

Lesson 05 Demo 03

Creating Various Events in Node.js

Objective: To create various events in Node.js for passing arguments to listeners, switching to asynchronous mode, emitting error events, handling events, and calling event listeners

Tools required: Node Package Manager and Visual Studio Code

Prerequisites: Basic Linux Commands, NPM commands, JavaScript events module

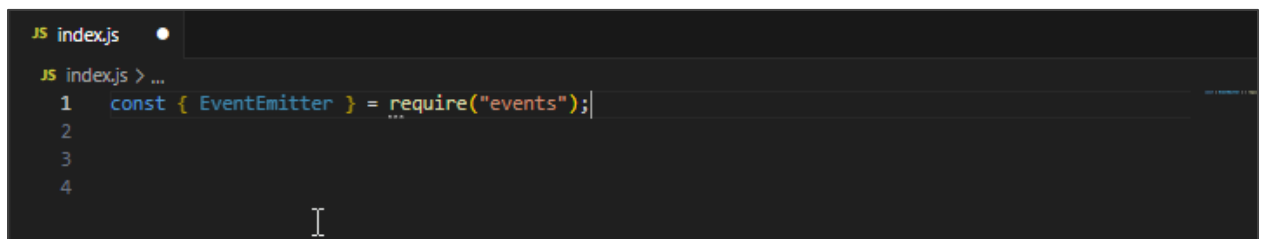
Steps to be followed:

1. Create a simple **EventEmitter** instance
2. Pass arguments to the listeners
3. Use the listener function to switch to asynchronous mode
4. Emit the error events
5. Handle the events
6. Call and register event listeners

Step 1: Create a simple EventEmitter instance

- 1.1 Create a file named index.js within the project directory and import the **events** module to create a simple EventEmitter instance:

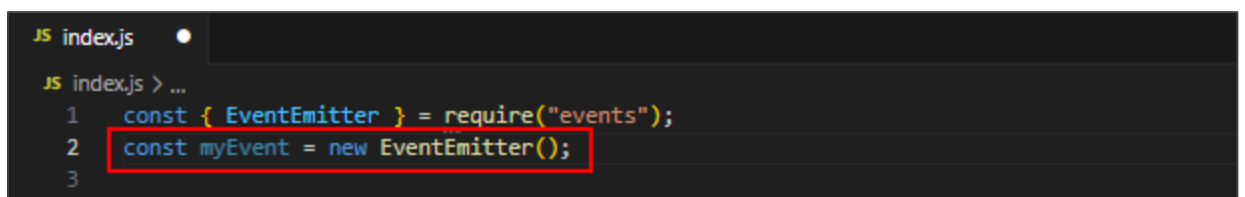
```
const { EventEmitter } = require("events");
```



```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3
4
```

- 1.2 Enter the following code statement to create an object of **EventEmitter** instance:

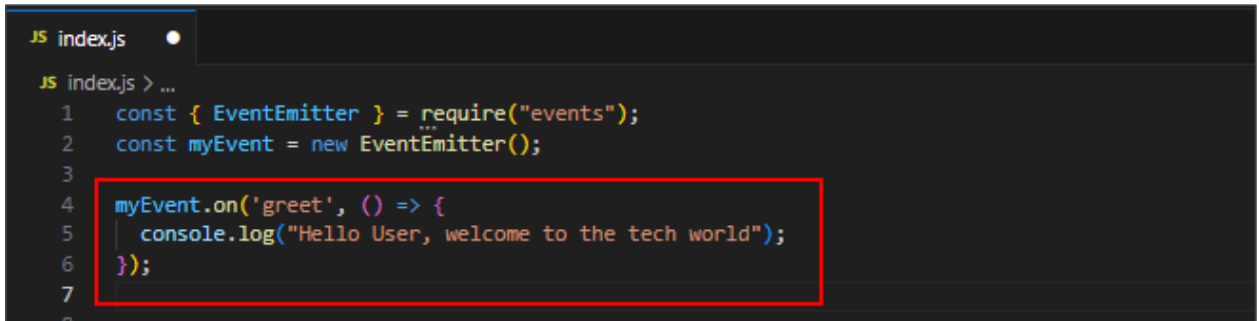
```
const myEvent = new EventEmitter();
```



```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2  const myEvent = new EventEmitter();
3
```

