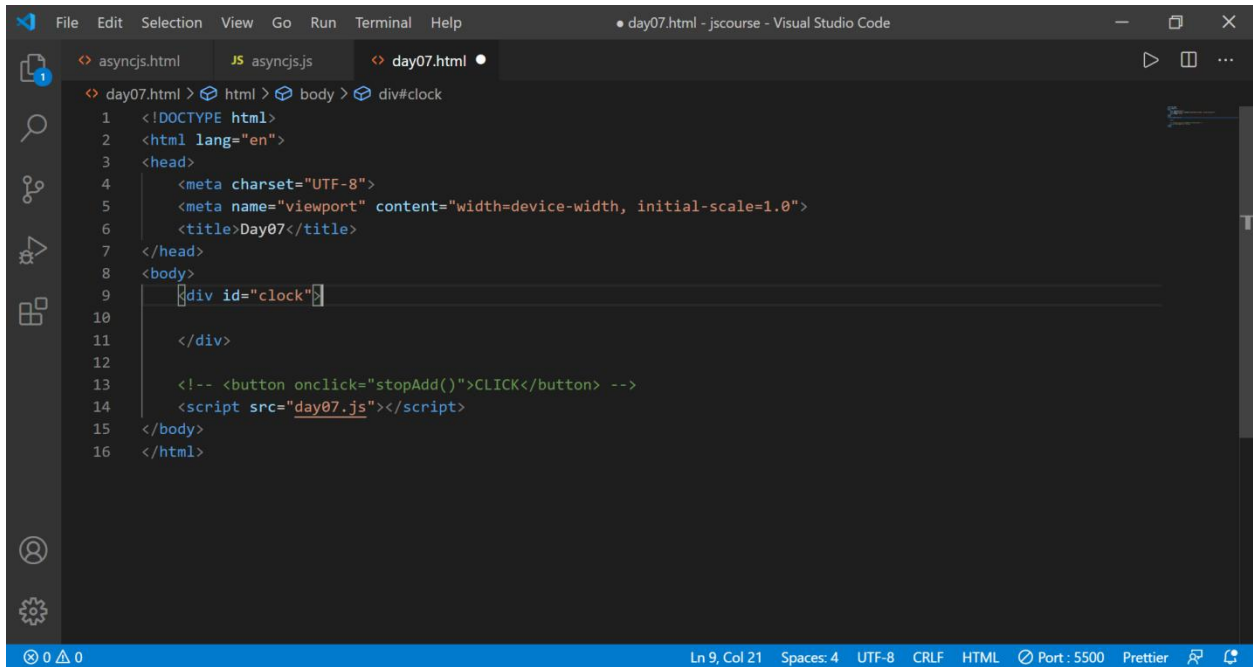


Letsupgrade js essentials Day – 07 notes

HTML CODE 1

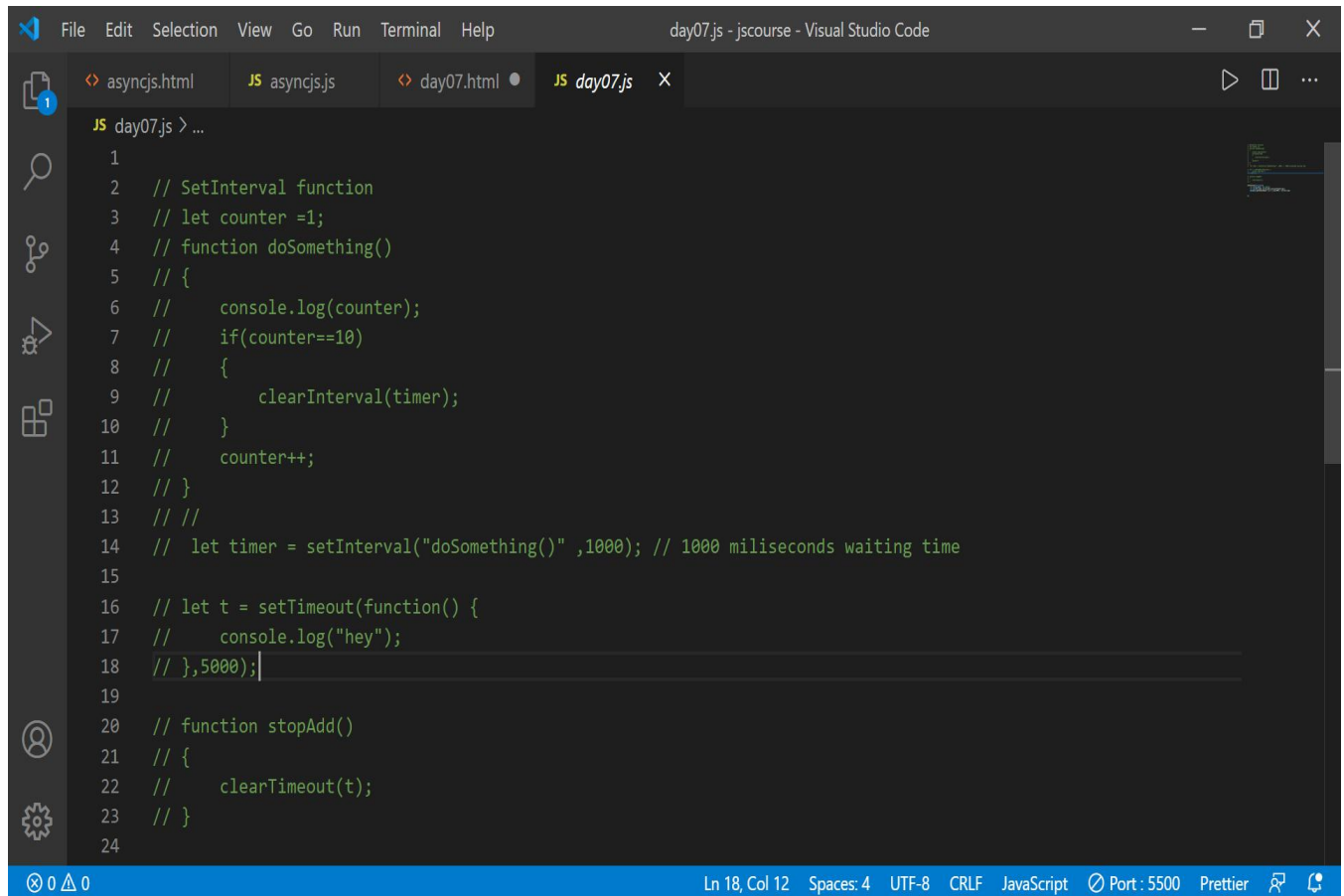


```
File Edit Selection View Go Run Terminal Help
• day07.html - jscourse - Visual Studio Code

day07.html > html > body > div#clock
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>Day07</title>
7 </head>
8 <body>
9   <div id="clock">
10
11   </div>
12
13   <!-- <button onclick="stopAdd()">CLICK</button> -->
14   <script src="day07.js"></script>
15 </body>
16 </html>
```

