

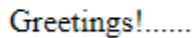
RollNo: 717823F137

Name: Nandhakumar P V

4. Promise, Promises chaining:

Task 16:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript</title>
</head>
<body>
  <script>
    let promise = new Promise((resolve,reject)=>{
      setTimeout(()=>resolve("Greetings!....."),1000)
    })
    promise.then(
      result => document.write(result),
      error => document.write(error)
    )
  </script>
</body>
</html>
```



Task 17:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>GitHub User Info</title>
</head>
<body>
  <script>
    fetch('https://api.github.com/users/iliakan')
      .then(response => {
        if (!response.ok) return Promise.reject('Failed to fetch data');
        return response.json();
      })
      .then(data => {
        const processedData = {
          username: data.login,
          name: data.name,
          publicRepos: data.public_repos
        };
        console.log(processedData);
      });
  </script>
</body>
</html>
```

```

    })
    .catch(error => console.error('Error:', error));
</script>
</body>
</html>

```

```

1
1

```

Task 18:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript</title>
</head>
<body>
  <script>
    randomnumber = (num)=>{
      return new Promise((response,reject)=>{
        if(num%2==0) response(true);
        else reject(false)
      })
    }
    let i1 = randomnumber(10)
    i1.then(
      result=>console.log(result),
      error=>console.log(error)
    )
  </script>
</body>
</html>

```

```

true

```

Task 19:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript</title>
</head>
<body>
  <script>
    let urls = [
      'https://api.github.com/users/iliakan',
      'https://api.github.com/users/remy',
      'https://api.github.com/users/jeresig'
    ];

```

```

let requests = urls.map(url => fetch(url));
Promise.all(requests)
  .then(responses => responses.forEach(
    response => console.log(`${response.url}: ${response.status}`)
  ));
</script>
</body>
</html>

```

```

https://api.github.com/users/iliakan: 200
https://api.github.com/users/remy: 200
https://api.github.com/users/jeresig: 200

```

Task 20:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Promise Chain Example</title>
</head>
<body>
  <script>
    function fetchData() {
      return new Promise((resolve) => {
        setTimeout(() => {
          const data = { id: 1, name: "John" };
          console.log('Fetched data:', data);
          resolve(data);
        }, 1000);
      });
    }
    function processData(data) {
      return new Promise((resolve) => {
        setTimeout(() => {
          data.name = data.name.toUpperCase();
          console.log('Processed data:', data);
          resolve(data);
        }, 1000);
      });
    }
    function logResult(data) {
      return new Promise((resolve) => {
        setTimeout(() => {
          console.log('Final result:', data);
          resolve('Process complete');
        }, 1000);
      });
    }
    fetchData()
      .then(data => processData(data))
      .then(processedData => logResult(processedData))

```

```
.then(result => console.log(result))
.catch(error => console.error('Error:', error));
</script>
</body>
</html>
```

```
> Fetched data: {id: 1, name: 'John'}
> Processed data: {id: 1, name: 'JOHN'}
> Final result: {id: 1, name: 'JOHN'}
  Process complete
```

5. Async/await:

Task 21:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Promise Chain Example</title>
</head>
<body>
  <script>
    async function fetchData() {
      try {
        const response = await fetch('https://api.github.com/users/iliakan');
        if (!response.ok) {
          throw new Error('Failed to fetch data');
        }
        const data = await response.json();
        console.log(data.name);
        return data;
      } catch (error) {
        console.error('Error:', error);
      }
    }
    fetchData();
  </script>
</body>
</html>
```

Ilya Kantor

Task 22:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<title>Promise Chain Example</title>
</head>
<body>
  <script>
    data = async()=>{
      try{
        const url = await fetch('https://api.github.com/users/iliakan')
        if(!url.ok)
          throw new Error("Can't able to fetch the data");
        console.log("Data Fetched...");
        const pro = await url.json()
        console.log("Fetched data: ",pro);
        const pdata = await pro.name.toUpperCase()
        return pdata;
      }catch(error){
        console.error(error)
      }
    }
    data().then((res)=>{
      console.log("Fetched Response",res);
    })
  </script>
</body>
</html>

```

