

**index2.html**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JS with College</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <div>
11    abcd<div></div>
12    <p>abcd</p>
13  </div>
14  <h1 class="myclass">DOM Demo by College</h1>
15  <h1 class="myclass">JavaScript</h1>
16  <h4 class="myclass">Topic1 : Starter Code</h4>
17  <p>Let's learn about DOM concepts in detail.</p>
18  <p>2nd paragraph</p>
19  <button id="myId">Click me!</button>
20  <script src="script2.js"></script>
21 </body>
22 </html>
```

## index2.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JS with College</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <h1>old Heading</h1>
11  <div>
12    <h3><i>Fruits</i></h3>
13    <ul>
14      <li>mango</li>
15      <li>orange</li>
16      <li>litchi</li>
17    </ul>
18  </div>
19  <script src="script2.js"></script>
20 </body>
21 </html>
```

## script2.js

```
1 let div = document.querySelector("div");
2 console.dir(div);
3
4 let heading = document.querySelector("h1");
5
```

## style.css

```
1 .heading {  
2     color: blue  
3     background-color: yellow  
4 }  
5  
6 body {  
7     background-color: pink  
8 }  
9  
10 button {  
11     background-color: blue  
12     color: white  
13 }
```

## index2.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JS with College</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <h2> Hello JavaScript!</h2>
11  <div class="box">first div</div>
12  <div class="box">second div</div>
13  <div class="box">third div</div>
14  <div class="box">four div</div>
15  <div class="box">five div</div>
16  <div class="box">fix div</div>
17  <div class="box">seven div</div>
18  <div class="box">eight div</div>
19  <div class="box">nine div</div>
20  <script src="script2.js"></script>
21 </body>
22 </html>
```

## script2.js

```
1 let divs = document.querySelectorAll(".box");
2
3 let idx = 1;
4 for(div of divs) {
5     div.innerText = `new unique value ${idx}`;
6     idx++;
7 }
8
9 // divs[0].innerText = "new unique value 1";
10 // divs[1].innerText = "new unique value 2";
11 // divs[2].innerText = "new unique value 3";
12
```

## style.css

```
1 .box {  
2     height: 100px;  
3     width: 100px;  
4     border: 1px solid black;  
5     margin: 5px;  
6     text-align: center;  
7     background-color: # aquamarine  
8 }
```

### script2.js

```
1 let div = document.querySelector("div");
2 console.log(div);
3
4 let id = div.getAttribute("id");
5 console.log(id);
6
7 let name = div.getAttribute("name");
8 console.log(name);
9
```

### script2.js

```
1 let para = document.querySelector("p");
2 console.log(para.setAttribute("class", "newClass"));
3
```

### script2.js

```
1 let div = document.querySelector("div");
2
3 div.style.backgroundColor = "green";
4 div.style.backgroundColor = "purple";
5 // div.style.visibility = "hidden";
6
7 div.style.fontSize = "23px";
8
9 div.innerText = "Hello!";
10
```

## script2.js

```
1 let newBtn = document.createElement("button");
2 newBtn.innerText = "click me!";
3 console.log(newBtn);
4
5 // let div = document.querySelector("div");
6 let p = document.querySelector("p");
7 // div.append(newBtn);
8 // div.prepend(newBtn);
9 // div.before(newBtn);
10 // div.after(newBtn);
11 p.append(newBtn);
12
```

## script2.js

```
1 let newHeading = document.createElement("h1");
2 newHeading.innerHTML = "<i>Hi, I am new!</i>";
3
4 document.querySelector("body").prepend(newHeading);
5
6 let para = document.querySelector("p");
7 para.remove();
8
9 // newHeading.remove();
10
```

## index2.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>JS with College</title>
7     <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10    <div id="box" name="JSDiv">
11        this is div
12        <ul>
13            List
14            <li>item1</li>
15            <li>item2</li>
16            <li>item3</li>
17        </ul>
18    </div>
19    <p class="para">this is a simple line</p>
20    <script src="script2.js"></script>
21 </body>
22 </html>
```

## style.css

```
1 #box {  
2     height: 250px;  
3     width: 100px;  
4     background-color:  dodgerblue  
5     color: white  
6     border: 1px solid black;  
7     text-align: center;  
8 }  
9 
```