Ln 9, Col 21 Spaces: 4 UTF-8 CRLF HTML Port: 5500 Prettier

JAVASCRIPT CODE1



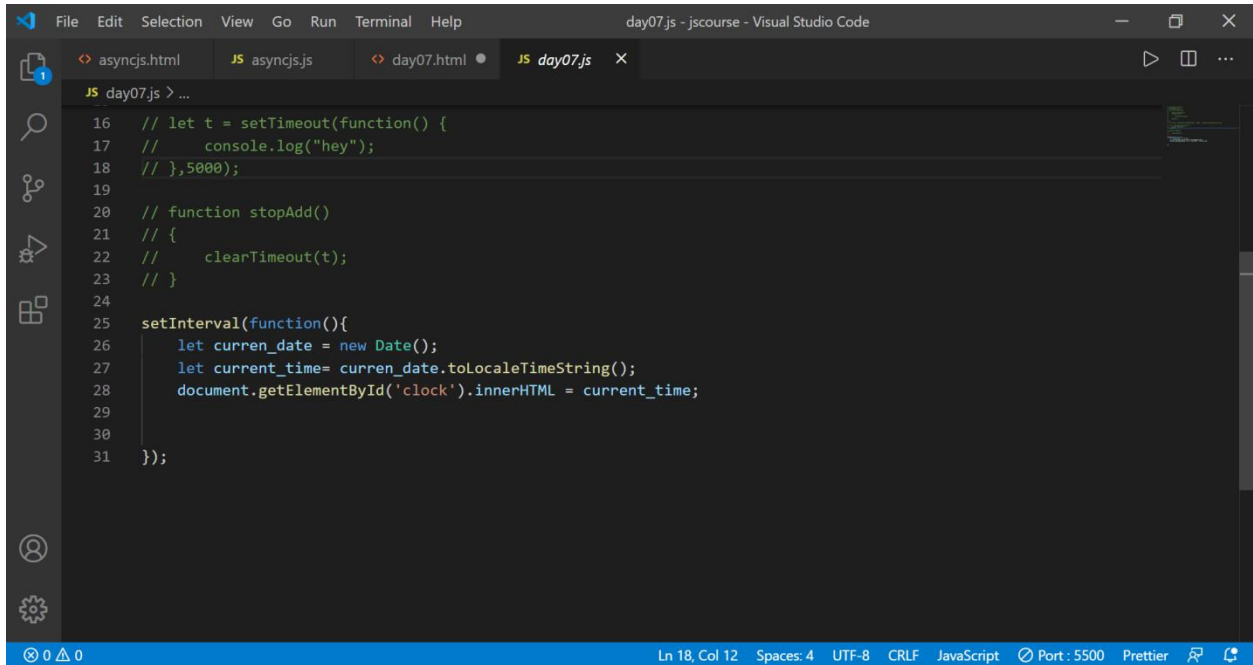
day07.js - jscourse - Visual Studio Code

File Edit Selection View Go Run Terminal Help

asyncjs.html JS asyncjs.js day07.html JS day07.js

```
JS day07.js > ...
1
2 // SetInterval function
3 // let counter =1;
4 // function doSomething()
5 // {
6 //     console.log(counter);
7 //     if(counter==10)
8 //     {
9 //         clearInterval(timer);
10 //     }
11 //     counter++;
12 // }
13 // //
14 // let timer = setInterval("doSomething()",1000); // 1000 milliseconds waiting time
15
16 // let t = setTimeout(function() {
17 //     console.log("hey");
18 // },5000);
19
20 // function stopAdd()
21 // {
22 //     clearTimeout(t);
23 // }
24
```

