

Code Issues Pull requests 2 Actions Projects Wiki Security Insights

## add asyncn javascript code

[Browse files](#)

main (#4)



harshitvee committed on Oct 31, 2021

1 parent [ca7d64e](#) commit [9494d64a6eb1baa0b8736e47355b95b21361c47a](#)

Showing 26 changed files with 1,294 additions and 0 deletions.

Split

Unified

26 part3/asynchronous-javascript/130.js

```
@@ -0,0 +1,26 @@
1 + // synchronous programming vs asynchronous programming
2 + // synchronous programming
3 + // synchronous programming single threaded
4 + // console.log("script start");
5 +
6 + // for (let i = 1; i < 10000; i++) {
7 + //   console.log("inside for loop");
8 + // }
9 +
10 + // console.log("script end");
11 +
12 + // setTimeout
13 +
14 + console.log("script start");
15 +
16 + const id = setTimeout(() => {
17 +   console.log("inside setTimeout");
18 + }, 1000);
19 +
20 + for (let i = 1; i < 100; i++) {
21 +   console.log("....");
22 + }
23 + console.log("setTimeout id is ", id);
24 + console.log("clearing time out");
25 + clearTimeout(id);
26 + console.log("Script end");
```


23 part3/asynchronous-javascript/131.js

```
@@ -0,0 +1,23 @@
```

```

1 + // setInterval
2 + // console.log("script start");
3 + // // setInterval(() => {
4 + // //   console.log(total);
5 + // //   console.log(Math.random());
6 + // // }, 500);
7 + // console.log("script end");
8 + const body = document.body;
9 + const button = document.querySelector("button");
10 + const intervalId = setInterval(() => {
11 +   const red = Math.floor(Math.random() * 256);
12 +   const green = Math.floor(Math.random() * 256);
13 +   const blue = Math.floor(Math.random() * 256);
14 +   const rgb = `rgb(${red},${green}, ${blue})`;
15 +   body.style.background = rgb;
16 + }, 1000);
17 +
18 + button.addEventListener("click", () => {
19 +   clearInterval(intervalId);
20 +   button.textContent = body.style.background;
21 + });
22 +
23 + console.log(intervalId);

```


✓ 28 ■■■■■ part3/asynchronous-javascript/132.js 

```

...    ...    @@ -0,0 +1,28 @@
1 + // understand callback
2 +
3 + // function myFunc(callback) {
4 + //   console.log("Function is doing task 1 ");
5 + //   callback();
6 + // }
7 +
8 + // myFunc(() => {
9 + //   console.log("function is doing task 2");
10 + // });
11 +
12 + function getTwoNumbersAndAdd(number1, number2, onSuccess, onFailure) {
13 +   if (typeof number1 === "number" && typeof number2 === "number") {
14 +     onSuccess(number1, number2);
15 +   } else {
16 +     onFailure();
17 +   }
18 + }
19 +
20 + function addTwoNumbers(num1, num2) {
21 +   console.log(num1 + num2);
22 + }

```

```
23 +
24 + function onFail(){
25 +     console.log("Wrong data type");
26 +     console.log("please pass numbers only")
27 + }
28 + getTwoNumbersAndAdd(4, 4, addTwoNumbers, onFail);
```