## index2.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JS with College</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <p class="content">i am a paragraph.</p>
11  <script src="script2.js"></script>
12 </body>
13 </html>
```

### script2.js

```
1 let newBtn = document.createElement("button");
2 newBtn.innerText = "click me!";
3
4 newBtn.style.color = "white";
5 newBtn.style.backgroundColor = "red";
6
7 document.querySelector("body").prepend(newBtn);
8
9 //Qs2
10 let para = document.querySelector("p");
```

### style.css

```
1 .content {  
2     color: red;  
3 }  
4  
5 .newClass {  
6     background-color: green;  
7 }  
8
```

## index2.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Events in JS</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <button onclick="console.log('button was clicked');alert('hello!')">
11    click me!
12  </button>
13  <button ondblclick="console.log('button was clicked 2x');">
14    click me 2 times!
15  </button>
16  <div onmouseover="console.log('you are inside div')">this is a box</div>
17  <script src="script2.js"></script>
18 </body>
19 </html>
```

## script2.js

```
1 let btn1 = document.querySelector("#btn1");
2
3 // btn1.onclick = () => {
4 //     console.log("btn1 was clicked");
5 //     let a = 25;
6 //     a++;
7 //     console.log(a);//26
8 // };
9
10 btn1.onclick = (evt) => {
11     console.log(evt);
12     console.log(evt.type);
13     console.log(evt.target);
14     console.log(evt.clientX, evt.clientY);
15 };
16
17 let div = document.querySelector("div");
18 div.onmouseover = (evt) => {
19     // console.log("you are inside div");
20     console.log(evt);
21     console.log(evt.type);
22     console.log(evt.target);
23     console.log(evt.clientX, evt.clientY);
24 };
25
```

## script2.js

```
1 let btn1 = document.querySelector("#btn1");
2
3 btn1.addEventListener("click", (evt) => {
4     console.log("button1 was clicked");
5     console.log(evt);
6     console.log(evt.type);
7 });
8
9 btn1.addEventListener("click", () => {
10    console.log("button1 was clicked - handler2");
11 });
12
13 let div = document.querySelector("div");
14
15
```

## index2.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Events in JS</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <button id="btn1" onclick="console.log('hello')">Btn1</button>
11  <button ondblclick="console.log('button was clicked 2x');">
12    click me 2 times!
13  </button>
14  <div>this is a box</div>
15  <script src="script2.js"></script>
16 </body>
17 </html>
```

## script2.js

```
1 let btn1 = document.querySelector("#btn1");
2
3 btn1.addEventListener("click", (evt) => {
4     console.log("button1 was clicked - handler1");
5 });
6
7 btn1.addEventListener("click", () => {
8     console.log("button1 was clicked - handler2");
9 });
10
11 const handler3 = () => {
12     console.log("button1 was clicked - handler3");
13 };
14
15 btn1.addEventListener("click", handler3);
16
17 btn1.addEventListener("click", (evt) => {
18     console.log("button1 was clicked - handler4");
19 });
20
21 btn1.removeEventListener("click", handler3);
22
```

## style.css

```
1 div {  
2     height: 100px;  
3     width: 100px;  
4     background-color: dodgerblue  
5     color: white  
6     border: 1px solid black;  
7 }  
8
```

**index2.html**

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Events in JS</title>
7   <link rel="stylesheet" href="style.css">
8 </head>
9 <body>
10  <button id="mode">change mode</button>
11  <p>Welcome to my website!</p>
12  <script src="script2.js"></script>
13 </body>
14 </html>
```

## script2.js

```
1 let modeBtn = document.querySelector("#mode");
2 let body = document.querySelector("body");
3 let currMode = "light"; //dark
4
5 modeBtn.addEventListener("click", () => {
6     if(currMode === "light") {
7         currMode = "dark";
8         // document.querySelector("body").style.backgroundColor = "black";
9         body.classList.add("dark");
10        body.classList.remove("light");
11    } else {
12        currMode = "light";
13        // document.querySelector("body").style.backgroundColor = "white";
14        body.classList.add("light");
15        body.classList.remove("dark");
16    }
17    console.log(currMode);
18 });
19
```

## style.css

```
1 .dark {  
2     background-color: ■ black  
3     color: □ white  
4 }  
5  
6 .light {  
7     background-color: □ white  
8     color: ■ black  
9 }  
10
```