1.3 Use the **on** function to register the listener:

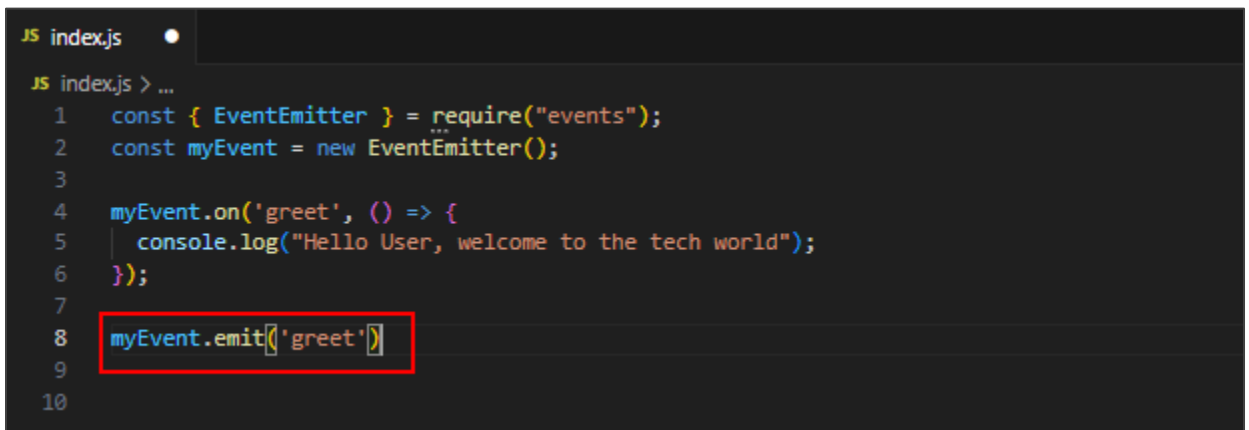
```
myEvent.on('greet', () => {  
  console.log("Hello User, welcome to the tech world");  
});
```

A screenshot of a code editor with a dark theme. The file is named 'index.js'. The code shows the setup of an EventEmitter and the registration of a listener for the 'greet' event. The listener function logs a welcome message. Lines 4 through 6 are highlighted with a red box.

```
JS index.js  
JS index.js > ...  
1  const { EventEmitter } = require("events");  
2  const myEvent = new EventEmitter();  
3  
4  myEvent.on('greet', () => {  
5    console.log("Hello User, welcome to the tech world");  
6  });  
7
```

1.4 Execute the emit function by passing the suitable event:

```
myEvent.emit('greet')
```

A screenshot of a code editor with a dark theme. The file is named 'index.js'. The code shows the setup of an EventEmitter and the registration of a listener for the 'greet' event. The listener function logs a welcome message. Line 8, which calls myEvent.emit('greet'), is highlighted with a red box.

```
JS index.js  
JS index.js > ...  
1  const { EventEmitter } = require("events");  
2  const myEvent = new EventEmitter();  
3  
4  myEvent.on('greet', () => {  
5    console.log("Hello User, welcome to the tech world");  
6  });  
7  
8  myEvent.emit('greet');  
9  
10
```

1.5 Go to the terminal, inside the project directory run the following command:

```
node index.js
```

A screenshot of a terminal window. The prompt shows the user is at a machine with IP 172-31-16-204, in the directory ~/Desktop/nodeProjec/demo4. The command 'node index.js' has been executed, and the output is 'Hello User, welcome to the tech world'.

```
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js  
Hello User, welcome to the tech world
```

Step 2: Pass arguments to the listener

2.1 Import the events module within the **index.js** file:

```
const { EventEmitter } = require("events");
```

A screenshot of a code editor window titled 'index.js'. The first line of code is `const { EventEmitter } = require("events");`. The cursor is positioned at the end of the line.

2.2 Create the object of **EventEmitter**:

```
const myEvent = new EventEmitter();
```

A screenshot of a code editor window titled 'index.js'. The first two lines of code are `const { EventEmitter } = require("events");` and `const myEvent = new EventEmitter();`. The second line is highlighted with a red box.

2.3 Use the **on** function to register the listener and create a function that takes **str1** and **str2** as two string inputs:

```
myEvent.on('greeting', function(str1, str2) {  
  console.log(str1, str2);  
  console.log(this)  
  console.log(this === myEvent);  
});
```

A screenshot of a code editor window titled 'index.js'. The code includes the previous lines plus a new listener function: `myEvent.on('greeting', function (str1, str2) { console.log(str1, str2); console.log(this); console.log(this === myEvent); });`. The entire function block is highlighted with a red box.