▼ 86 ■■■■■ part3/asynchronous-javascript/133.js 

```
...    ...    @@ -0,0 +1,86 @@
1 + // callbacks , callback hell, pyramid of doom
2 + // asynchronous programming
3 + const heading1 = document.querySelector(".heading1");
4 + const heading2 = document.querySelector(".heading2");
5 + const heading3 = document.querySelector(".heading3");
6 + const heading4 = document.querySelector(".heading4");
7 + const heading5 = document.querySelector(".heading5");
8 + const heading6 = document.querySelector(".heading6");
9 + const heading7 = document.querySelector(".heading7");
10 + const heading8 = document.querySelector(".heading8");
11 + const heading9 = document.querySelector(".heading9");
12 + const heading10 = document.querySelector(".heading10");
13 +
14 + // Text      Delay   Color
15 +
16 + // one       1s      Violet
17 + // two       2s      purple
18 + // three     2s      red
19 + // four      1s      Pink
20 + // five      2s      green
21 + // six       3s      blue
22 + // seven     1s      brown
23 +
24 + // callback hell
25 + // setTimeout(()=>{
26 + //     heading1.textContent = "one";
27 + //     heading1.style.color = "violet";
28 + //     setTimeout(()=>{
29 + //         heading2.textContent = "two";
30 + //         heading2.style.color = "purple";
31 + //         setTimeout(()=>{
32 + //             heading3.textContent = "three";
33 + //             heading3.style.color = "red";
34 + //             setTimeout(()=>{
35 + //                 heading4.textContent = "four";
36 + //                 heading4.style.color = "pink";
37 + //                 setTimeout(()=>{
38 + //                     heading5.textContent = "five";
39 + //                     heading5.style.color = "green";
```

```

40 + //      },2000)
41 +
42 + //      },1000)
43 +
44 + //      },2000)
45 +
46 + //      },2000)
47 +
48 + // },1000)
49 +
50 + function changeText(element, text, color, time, onSuccessCallback, onFailureCallback)
51 +   setTimeout(()=>{
52 +     if(element){
53 +       element.textContent = text;
54 +       element.style.color = color;
55 +       if(onSuccessCallback){
56 +         onSuccessCallback();
57 +       }
58 +     }else{
59 +       if(onFailureCallback){
60 +         onFailureCallback();
61 +       }
62 +     }
63 +   },time)
64 + }
65 + // pyramid of doom
66 + changeText(heading1, "one","violet",1000,()=>{
67 +   changeText(heading2, "two","purple",2000,()=>{
68 +     changeText(heading3, "three","red",1000,()=>{
69 +       changeText(heading4, "four","pink",1000,()=>{
70 +         changeText(heading5, "five","green",2000,()=>{
71 +           changeText(heading6, "six","blue",1000,()=>{
72 +             changeText(heading7, "seven","brown",1000,()=>{
73 +               changeText(heading8, "eight","cyan",1000,()=>{
74 +                 changeText(heading9, "nine","#cda562",1000,()=>{
75 +                   changeText(heading10, "ten","dca652",1000,()=>{
76 +
77 +                     },()=>{console.log("Heading10 does not exist")})
78 +                     },()=>{console.log("Heading9 does not exist")})
79 +                     },()=>{console.log("Heading8 does not exist")})
80 +                     },()=>{console.log("Heading7 does not exist")})
81 +                     },()=>{console.log("Heading6 does not exist")})
82 +                     },()=>{console.log("Heading5 does not exist")})
83 +                     },()=>{console.log("Heading4 does not exist")})
84 +                     },()=>{console.log("Heading3 does not exist")})
85 +                     },()=>{console.log("Heading2 does not exist")})
86 +                     },()=>{console.log("Heading1 does not exist")})

```

```

...      @@ -0,0 +1,41 @@
1  + // Promise
2  + console.log("script start");
3  + const bucket = ['coffee', 'chips', 'vegetables', 'salt', 'rice'];
4  +
5  +
6  +
7  + const friedRicePromise = new Promise((resolve, reject)=>{
8  +     if(bucket.includes("vegetables") && bucket.includes("salt") && bucket.includes("ri
9  +         resolve({value: "friedrice"});
10 +     }else{
11 +         reject("could not do it");
12 +     }
13 + })
14 +
15 + // produce
16 +
17 +
18 +
19 + // consume
20 + // how to consume
21 +
22 + friedRicePromise.then(
23 +     // jab promise resolve hoga
24 +     (myfriedRice)=>{
25 +         console.log("lets eat ", myfriedRice);
26 +     }
27 + ).catch(
28 +     (error)=>{
29 +         console.log(error)
30 +     })
31 +
32 +
33 + setTimeout(()=>{
34 +     console.log("hello from setTimeout")
35 + }, 0)
36 +
37 + for(let i = 0; i <= 100; i++){
38 +     console.log(Math.random(), i);
39 + }
40 +
41 + console.log("script end!!!!") 