### script2.js

```
1 const student = {  
2     fullName: "Shubham Kumar",  
3     marks: 71,  
4     printMarks: function () {  
5         console.log("marks = ", this.marks);  
6     },  
7 };  
8
```

### script2.js

```
1 const employee = {
2     calcTax() {
3         console.log("tax rate is 10%");
4     },
5     // calcTax2 : function {
6     //     console.log("tax rate is 10%");
7     // },
8 };
9
10 const KaranArjun = {
11     salary: 50000,
12 };
13
14 const KaranArjun2 = {
15     salary: 70000,
16 };
17
18 const KaranArjun3 = {
19     salary: 80000,
20 };
21
22 const KaranArjun4 = {
23     salary: 90000,
24 };
25
26 const KaranArjun5 = {
27     salary: 30000,
28 };
29
30 const KaranArjun6 = {
31     salary: 60000,
32 };
33
34
35 KaranArjun.__proto__ = employee;
36 KaranArjun2.__proto__ = employee;
```

```
37 | KaranArjun3.__proto__ = employee;  
38 | KaranArjun4.__proto__ = employee;  
39 | KaranArjun5.__proto__ = employee;  
40 | KaranArjun6.__proto__ = employee;  
41 |
```

### script2.js

```
1 const employee = {  
2     calcTax() {  
3         console.log("tax rate is 10%");  
4     },  
5 };  
6  
7 const KaranArjun = {  
8     salary: 50000,  
9     calcTax() {  
10        console.log("tax rate is 20%");  
11    },  
12 };  
13  
14 KaranArjun.__proto__ = employee;  
15
```

## script2.js

```
1 class ToyotaCar {
2     constructor(brand) {
3         console.log("creating new object");
4         this.brand = brand;
5     }
6
7     start() {
8         console.log("start");
9     }
10
11    stop() {
12        console.log("stop");
13    }
14    // setBrand(brand) {
15    //     this.brandName = brand;
16    // }
17
18 }
19
20 let fortuner = new ToyotaCar("fortuner"); // constructor
21 console.log(fortuner);
22 // fortuner.setBrand("fortuner");
23 let lexus = new ToyotaCar("lexus"); // constructor
24 // lexus.setBrand("lexus");
25 console.log(lexus);
26
```

### script2.js

```
1 class Parent {  
2     hello() {  
3         console.log("hello");  
4     }  
5 }  
6  
7 class Child extends Parent {}  
8  
9 let obj = new Child();  
10
```

### script2.js

```
1 class Person {
2     constructor() {
3         this.species = "homo sapiens";
4     }
5     eat() {
6         console.log("eat");
7     }
8
9     sleep() {
10        console.log("sleep");
11    }
12
13    work() {
14        console.log("do nothing");
15    }
16 }
17
18 class Engineer extends Person {
19     work() {
20         console.log("solve problems, build something");
21     }
22 }
23
24 class Doctor extends Person {
25     work() {
26         console.log("treat patients");
27     }
28 }
29
30 let ShubhamObj = new Engineer();
31
```

### script2.js

```
1 class Person {
2     constructor() {
3         console.log("enter parent construncor");
4         this.species = "homo sapiens";
5     }
6     eat() {
7         console.log("eat");
8     }
9
10    sleep() {
11        console.log("sleep");
12    }
13
14    work() {
15        console.log("do nothing");
16    }
17 }
18
19 class Engineer extends Person {
20     constructor(branch) {
21         console.log("enter child construncor");
22         super(); // to invoke parent class constructor
23         this.branch = branch;
24         console.log("exit child construncor");
25     }
26     work() {
27         console.log("solve problems, build something");
28     }
29 }
30
31 let engObj = new Engineer("chemical engg");
32
```

### script2.js

```
1  class Person {
2      constructor(name) {
3          this.species = "homo sapiens";
4          this.name = name;
5      }
6      eat() {
7          console.log("eat");
8      }
9  }
10
11 class Engineer extends Person {
12     constructor(name) {
13         super(name); // to invoke parent class constructor
14     }
15     work() {
16         super.eat();
17         console.log("solve problems, build something");
18     }
19 }
20
21 let engObj = new Engineer("Shubham");
22
```

