)

```
var myPromise = new Promise( ( resolve, reject ) => {  
  resolve( 'successPayload' );  
  // reject( 'errorPayload' );  
} );  
myPromise  
  .then( successCallback )  
  .catch( errorCallback )  
  .finally( finallyCallback );
```

How Do Promises work?

- The biggest misconception about Promises in JavaScript is that they are asynchronous. Well, not everything of Promises is asynchronous. Let's have a look at below example which explains a lot of things.

Promise.all()

`Promise.all([promises])` method takes an array of promises and returns a new promise. This promise will be fulfilled when and if all promises are fulfilled

If any of the promises is rejected early, promise returned by this methods will be rejected and the results of other promises will.

Async/Await

Async/Await is a fancier syntax to handle multiple promises in synchronous code fashion.

When we put `async` keyword before a function declaration, it will return a promise and we can use `await` keyword inside it which blocks the code until promise it awaits resolves or rejects.

```
async function myFunction() {  
    var result = await new MyPromise();  
    console.log( result );  
}  
myFunction(); // returns a promise
```


THANKS!

Any questions?

You can find me at @shugo111

