JavaScript Day 3

**Promises**

* **Are Objects that helps us to work with asynchronous tasks.**

JavaScript uses callbacks, [among other things](http://exploringjs.com/es6/ch_async.html#sec_receiving-results-asynchronously). However, callbacks suffer from problems like [Callback Hell](http://callbackhell.com/" \t "_blank)/[Pyramid of Doom](https://en.wikipedia.org/wiki/Pyramid_of_doom_(programming)).

**The Promise object represents the eventual completion (or failure) of an asynchronous operation, and its resulting value.**

Example for Promises:

For example, you book a flight ticket to go to India for travelling to the beautiful hill station [Darjeeling](https://en.wikipedia.org/wiki/Darjeeling). After booking, you get a **ticket**. That ticket is a **promise** by the airline that you will get a seat on the day of your departure. In essence, the ticket is a placeholder for a future value, namely, the seat.

# Creating and Resolving Promises

let promise=new Promise(function(resolve, reject){

setTimeout(function(){

resolve('Done');

},1500);// 1 and half seconds

});

promise.**then(function(value**){ //success

console.log(value);

});

# Rejecting Promise

let promise=new Promise(function(resolve, reject){

setTimeout(function(){

reject('Failed');

},1500);

});

promise.then(function(value){

console.log(value);

}, function(error){

console.log(error);

});

# Chaining Promises and catching Errors

function waitASeconds(seconds){

return new Promise(function(resolve,reject){

if(seconds>2){

reject('Rejected');

} else{

setTimeout(function(){

seconds++;

resolve(seconds);

},1000);

}

});

}

waitASeconds(3)

.then(waitASeconds)

.then(function(seconds){

console.log(seconds);

})

.catch(function(error){

console.log(error);

});

Referenec Links

<https://exploringjs.com/es6/ch_promises.html#sec_first-example-promises>