2.4 Execute the **emit** function by passing the suitable event and input strings
myEvent.emit('greeting', 'Fionna', 'Jack')

```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2  const myEvent = new EventEmitter();
3
4  myEvent.on('greeting', function (str1, str2) {
5    console.log(str1, str2);
6    console.log(this)
7    console.log(this === myEvent);
8  });
9
10 myEvent.emit('greeting', 'Fionna', 'Jack')
11
12
13
```

2.5 Go to the terminal, inside the project directory run the following command:
node index.js

```
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
Fionna Jack
EventEmitter {
  _events: [Object: null prototype] { greeting: [Function (anonymous)] },
  _eventsCount: 1,
  _maxListeners: undefined,
  [Symbol(kCapture)]: false
}
true
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
Fionna Jack
{}
false
```

Step 3: Use the listener function to switch to asynchronous mode

3.1 Import the `events` module within the `index.js` file:

```
const { EventEmitter } = require("events");
```

A screenshot of a VS Code editor window with a dark theme. The file name 'index.js' is shown in the top-left corner. The editor content shows the first line of code: `const { EventEmitter } = require("events");`. The cursor is positioned at the end of this line. Line numbers 1, 2, 3, and 4 are visible on the left margin.

3.2 Create the object of `EventEmitter` instance:

```
const myEvent = new EventEmitter();
```

A screenshot of a VS Code editor window. The second line of code, `const myEvent = new EventEmitter();`, is highlighted with a red rectangular box. The first line of code is also visible. The cursor is at the end of the second line. Line numbers 1, 2, 3, and 4 are visible on the left margin.

3.3 Use the `on` function to register the listener and create a function that takes `str` as string input:

```
myEvent.on('greet', (str) => {  
  setImmediate(() => {  
    console.log(">>> Hello", str);  
  })  
});
```

A screenshot of a VS Code editor window showing the complete code. The first two lines are `const { EventEmitter } = require("events");` and `const myEvent = new EventEmitter();`. The next block of code, which registers the listener, is highlighted with a red rectangular box. This block includes `myEvent.on('greeting', function (str1, str2) { ... });` and `myEvent.on('greet', (str) => { ... });`. The cursor is at the end of the second `on` function call. Line numbers 1 through 16 are visible on the left margin.

3.4 Execute the emit function by passing the suitable event and input strings:
`myEvent.emit('greet', 'Fionna')`

```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2  const myEvent = new EventEmitter();
3
4  myEvent.on('greeting', function (str1, str2) {
5    console.log(str1, str2);
6    console.log(this)
7    console.log(this === myEvent);
8  });
9
10 myEvent.on('greet', (str) => {
11   setImmediate(() => {
12     console.log(">>> Hello", str);
13   })
14 });
15
16 myEvent.emit('greet', 'Fionna')
17
```

3.5 Go to the terminal, inside the project directory run the following command:
`node index.js`

```
demopython1yopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
>>> Hello Fionna
```

Step 4: Emit the error events

4.1 Import the `events` module within the `index.js` file:
`const { EventEmitter } = require("events");`

```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3
4
```

4.2 Create the object of **EventEmitter** instance:

```
const myEvent = new EventEmitter();
```

```
JS index.js •
JS index.js > ...
1  const { EventEmitter } = require("events");
2  const myEvent = new EventEmitter();
3
```

4.3 Execute the **emit** function with a new error object

```
myEvent.emit('event', new Error('whoop!!'))
```

```
JS index.js •
JS index.js > ...
1  const { EventEmitter } = require("events");
2  const myEvent = new EventEmitter();
3
4  myEvent.emit('event', new Error('whoop!!'));
5
```

4.4 Go to the terminal, inside the project directory run the following command:

```
node index.js
```

```
>>> netto 110m
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$
```

4.5 Use the following code in the **index.js** file to add listeners for the error events:

```
const { EventEmitter } = require("events");

const myEvent = new EventEmitter();
myEvent.on('event', (error) => {
  console.log("Something went wrong!! ");
  console.error(error);
})

myEvent.emit('event', new Error("Sorry, bug appears :("))
```



```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3  const myEvent = new EventEmitter();
4  myEvent.on('event', (error) => {
5    console.log("Something went wrong!! ");
6    console.error(error);
7  })
8
9  myEvent.emit('event', new Error("Sorry, bug appears :("))
10
11
```