### script2.js

```
1 // Practice Qs1
2 let DATA = "secret information";
3
4 class User {
5     constructor(name, email) {
6         this.name = name;
7         this.email = email;
8     }
9
10    viewData() {
11        console.log("data = ", DATA);
12    }
13 }
14
15 let student1 = new User("Shubham", "kumarshubham20167@email.com");
16 let student2 = new User("aman", "aman@email.com");
17
18 let teacher1 = new User("Dean", "dean@college.com");
19
```

### script2.js

```
1 // Practice Qs2
2 let DATA = "secret information";
3
4 class User {
5     constructor(name, email) {
6         this.name = name;
7         this.email = email;
8     }
9
10    viewData() {
11        console.log("data = ", DATA);
12    }
13}
14
15 class Admin extends User {
16     constructor(name, email) {
17         super(name, email);
18     }
19     editData() {
20         DATA = "some new value";
21     }
22}
23
24 let student1 = new User("Shubham", "kumarshubham20167@email.com");
25 let student2 = new User("Aman", "aman@email.com");
26
27 let teacher1 = new User("Dean", "dean@college.com");
28
29 let admin1 = new Admin("Admin", "admincollege@email.com");
30
```

### script2.js

```
1 // Error Handling
2 let a = 5;
3 let b = 10;
4 console.log("a = ", a);
5 console.log("b = ", b);
6 console.log("a + b = ", a + b);
7
8 try {
9     console.log("a + b = ", a + c); //error
10 } catch(err) {
11     console.log(err);
12 }
13
14 console.log("a + b = ", a + b);
15 console.log("a + b = ", a + b);
16 console.log("a + b = ", a + b);
17 console.log("a + b = ", a + b);
18
19
```

### script2.js

```
1 // Synchronous  
2 console.log("one");  
3 console.log("two");  
4 console.log("three");  
5
```

### script2.js

```
1 //Asynchronous
2 /* function hello() {
3     console.log("hello");
4 }
5
6 setTimeout(hello, 4000) */ //timeout ; 2s = 2000ms
7
8 console.log("one");
9 console.log("two");
10
11 setTimeout(() => {
12     console.log("hello");
13 }, 4000); //timeout ; 2s = 2000ms
14
15 console.log("three");
16 console.log("four");
17
```

### script2.js

```
1 // Callbacks
2 function sum(a, b) {
3     console.log(a+b);
4 }
5
6 function calculator(a, b, sumCallback) {
7     sumCallback(a, b)
8 }
9
10 // calculator(1, 2, sum);
11 calculator(1, 3, (a, b) => {
12     console.log(a + b);
13 });
14
15 // Asynchronous
16 const hello = () => {
17     console.log("hello");
18 };
19
20 setTimeout(hello, 3000);
21
```

### script2.js

```
1 // Nesting
2 let age = 19;
3 if(age >= 18) {
4     if(age >= 60){
5         console.log("senior");
6     } else {
7         console.log("middle");
8     }
9 } else {
10    console.log("child");
11 }
12
```

### script2.js

```
1 // Nesting
2
3 for(let i = 0; i < 5; i++) {
4     let str = "";
5     for(let j = 0; j < 5; j++) {
6         str = str + j;
7     }
8     console.log(i, str);
9 }
10
```

### script2.js

```
1 function getData(dataId, getNextData) { // 2s
2     setTimeout(() => {
3         console.log("data", dataId);
4         if(getNextData) {
5             getNextData();
6         }
7     }, 2000);
8 }
9
10 // Callback Hell
11 getData(1, () => {
12     console.log("getting data1 ....")
13     getData(2, () => {
14         console.log("getting data2 ....")
15         getData(3, () => {
16             console.log("getting data3 ....")
17             getData(4, () => {
18                 console.log("getting data4 ....")
19                 getData(5, () => {
20                     console.log("getting data5 ....")
21                     getData(6, () => {
22                         console.log("getting data6 ....")
23                         getData(7)
24                     });
25                 });
26             });
27         });
28     });
29 });
30 /* getData(2); //2s
31 getData(3); */ //2s
32
33 // data1
34 // data2
35 // data3
36
```



### script2.js

```
1 // Promises
2 // let promise = new Promise((resolve, reject) => {
3 //     console.log("I am a promise");
4 //     // resolve("success");
5 //     resolve("some error occurred");
6 // });
7
8 function getData(dataId, getNextData) {
9     return new Promise((resolve, reject) => {
10         setTimeout(() => {
11             console.log("data", dataId);
12             resolve("success");
13             // reject("error");
14             if (getNextData) {
15                 getNextData();
16             }
17         }, 5000);
18     });
19 }
20
```