```

```

@@ -0,0 +1,23 @@
1 + // function returning promise
2 +
3 + function ricePromise(){
4 +   const bucket = ['coffee', 'chips','vegetables','salts','rice'];
5 +   return new Promise((resolve,reject)=>{
6 +     if(bucket.includes("vegetables")&& bucket.includes("salt") && bucket.includes("ri
7 +       resolve({value:"friedrice"});
8 +     }else{
9 +       reject("could not do it");
10 +    }
11 +  })
12 + }
13 +
14 +
15 + ricePromise().then(
16 +   // jab promise resolve hoga
17 +   (myfriedRice)=>{
18 +     console.log("lets eat ", myfriedRice);
19 +   }
20 +   ).catch(
21 +   (error)=>{
22 +     console.log(error)
23 +   })

```

▼ 22 ■■■ part3/asynchronous-javascript/136.js

```

...   ...   @@ -0,0 +1,22 @@
1 + // promise && setTimeout
2 +
3 +
4 + // I want to resolve / reject promise after 2 seconds
5 +
6 +
7 + function myPromise(){
8 +   return new Promise((resolve, reject)=>{
9 +     const value = false;
10 +     setTimeout(()=>{
11 +       if(value){
12 +         resolve();
13 +       }else{
14 +         reject();
15 +       }
16 +     },2000)
17 +   })
18 + }
19 +
20 + myPromise()

```

```
21 + .then(()=>{console.log("resolved")})
22 + .catch(()=>{console.log("rejected")})
```

35 part3/asynchronous-javascript/137.js

```
... @@ -0,0 +1,35 @@
1 + // Promise.resolve
2 + // Promise chaining
3 +
4 + // const myPromise = Promise.resolve(5);
5 + // Promise.resolve(5).then(value=>{
6 + //   console.log(value);
7 + // })
8 +
9 +
10 + // then()
11 + // then method hamesha promise return karta hai
12 +
13 +
14 + function myPromise(){
15 +   return new Promise((resolve, reject)=>{
16 +     resolve("foo");
17 +   })
18 + }
19 +
20 + myPromise()
21 + .then((value)=>{
22 +   console.log(value);
23 +   value += "bar";
24 +   return value
25 + })
26 + .then((value) =>{
27 +   console.log(value);
28 +   value += "baaz";
29 +   return value;
30 + })
31 + .then(value=>{
32 +   console.log(value);
33 + })
34 +
35 +
```


39 part3/asynchronous-javascript/138.js

```
... @@ -0,0 +1,39 @@
1 + const heading1 = document.querySelector(".heading1");
2 + const heading2 = document.querySelector(".heading");
3 + const heading3 = document.querySelector(".heading3");
```

```

4 + const heading4 = document.querySelector(".heading4");
5 + const heading5 = document.querySelector(".heading5");
6 + const heading6 = document.querySelector(".heading6");
7 + const heading7 = document.querySelector(".heading7");
8 + const heading8 = document.querySelector(".heading8");
9 + const heading9 = document.querySelector(".heading9");
10 + const heading10 = document.querySelector(".heading10");
11 +
12 +
13 + function changeText(element, text, color, time) {
14 +     return new Promise((resolve, reject) => {
15 +         setTimeout(()=>{
16 +             if(element){
17 +                 element.textContent = text;
18 +                 element.style.color = color;
19 +                 resolve();
20 +             }else{
21 +                 reject("element not found");
22 +             }
23 +         },time)
24 +     })
25 + }
26 +
27 + changeText(heading1, "one", "red", 1000)
28 + .then(()=>changeText(heading2, "two", "purple", 1000))
29 + .then(()=>changeText(heading3, "three", "green", 1000))
30 + .then(()=>changeText(heading4, "four", "orange", 1000))
31 + .then(()=>changeText(heading5, "four", "orange", 1000))
32 + .then(()=>changeText(heading6, "four", "orange", 1000))
33 + .then(()=>changeText(heading7, "four", "orange", 1000))
34 + .then(()=>changeText(heading8, "four", "orange", 1000))
35 + .then(()=>changeText(heading9, "four", "orange", 1000))
36 + .then(()=>changeText(heading10, "four", "orange", 1000))
37 + .catch((error)=>{
38 +     alert(error);
39 + })

```

✓ 24 ■■■■ part3/asynchronous-javascript/139.js 

...    ...    @@ -0,0 +1,24 @@

```

1 + // BASIC THEORY
2 +
3 + // AJAX : asynchronous javascript and XML
4 +
5 + // HTTP request
6 +
7 + // is a set of "web development techniques"
8 + // using many web technologies on the "client-side "
9 + // to create asynchronous web applications.