4.6 Run the Node.js app using the following command:

node index.js



```
demopython1yopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
Something went wrong!!
Error: Sorry, bug appears :(
    at Object.<anonymous> (/home/demopython1yopm/Desktop/nodeProjec/demo4/index.js:9:23)
    at Module._compile (node:internal/modules/cjs/loader:1165:14)
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1219:10)
    at Module.load (node:internal/modules/cjs/loader:1043:32)
    at Function.Module._load (node:internal/modules/cjs/loader:878:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)
    at node:internal/main/run_main_module:22:47
```


Step 5: Handle the events

5.1 Import the events module within the index.js file:

```
const { EventEmitter } = require("events");
```

A screenshot of a VS Code editor window with the file 'index.js' open. The editor shows the first line of code: `const { EventEmitter } = require("events");`. The cursor is positioned at the end of this line. Line numbers 1, 2, 3, and 4 are visible on the left margin.

5.2 Create the object of **EventEmitter** instance:

```
const myEvent = new EventEmitter();
```

A screenshot of a VS Code editor window showing the first two lines of code in 'index.js'. The first line is `const { EventEmitter } = require("events");` and the second line is `const myEvent = new EventEmitter();`. The second line is highlighted with a red rectangular box. Line numbers 1, 2, 3, and 4 are visible on the left margin.

5.3 Create a variable **eventCount** and initialize it to 0:

```
let eventCount = 0;
```

A screenshot of a VS Code editor window showing the first four lines of code in 'index.js'. The first line is `const { EventEmitter } = require("events");`, the second is `const myEvent = new EventEmitter();`, and the third is `let eventCount = 0;`. The third line is highlighted with a red rectangular box. Line numbers 1, 2, 3, and 4 are visible on the left margin.

5.4 Register the listener for the **greet** event where the function will increment the **eventCount** variable and print its latest value on the console:

```
myEvent.on('greet', () => {  
  eventCount++;  
  console.log("Hello User, Welcome to Tech World", eventCount);  
})
```

```
JS index.js > ...  
1  const { EventEmitter } = require("events");  
2  
3  const event = new EventEmitter();  
4  let eventCount = 0;  
5  
6  myEvent.on('greet', () => {  
7    eventCount++;  
8    console.log("Hello User, Welcome to Tech World", eventCount);  
9  })  
10
```

5.5 Trigger the event using the **emit** function twice:

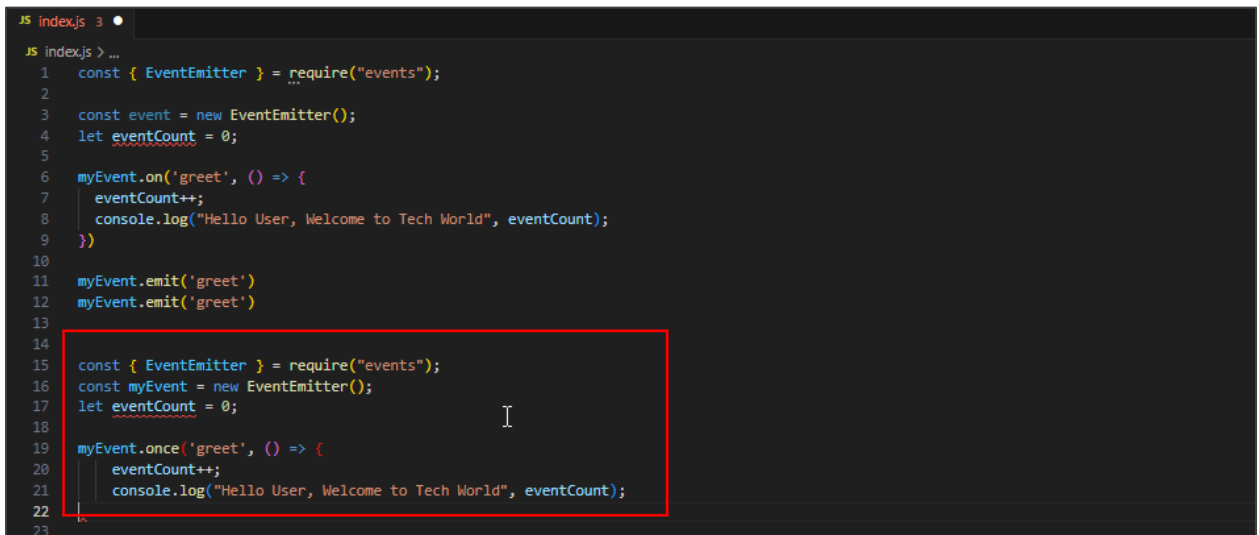
```
myEvent.emit('greet')  
myEvent.emit('greet')
```

```
JS index.js  
JS index.js > ...  
1  const { EventEmitter } = require("events");  
2  
3  const event = new EventEmitter();  
4  let eventCount = 0;  
5  
6  myEvent.on('greet', () => {  
7    eventCount++;  
8    console.log("Hello User, Welcome to Tech World", eventCount);  
9  })  
10  
11  myEvent.emit('greet')  
12  myEvent.emit('greet')  
13
```