### script2.js

```
1 // Promises
2 let promise = new Promise((resolve, reject) => {
3     console.log("I am a promise");
4     // resolve("success");
5     resolve("some error occurred");
6 });
7
```

## script2.js

```
1 const getPromise = () => {
2     return new Promise((resolve, reject) => {
3         console.log("I am a promise");
4         resolve("success");
5         // reject("network error");
6     });
7 };
8
9 let promise = getPromise();
10 promise.then((res) => {
11     console.log("promise fulfilled", res);
12 });
13
14 promise.catch((err) => {
15     console.log("rejected", err);
16 });
17
```

### script2.js

```
1 function asyncFunc() {
2     return new Promise((resolve, reject) => {
3         setTimeout(() => {
4             console.log("some data1");
5             resolve("success");
6         }, 4000);
7     });
8 }
9
10 console.log("fetching data1....");
11 let p1 = asyncFunc();
12 p1.then((res) => {
13     console.log(res);
14 })
15
```

### script2.js

```
1 function asyncFunc1() {
2     return new Promise((resolve, reject) => {
3         setTimeout(() => {
4             console.log("data1");
5             resolve("success");
6         }, 4000);
7     });
8 }
9
10 function asyncFunc2() {
11     return new Promise((resolve, reject) => {
12         setTimeout(() => {
13             console.log("data2");
14             resolve("success");
15         }, 4000);
16     });
17 }
18
19 function asyncFunc3() {
20     return new Promise((resolve, reject) => {
21         setTimeout(() => {
22             console.log("data3");
23             resolve("success");
24         }, 4000);
25     });
26 }
27
28 console.log("fetching data1....");
29 let p1 = asyncFunc1();
30 p1.then((res) => {
31     console.log(res);
32 });
33
34 console.log("fetching data2....");
35 let p2 = asyncFunc2();
36 p2.then((res) => {
```

```
37     console.log(res);
38 });
39
40 console.log("fetching data3....");
41 let p3 = asyncFunc3();
42 p3.then((res) => {
43     console.log(res);
44 });
45
46
47
```

## script2.js

```
1 function asyncFunc1() {
2     return new Promise((resolve, reject) => {
3         setTimeout(() => {
4             console.log("data1");
5             resolve("success");
6         }, 4000);
7     });
8 }
9
10 function asyncFunc2() {
11     return new Promise((resolve, reject) => {
12         setTimeout(() => {
13             console.log("data2");
14             resolve("success");
15         }, 4000);
16     });
17 }
18
19 function asyncFunc3() {
20     return new Promise((resolve, reject) => {
21         setTimeout(() => {
22             console.log("data3");
23             resolve("success");
24         }, 4000);
25     });
26 }
27
28 // console.log("fetching data1....");
29 // let p1 = asyncFunc1();
30 // p1.then((res) => {
31 //     console.log(res);
32 //     console.log("fetching data2....");
33 //     let p2 = asyncFunc2();
34 //     p2.then((res) => {
35 //         console.log(res);
36 //     });
37 }
```

```
37 // });
38
39 console.log("fetching data1....");
40 asyncFunc1().then((res) => {
41   console.log("fetching data2....");
42   asyncFunc2().then((res) => {
43     console.log("fetching data3....");
44     asyncFunc3().then((res) => {});
45   });
46 });
47
48 // console.log("fetching data3....");
49 // let p3 = asyncFunc3();
50 // p3.then((res) => {
51 //   console.log(res);
52 // });
53
54
55
```

## script2.js

```
1 function getData(dataId, getNextData) {
2     return new Promise((resolve, reject) => {
3         setTimeout(() => {
4             console.log("data", dataId);
5             resolve("success");
6             if (getNextData) {
7                 getNextData();
8             }
9         }, 3000)
10    });
11 }
12
13 // Promise Chain
14 // let p1 = getData(1);
15 // p1.then((res) => {
16 //     console.log(res);
17 // });
18
19 getData(1).then((res) => {
20     console.log(res);
21     getData(2).then(() => {
22         console.log(res);
23     });
24 });
25
26
27
```