```



```

10 +
11 +
12 + // With Ajax, web applications can send and retrieve
13 + // data from a server asynchronously (in the background)
14 + // without interfering with the display and
15 + // behaviour of the existing page
16 +
17 + // We don't use data in XML format anymore.
18 + // we use JSON now.
19 +
20 +
21 + // we have 3 most common ways to create and send request to server
22 + // 1.) XMLHttpRequest (old way of doing)
23 + // 2.) fetch API (new way of doing)
24 + // 3.) axios (this is third party library)


```

✓ 24 ■■■■■ part3/asynchronous-javascript/140.js 

```

...    ...    @@ -0,0 +1,24 @@
1    + const URL = "https://jsonplaceholder.typicode.com/posts";
2    + const xhr = new XMLHttpRequest();
3    + // console.log(xhr);
4    + // step1
5    + // console.log(xhr.readyState);
6    + xhr.open("GET",URL);
7    + // console.log(xhr.readyState);
8    + // xhr.onreadystatechange = function(){
9    + //     // console.log(xhr.readyState);
10   + //     if(xhr.readyState === 4){
11   + //         console.log(xhr)
12   + //         const response = xhr.response;
13   + //         const data = JSON.parse(response);
14   + //         console.log(typeof data);
15   + //     }
16   + // }
17   +
18   + xhr.onload = function(){
19   +     const response = xhr.response;
20   +     const data = JSON.parse(response);
21   +     console.log(data);
22   + }
23   +
24   + xhr.send(); 

```

✓ 29 ■■■■■ part3/asynchronous-javascript/141.js 

```


...    ...    @@ -0,0 +1,29 @@
1    + const URL = "https://jsonplaceholder.typicode.com/posts";

```

```

2 +
3 + const xhr = new XMLHttpRequest();
4 +
5 + xhr.open("GET", URL);
6 + xhr.onload = () => {
7 +     if(xhr.status >= 200 && xhr.status < 300) {
8 +         const data = JSON.parse(xhr.response);
9 +         console.log(data);
10 +         const id = data[3].id;
11 +         const xhr2 = new XMLHttpRequest();
12 +         const URL2 = `${URL}/${id}`
13 +         console.log(URL2);
14 +         xhr2.open("GET", URL2);
15 +         xhr2.onload = () => {
16 +             const data2 = JSON.parse(xhr2.response);
17 +             console.log(data2);
18 +         }
19 +         xhr2.send();
20 +     }
21 +     else{
22 +         console.log("something went wrong");
23 +     }
24 + }
25 +
26 + xhr.onerror = () => {
27 +     console.log("network error");
28 + }
29 + xhr.send(); ⊖

```

✓ 44 ■■■■■ part3/asynchronous-javascript/142.js 

```


...    ...    @@ -0,0 +1,44 @@
1 + const URL = "https://jsonplaceholder.typicode.com/posts";
2 +
3 + function sendRequest(method, url) {
4 +     return new Promise(function(resolve, reject) {
5 +         const xhr = new XMLHttpRequest();
6 +         xhr.open(method, url);
7 +         xhr.onload = function() {
8 +             if(xhr.status >= 200 && xhr.status < 300){
9 +                 resolve(xhr.response);
10 +             }
11 +             else{
12 +                 reject(new Error("Something Went wrong"));
13 +             }
14 +         }
15 +
16 +         xhr.onerror = function() {
17 +             reject(new Error("Something went wrong"));

```

```

18 +     }
19 +
20 +     xhr.send();
21 + })
22 + }
23 +
24 + sendRequest("GET", URL)
25 + .then(response => {
26 +     const data = JSON.parse(response);
27 +     // console.log(data)
28 +     return data;
29 + })
30 + .then(data=>{
31 +     const id = data[3].id;
32 +     return id;
33 + })
34 + .then(id=>{
35 +     const url = `${URL}/${id}ssss`;
36 +     return sendRequest("GET", url);
37 + })
38 + .then(newResponse => {
39 +     const newData = JSON.parse(newResponse);
40 +     console.log(newData);
41 + })
42 + .catch(error =>{
43 +     console.log(error);
44 + })

```

✓ 29 ■■■■■ part3/asynchronous-javascript/143.js 

```


...    ...    @@ -0,0 +1,29 @@
1 + // fetch
2 +
3 + const URL = "https://jsonplaceholder.typicode.com/postssss";
4 +
5 + fetch(URL,{
6 +     method: 'POST',
7 +     body: JSON.stringify({
8 +         title: 'foo',
9 +         body: 'bar',
10 +         userId: 1,
11 +     }),
12 +     headers: {
13 +         'Content-type': 'application/json; charset=UTF-8',
14 +     },
15 + })
16 + .then(response =>{
17 +     if(response.ok){
18 +         return response.json()