Ln 18, Col 12 Spaces: 4 UTF-8 CRLF JavaScript Port: 5500 Prettier

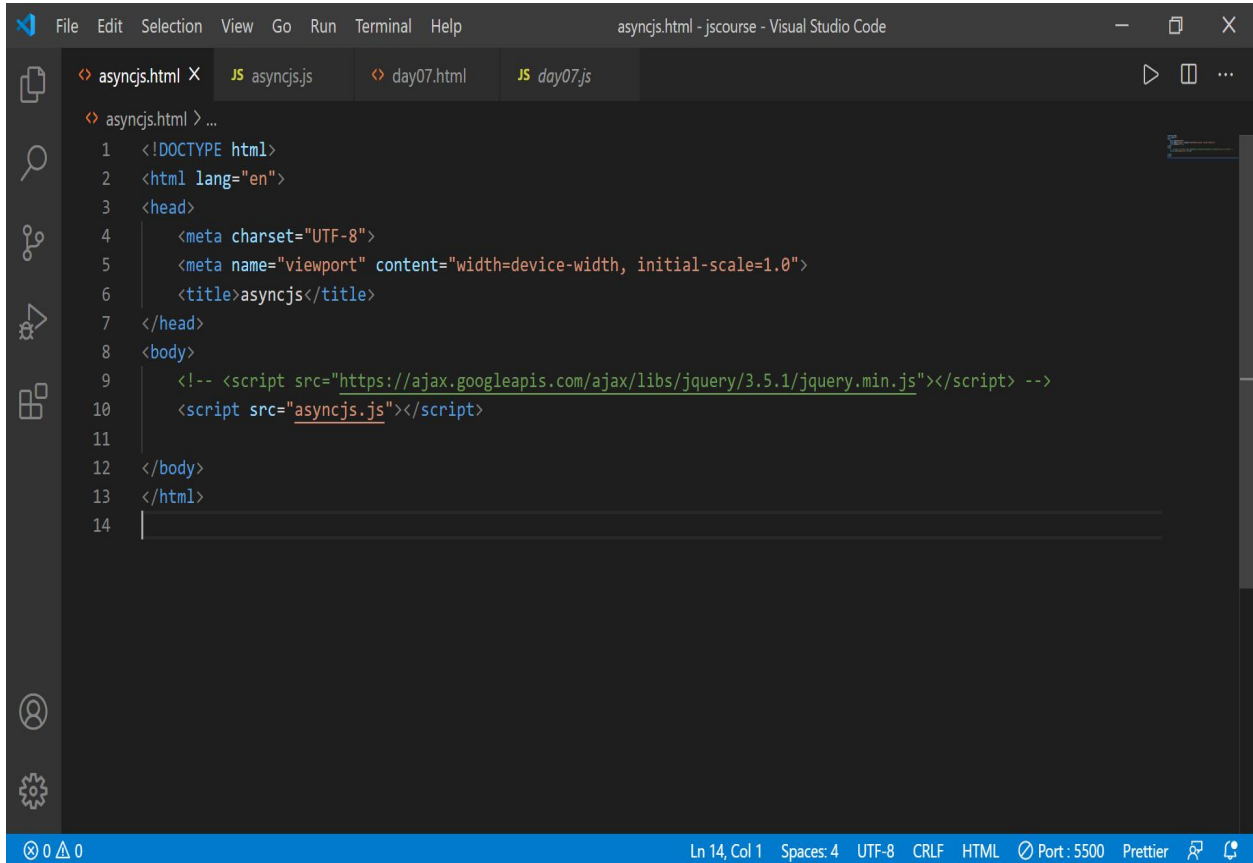


```
16 // let t = setTimeout(function() {
17 //     console.log("hey");
18 // }, 5000);
19
20 // function stopAdd()
21 // {
22 //     clearTimeout(t);
23 // }
24
25 setInterval(function(){
26     let current_date = new Date();
27     let current_time= current_date.toLocaleTimeString();
28     document.getElementById('clock').innerHTML = current_time;
29
30
31 });
```