5.6 Use the following code snippet to call the event using **eventEmitter.once()** method:

```
const { EventEmitter } = require("events");
const myEvent = new EventEmitter();
let eventCount = 0;

myEvent.once('greet', () => {
  eventCount++;
  console.log("Hello User, Welcome to Tech World", eventCount);
});
```



```
JS index.js 3
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3  const event = new EventEmitter();
4  let eventCount = 0;
5
6  myEvent.on('greet', () => {
7    eventCount++;
8    console.log("Hello User, Welcome to Tech World", eventCount);
9  })
10
11 myEvent.emit('greet')
12 myEvent.emit('greet')
13
14
15 const { EventEmitter } = require("events");
16 const myEvent = new EventEmitter();
17 let eventCount = 0;
18
19 myEvent.once('greet', () => {
20   eventCount++;
21   console.log("Hello User, Welcome to Tech World", eventCount);
22 })
23
```

5.7 Execute the command **node index.js** on the terminal window to view the output:

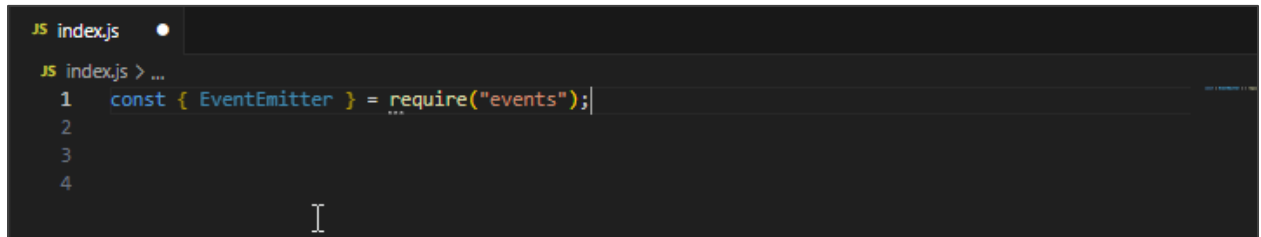


```
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
Hello User, Welcome to Tech World 1
Hello User, Welcome to Tech World 2
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
Hello User, Welcome to Tech World 1
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$
```

Step 6: Call and register event listeners

6.1 Import the **events** module within the **index.js** file:

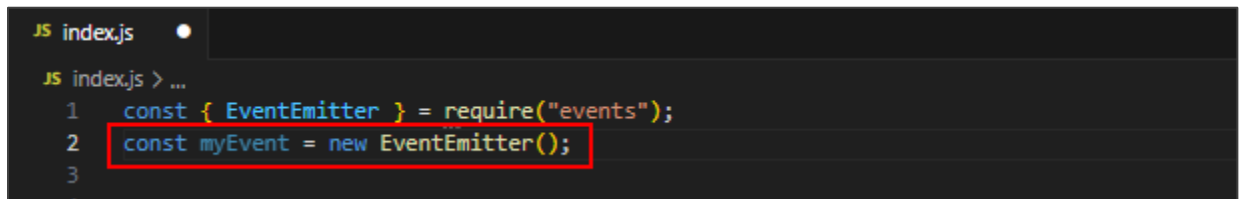
```
const { EventEmitter } = require("events");
```



```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3
4
```

6.2 Create the object of **EventEmitter**:

```
const myEvent = new EventEmitter();
```



```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2  const myEvent = new EventEmitter();
3
```

6.3 Create an **eventCount** variable and initialize it to 0:

```
let eventCount = 0;
```



```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3  const event = new EventEmitter();
4  let eventCount = 0;
```