```

```

19 +         }else{
20 +             throw new Error("Something went wrong!!!")
21 +         }
22 +     })
23 +     .then(data =>{
24 +         console.log(data);
25 +     })
26 +     .catch(error =>{
27 +         console.log("inside catch");
28 +         console.log(error);
29 +     })

```

✓ 45 ■■■■■ part3/asynchronous-javascript/144.js 

```

...    ...    @@ -0,0 +1,45 @@
1 + // async await
2 +
3 + // fetch(URL)
4 + //     .then(response => {
5 + //         return response.json()
6 + //     })
7 + //     .then(data => {
8 + //         console.log(data);
9 + //     })
10 + console.log("script start");
11 + const URL = "https://jsonplaceholder.typicode.com/posts";
12 +
13 + // async function getPosts(){
14 + //     const response = await fetch(URL);
15 + //     if(!response.ok){
16 + //         throw new Error("Something went wrong")
17 + //     }
18 + //     const data = await response.json();
19 + //     return data;
20 + // }
21 +
22 +
23 + const getPosts = async() =>{
24 +     const response = await fetch(URL);
25 +     if(!response.ok){
26 +         throw new Error("Something went wrong")
27 +     }
28 +     const data = await response.json();
29 +     return data;
30 + }
31 +
32 + // const myData = getPosts();
33 + // console.log(myData);
34 +

```

```

35 +   getPosts()
36 +       .then((myData) => {
37 +           console.log(myData);
38 +       })
39 +       .catch(error =>{
40 +           console.log("inside catch")
41 +           console.log(error);
42 +       })
43 +
44 +
45 +   console.log("script end ");

```

✓ 13 part3/asynchronous-javascript/async-code.html

```

...   ...   @@ -0,0 +1,13 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 +   <head>
4 +     <meta charset="UTF-8" />
5 +     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6 +     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 +     <!-- <script src="./130.js" defer></script> -->
8 +     <!-- <script src="./131.js" defer></script> -->
9 +     <script src="./132.js" defer></script>
10 +     <title>Asynchronous Javascript</title>
11 +   </head>
12 +   <body></body>
13 + </html>

```

✓ 25 part3/asynchronous-javascript/callback-demo.html

```

...   ...   @@ -0,0 +1,25 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 +   <head>
4 +     <meta charset="UTF-8">
5 +     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 +     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 +     <script src="./133.js" defer></script>
8 +     <style>
9 +       body{font-family:arial,sans-serif}
10 +     </style>
11 +     <title>Document</title>
12 +   </head>
13 +   <body>
14 +     <h1 class="heading1">Hello World</h1>
15 +     <h1 class="heading2">Hello World</h1>
16 +     <h1 class="heading3">Hello World</h1>

```

```

17 + <h1 class="heading4">Hello World</h1>
18 + <h1 class="heading5">Hello World</h1>
19 + <h1 class="heading6">Hello World</h1>
20 + <h1 class="heading7">Hello World</h1>
21 + <h1 class="heading8">Hello World</h1>
22 + <h1 class="heading9">Hello World</h1>
23 + <h1 class="heading10">Hello World</h1>
24 + </body>
25 + </html> ⊖

```

✓ 13 part3/asynchronous-javascript/moreAboutPromise.html 

```

...  ...  @@ -0,0 +1,13 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +   <meta charset="UTF-8">
5 +   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 +   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 +   <script src="./moreAboutPromise.js" defer></script>
8 +   <title>Document</title>
9 + </head>
10 + <body>
11 +   <h1>More about promise</h1>
12 + </body>
13 + </html> ⊖

```

✓ 22 part3/asynchronous-javascript/moreAboutPromise.js 

```

...  ...  @@ -0,0 +1,22 @@
1 + function myPromise(){
2 +   return new Promise((resolve, reject) =>{
3 +     resolve(1);
4 +   })
5 + }
6 +
7 +
8 + myPromise()
9 +   .then((data)=>{
10 +     console.log(data);
11 +     return 2;
12 +   })
13 +   .then(data => {
14 +     console.log(data);
15 +     return 3;
16 +   })
17 +   .then(data => {
18 +     console.log(data);

```

```
19 +     return 4;
20 +     }).then(data => {
21 +         console.log(data);
22 +     })
```



▼ 602 part3/asynchronous-javascript/myfile.json

Load diff

Large diffs are not rendered by default.