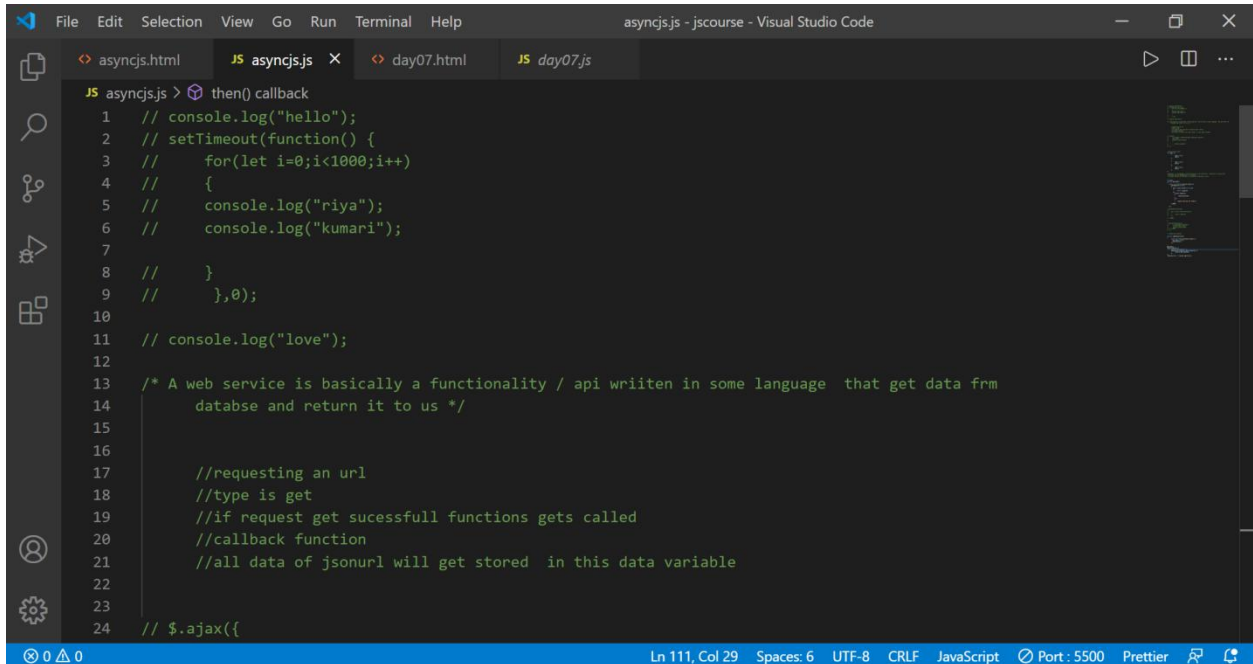
2:19:02 PM

HTML CODE2

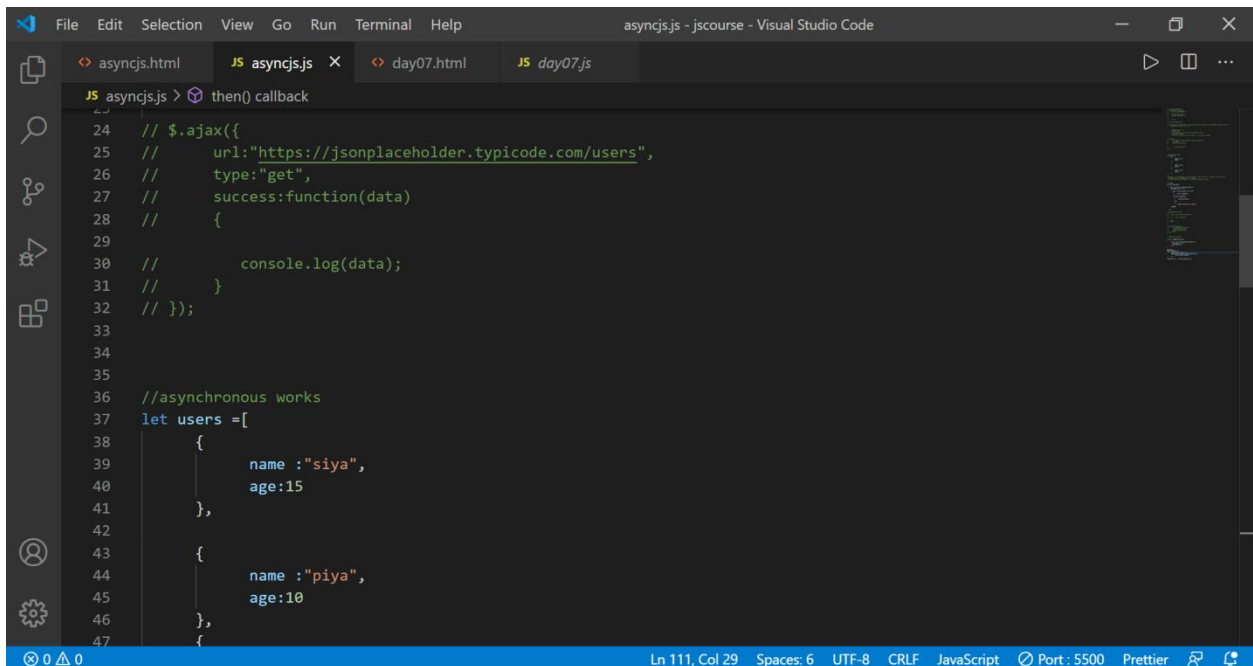


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>asyncjs</title>
7 </head>
8 <body>
9   <!-- <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script> -->
10  <script src="asyncjs.js"></script>
11
12 </body>
13 </html>
14
```

JAVASCRIPT CODE2