```

Data Fetched...
> Fetched data: {login: 'iliakan', id: 349336,
u/349336?v=4', gravatar_id: '', ...}
Fetched Response ILYA KANTOR

```

Task 23:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Promise Chain Example</title>
</head>
<body>
  <script>
    async function res() {
      try {
        let response = await fetch('http://no-such-url');
      } catch (err) {
        console.log(err);
      }
    }
    res();
  </script>
</body>
</html>

```

```
TypeError: Failed to fetch
    at res (d:\f142 js\Task.html:12:38)
    at file:///D:/f142%20js/Task.html:17:9 {stack: 'TypeError: Failed to fetch
    at res (file:/...8)
    at file:///D:/f142%20js/Task.html:17:9', message: 'Failed to fetch'}
```

Task 24:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Promise Chain Example</title>
</head>
<body>
  <script>
    async function name(){
      let urls = [
        'https://api.github.com/users/iliakan',
        'https://api.github.com/users/remy',
        'https://api.github.com/users/jeresig'
      ];
      let requests = await Promise.all(urls.map(url => fetch(url)))
      let obj = await Promise.all(requests.map((res)=>res.json()))
      return obj;
    }
    name().then(responses => responses.forEach(
      response => console.log(`${response.name}: ${response.login}`)
    )).catch(error => {
      console.error('Error:', error);
    });
  </script>
</body>
</html>
```

```
Ilya Kantor: iliakan
Remy Sharp: remy
John Resig: jeresig
```

Task 25:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript Quiz</title>
</head>
<body>
  <script>
    orderfood = (order) => {
      document.write(`Getting Order!.....`)
      document.write("<br>")
    }
  </script>
</body>
</html>
```

```

return new Promise((resolve)=>{
  setTimeout(() => {
    document.write(`Order is placed for ${order}`)
    document.write("<br>")
    resolve(order)
  }, 1000);
})
}
preparefood = (order) => {
  document.write(`Your Food ${order} is Preparing!.....`)
  document.write("<br>")
  return new Promise((resolve)=>{
    setTimeout(() => {
      document.write(`Your Food ${order} is Prepared!.....`)
      document.write("<br>")
      resolve(order)
    }, 1000);
  })
}
deliverfood = (order) => {
  document.write(`Getting Order to delivery!.....`)
  document.write("<br>")
  return new Promise((resolve)=>{
    setTimeout(() => {
      document.write(`${order} is delivered`)
      document.write("<br>")
      resolve(order)
    }, 1500);
  })
}
food = async(value) => {
  const order = await orderfood(value)
  const Prepare = await preparefood(order)
  const deliver = await deliverfood(Prepare)
  if(deliver == order){
    document.write("Thank You!.....");
  }else{
    document.write("Please Wait!...")
  }
}
value = prompt("Enter the Dish you desire: ")
food(value)
</script>
</body>
</html>

```

```

Order is placed for Pizza
Your Food Pizza is Preparing!.....
Your Food Pizza is Prepared!.....
Getting Order to delivery!.....
Pizza is delivered
Thank You!.....

```

6. Modules introduction, Export and Import:

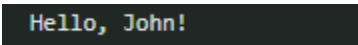
Task 26:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Promise Chain Example</title>
</head>
<body>
  <script>
    export let name = ()=>{console.log("Hell!js..")};
    export const val = 2015;
    export class User {
      constructor(name) {
        this.name = name;
      }
      sayhi = ()=>{console.log("Hai");}
    }
  </script>
</body>
</html>
```



Task 27:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script type="module">
    import { sayHi } from './task.js';
    sayHi('John');
  </script>
</body>
</html>
```



Task 28:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Promise Chain Example</title>
</head>
<body>
```



```

<script type="module">
  export function sayHi(user) {
    console.log(`Hello, ${user}!`);
  }

  export function sayBye(user) {
    console.log(` Bye, ${user}!`);
  }
  sayHi("Nandha")
</script>
</body>
</html>

```

Hello, Nandha!

Task 29:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script type="module">
    import { sayHi,sayBye } from './task.js';
    sayHi('John');
    sayBye('john')
  </script>
</body>
</html>

```

Hello, John!

Bye, john!

Task 30:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script type="module">
    import User from './task.js';
    new User('John');
  </script>
</body>
</html>

```

John