**PROMISES ASSIGNMENT**

1).

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');

});

What is the output of the code above ?

Output:

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node abc.js

Error 1

Success 4

===========================================================

2). Write a sleep function using a promise in javascript?

const sleep = (milliseconds) => {

    return new Promise(resolve => setTimeout(resolve, milliseconds))

  }

  /\*Use like so\*/

  async function timeSensativeAction(){ //must be async func

    console.log("do something here");

    await sleep(5000) //wait 5 seconds

    console.log("do nothing here");

}

timeSensativeAction()

Output:

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node abc.js

do something here //(pause for 5 sec)

do nothing here

===========================================================

3). What is the output of the following code?

const promise = new Promise(res => res(2)); promise.then(v => {

console.log(v);

return v \* 2;

})

.then(v => {

console.log(v);

return v \* 2;

})

.finally(v => {

console.log(v);

return v \* 2;

})

.then(v => {

console.log(v);

});

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node tryyy.js

2

4

undefined

8

===========================================================

4).

console.log('start')

const fn = () => (new Promise((resolve, reject) => { console.log(1);

resolve('success')

}))

console.log('middle')

fn().then(res => {

console.log(res)

})

console.log('end')

What is the output of this code snippet?

OUTPUT:

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node tryyy.js

start

middle

1

end

success

===========================================================

5). Write a function delay that returns a promise. And that promise should return a setTimeout that calls resolve after 1000ms.

function delay(ms) {

    return new Promise(resolve => setTimeout(resolve, ms));

  }

delay(1000).then(() => console.log('runs after 1000 miliseconds'));

Output:

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node abc.js

runs after 1000 miliseconds

===========================================================

6).

Promise.resolve('Success!')

.then(data => {

return data.toUpperCase()

})

.then(data => {

console.log(data)

})

What will the output be?

Output:

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node tryyy.js

SUCCESS!

==========================================================

7).

var p = new Promise((resolve, reject) => { reject(Error('The Fails!'))

})

.catch(error => console.log(error))

.then(error => console.log(error))

What will the output be?

Output:

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node tryyy.js

Error: The Fails!

at C:\Users\ADMIN\Documents\c\JS\_CODE\tryyy.js:20:51

at new Promise (<anonymous>)

at Object.<anonymous> (C:\Users\ADMIN\Documents\c\JS\_CODE\tryyy.js:20:9)

at Module.\_compile (node:internal/modules/cjs/loader:1126:14)

at Object.Module.\_extensions..js (node:internal/modules/cjs/loader:1180:10)

at Module.load (node:internal/modules/cjs/loader:1004:32)

at Function.Module.\_load (node:internal/modules/cjs/loader:839:12)

at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run\_main:81:12)

at node:internal/main/run\_main\_module:17:47

undefined

===========================================================

8).

console.log('start')

setTimeout(() => {

console.log('setTimeout')

})

Promise.resolve().then(() => { console.log('resolve')

})

console.log('end')

What will the output be?

Output:

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node tryyy.js

start

end

resolve

setTimeout

===========================================================

10).

console.log('start')

Promise.resolve(1).then((res) => { console.log(res)

})

Promise.resolve(2).then((res) => { console.log(res)

})

console.log('end')

What will the output be?

Output:

PS C:\Users\ADMIN\Documents\c\JS\_CODE> node tryyy.js

start

end

1

2