```
File Edit Selection View Go Run Terminal Help
asyncjs.js - jscourse - Visual Studio Code
asyncjs.html JS asyncjs.js day07.html JS day07.js
JS asyncjs.js > then() callback
1 // console.log("hello");
2 // setTimeout(function() {
3 //   for(let i=0;i<1000;i++)
4 //   {
5 //     console.log("riya");
6 //     console.log("kumari");
7
8 //   }
9 //   },0);
10
11 // console.log("love");
12
13 /* A web service is basically a functionality / api written in some language that get data frm
14    database and return it to us */
15
16
17 //requesting an url
18 //type is get
19 //if request get sucessfull functions gets called
20 //callback function
21 //all data of jsonurl will get stored in this data variable
22
23
24 // $.ajax({
```



```
File Edit Selection View Go Run Terminal Help
asyncjs.js - jscourse - Visual Studio Code
asyncjs.html JS asyncjs.js day07.html JS day07.js
JS asyncjs.js > then() callback
24 // $.ajax({
25 //   url:"https://jsonplaceholder.typicode.com/users",
26 //   type:"get",
27 //   success:function(data)
28 //   {
29
30 //     console.log(data);
31 //   }
32 // });
33
34
35
36 //asynchronous works
37 let users =[
38   {
39     name : "siya",
40     age:15
41   },
42
43   {
44     name : "piya",
45     age:10
46   },
47   {
```

```
47     {
48         name : "diya",
49         age: 25
50     },
51 ];
52
53 /*promises --> introduced in the es6 version of the javascript , basically to replace the
54 functions and the fundamentals of callbacks
55 it checks either the functions is succesfully executed or not*/
56
57
58 let user;
59 function get_data()
60 {
61     return new Promise((resolve, reject) => {
62         setTimeout(function()
63         {
64             user = users.filter(function(u)
65             {
66                 return u.age == 25;
67             });
68             if (user.length == 1)
69             {
70                 resolve(user[0]);
71             }
72         }, 3000);
73     });
74 }
```

```
69     {
70         resolve(user[0]);
71     }
72     else
73     {
74         reject("USER WAS NOT FOUND");
75     }
76 }, 3000);
77
78 });
79
80 // setTimeout(function()
81 // {
82 //     user = users.filter(function(u)
83 //     {
84 //         return u.age == 25;
85 //     });
86 // }, 3000);
87
88 }
89
90 // get_data(function(){
91 //     // setTimeout(function(){
92 //         // resolve(user[0]);
93 //     }, 3000);
94 // });
95 }
```

```
File Edit Selection View Go Run Terminal Help
asyncjs.js - jscourse - Visual Studio Code

asyncjs.html JS asyncjs.js X day07.html JS day07.js

JS asyncjs.js > then() callback
92 // // setTimeout(function(){
93 //   user[0].name = "diyu";
94 //   console.log(user[0]);
95 // }, 5000)
96 // });
97
98
99 // handling promises
100
101 function updateusers(user)
102 {
103   return new Promise((resolve, reject) => {
104     user.name = "cutie";
105     resolve(user);
106   });
107 }
108
109 get_data().
110 then((response) => {
111   console.log(response);
112   updateusers(response).then((response1) => {
113     console.log(response1)
114   })
115 })
116
```

Ln 111, Col 29 Spaces: 6 UTF-8 CRLF JavaScript Port: 5500 Prettier

```
File Edit Selection View Go Run Terminal Help
asyncjs.js - jscourse - Visual Studio Code

asyncjs.html JS asyncjs.js X day07.html JS day07.js

JS asyncjs.js > then() callback
104   user.name = "cutie";
105   resolve(user);
106 });
107 }
108
109 get_data().
110 then((response) => {
111   console.log(response);
112   updateusers(response).then((response1) => {
113     console.log(response1)
114   })
115 })
116 .catch((error) => console.log(error));
```

Ln 111, Col 29 Spaces: 6 UTF-8 CRLF JavaScript Port: 5500 Prettier