### script2.js

```
1 function getData(dataId, getNextData) {
2     return new Promise((resolve, reject) => {
3         setTimeout(() => {
4             console.log("data", dataId);
5             resolve("success");
6             if (getNextData) {
7                 getNextData();
8             }
9         }, 2000)
10    });
11 }
12
13 // Promise Chain
14
15 console.log("getting data1 ....");
16 getData(1)
17 .then((res) => {
18     console.log("getting data2 ....");
19     return getData(2);
20 })
21 .then((res) => {
22     console.log("getting data3 ....");
23     return getData(3);
24 })
25 .then((res) => {
26     console.log("getting data4 ....");
27     return getData(4);
28 })
29 .then((res) => {
30     console.log(res);
31 });
32 }
```

### script2.js

```
1 | async function hello() {
2 |     console.log("hello");
3 | }
4 |
5 | function getData(dataId, getNextData) {
6 |     return new Promise((resolve, reject) => {
7 |         setTimeout(() => {
8 |             console.log("data", dataId);
9 |             resolve("success");
10 |             if (getNextData) {
11 |                 getNextData();
12 |             }
13 |         }, 2000)
14 |     });
15 |
16 | }
```

## script2.js

```
1 function api() {
2     return new Promise((resolve, reject) => {
3         setTimeout(() => {
4             console.log("wether data");
5             resolve(200);
6         }, 2000);
7     });
8 }
9
10 async function getWetherData() {
11     await api(); //1st
12     await api(); //2nd
13 }
14
15 function getData(dataId, getNextData) {
16     return new Promise((resolve, reject) => {
17         setTimeout(() => {
18             console.log("data", dataId);
19             resolve("success");
20             if (getNextData) {
21                 getNextData();
22             }
23         }, 2000)
24     });
25 }
26 }
```

### script2.js

```
1 function getData(dataId, getNextData) {
2     return new Promise((resolve, reject) => {
3         setTimeout(() => {
4             console.log("data", dataId);
5             resolve("success");
6         }, 2000)
7     });
8 }
9
10 //Async-await
11 async function getAllData() {
12     console.log("getting data1....");
13     await getData(1);
14     console.log("getting data2....");
15     await getData(2);
16     console.log("getting data3....");
17     await getData(3);
18     console.log("getting data4....");
19     await getData(4);
20     console.log("getting data5....");
21     await getData(5);
22     console.log("getting data6....");
23     await getData(6);
24 }
25
```

### script2.js

```
1 function getData(dataId, getNextData) {
2     return new Promise((resolve, reject) => {
3         setTimeout(() => {
4             console.log("data", dataId);
5             resolve("success");
6         }, 2000)
7     });
8 }
9
10 //Async-await
11 (async function () {
12     console.log("getting data1....");
13     await getData(1);
14     console.log("getting data2....");
15     await getData(2);
16     console.log("getting data3....");
17     await getData(3);
18     console.log("getting data4....");
19     await getData(4);
20     console.log("getting data5....");
21     await getData(5);
22     console.log("getting data6....");
23     await getData(6);
24 })();
```

### script2.js

```
1 const URL = "https://dog.ceo/api/breeds/image/random";
2
3 let promise = fetch(URL);
4 console.log(promise);
5
```

### script2.js

```
1 const URL = "https://dog.ceo/api/breeds/image/random";
2
3 const getFacts = async() => {
4     console.log("getting data .....");
5     let response = await fetch(URL);
6     console.log(response); //JSON format
7     let data = await response.json();
8     console.log(data);
9 };
10
```

### script2.js

```
1 const URL  = "https://api.thecatapi.com/v1/images/0XYvRd7oD";
2 const btn = document.querySelector("#btn");
3
4 const getFacts = async() => {
5   console.log("getting data ....");
6   let response = await fetch(URL);
7   console.log(response); //JSON format
8   let data = await response.json();
9   console.log(data);
10 };
11
12 btn.addEventListener("click", getFacts);
13
```

### script2.js

```
1 const URL  = "https://cat-fact.herokuapp.com/facts";
2 const btn = document.querySelector("#btn");
3
4 function getFacts() {
5   fetch(URL).this((response) => {
6     return response.json();
7   })
8   .then((data) => {
9     console.log(data);
10  });
11 }
12
13 btn.addEventListener("click", getFacts);
14
```