▼ 13 part3/asynchronous-javascript/promise.html


```
...  ...  @@ -0,0 +1,13 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +     <meta charset="UTF-8">
5 +     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 +     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 +     <script src="/134.js" defer></script>
8 +     <title>Promise</title>
9 + </head>
10 + <body>
11 +     <h1>Promise</h1>
12 + </body>
13 + </html>
```




▼ 16 part3/asynchronous-javascript/promise\_demo.html

```
...  ...  @@ -0,0 +1,16 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +     <meta charset="UTF-8">
5 +     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 +     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 +     <!-- <script src="/134.js" defer></script> -->
8 +     <!-- <script src="/135.js" defer></script> -->
9 +     <!-- <script src="/136.js" defer></script> -->
10 +     <script src="/137.js" defer></script>
11 +     <title>Document</title>
12 + </head>
13 + <body>
14
```

```
+
15 + </body>
16 + </html> ⊖
```

✓ 25 ■■■■■ part3/asynchronous-javascript/refactor-with-promise.html 

```
...      ...      @@ -0,0 +1,25 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 + <head>
4 +   <meta charset="UTF-8">
5 +   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 +   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 +   <script src="./138.js" defer></script>
8 +   <style>
9 +     body{font-family:arial,sans-serif}
10 +   </style>
11 +   <title>Document</title>
12 + </head>
13 + <body>
14 +   <h1 class="heading1">Hello World 1</h1>
15 +   <h1 class="heading2">Hello World 2</h1>
16 +   <h1 class="heading3">Hello World 3</h1>
17 +   <h1 class="heading4">Hello World 4</h1>
18 +   <h1 class="heading5">Hello World 5</h1>
19 +   <h1 class="heading6">Hello World 6</h1>
20 +   <h1 class="heading7">Hello World 7</h1>
21 +   <h1 class="heading8">Hello World 8</h1>
22 +   <h1 class="heading9">Hello World 9</h1>
23 +   <h1 class="heading10">Hello World 10</h1>
24 + </body>
25 + </html> ⊖
```

✓ 16 ■■■■■ part3/asynchronous-javascript/setInterval.css 


```
...      ...      @@ -0,0 +1,16 @@
1 + button {
2 +   position: absolute;
3 +   top: 50%;
4 +   left: 50%;
5 +   transform: translate(-50%, -50%);
6 +   padding: 1rem 1.5rem;
7 +   border: none;
8 +   outline: none;
9 +   cursor: pointer;
10 +   font-size: 3rem;
11 +   box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.322);
12 +   transition: all 0.4s;
```



```
13 + }
14 + button:hover {
15 +   box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.438);
16 + }
```

✓ 14 ■■■■■ part3/asynchronous-javascript/setInterval.html 

```
...    ...    @@ -0,0 +1,14 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 +   <head>
4 +     <meta charset="UTF-8" />
5 +     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6 +     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 +     <link rel="stylesheet" href="./setInterval.css" />
8 +     <script src="./131.js" defer></script>
9 +     <title>Set Interval Example</title>
10 +   </head>
11 +   <body>
12 +     <button>Stop Changing!!!!</button>
13 +   </body>
14 + </html>
```

✓ 17 ■■■■■ part3/asynchronous-javascript/xhr.html 

```
...    ...    @@ -0,0 +1,17 @@
1 + <!DOCTYPE html>
2 + <html lang="en">
3 +   <head>
4 +     <meta charset="UTF-8">
5 +     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 +     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 +     <!-- <script src="./140.js" defer></script> -->
8 +     <!-- <script src="./141.js" defer></script> -->
9 +     <!-- <script src="./142.js" defer></script> -->
10 +    <!-- <script src="./144.js" defer></script> -->
11 +     <script src="./144.js" defer></script>
12 +     <title>XHR</title>
13 +   </head>
14 +   <body>
15 +
16 +   </body>
17 + </html> 
```

0 comments on commit 9494d64

Please [sign in](#) to comment.