6.4 Create an event named **greeting**:

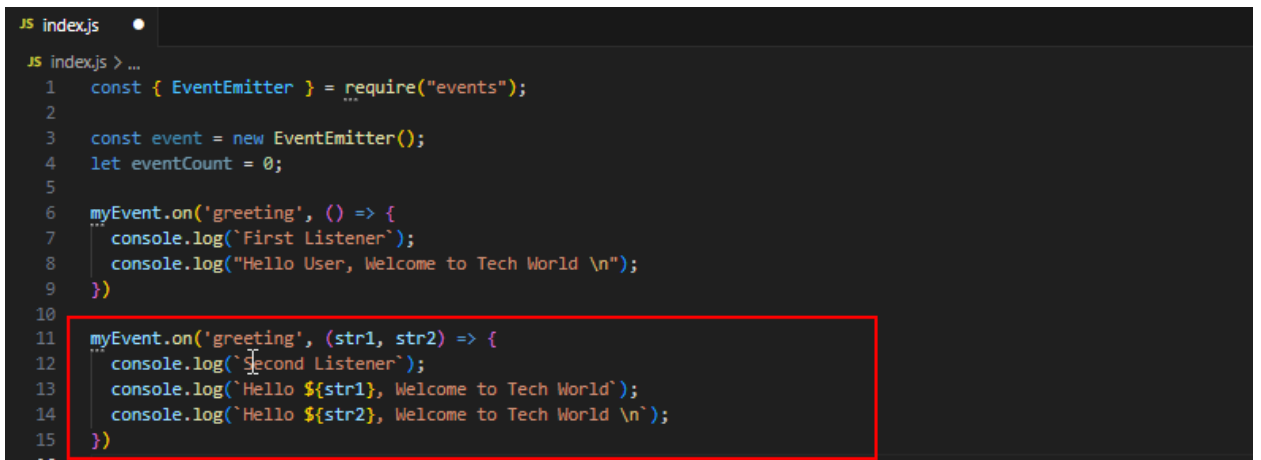
```
myEvent.on('greeting', () => {
  console.log(`First Listener`);
  console.log("Hello User, Welcome to Tech World \n");
})
```



```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3  const event = new EventEmitter();
4  let eventCount = 0;
5
6  myEvent.on('greeting', () => {
7    console.log(`First Listener`);
8    console.log("Hello User, Welcome to Tech World \n");
9  })
10
11
```

6.5 Create another event greeting with two string inputs:

```
myEvent.on('greeting', (str1, str2) => {
  console.log(`Second Listener`);
  console.log(`Hello ${str1}, Welcome to Tech World`);
  console.log(`Hello ${str2}, Welcome to Tech World \n`);
})
```



```
JS index.js
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3  const event = new EventEmitter();
4  let eventCount = 0;
5
6  myEvent.on('greeting', () => {
7    console.log(`First Listener`);
8    console.log("Hello User, Welcome to Tech World \n");
9  })
10
11  myEvent.on('greeting', (str1, str2) => {
12    console.log(`Second Listener`);
13    console.log(`Hello ${str1}, Welcome to Tech World`);
14    console.log(`Hello ${str2}, Welcome to Tech World \n`);
15  })
16
```

6.6 Trigger the events by passing different inputs:

```
myEvent.emit('greeting')
```

```
myEvent.emit('greeting', 'Fionna', 'Jack')
```

```
myEvent.emit('greeting', 'Fionna', 'Jack', 'John', 'Dave')
```

```
JS index.js •
JS index.js > ...
1  const { EventEmitter } = require("events");
2
3  const event = new EventEmitter();
4  let eventCount = 0;
5
6  myEvent.on('greeting', () => {
7    console.log('First Listener');
8    console.log("Hello User, Welcome to Tech World \n");
9  })
10
11 myEvent.on('greeting', (str1, str2) => {
12   console.log('Second Listener');
13   console.log(`Hello ${str1}, Welcome to Tech World`);
14   console.log(`Hello ${str2}, Welcome to Tech World \n`);
15 })
16
17 myEvent.emit('greeting')
18 myEvent.emit('greeting', 'Fionna', 'Jack')
19 myEvent.emit('greeting', 'Fionna', 'Jack', 'John', 'Dave')
20
21
```

6.7 Execute the command **node index.js** on the terminal window to view the output:

```
demopythonlyopm@ip-172-31-16-204:~/Desktop/nodeProjec/demo4$ node index.js
First Listener
Hello User, Welcome to Tech World

Second Listener
Hello Fionna, Welcome to Tech World
Hello Jack, Welcome to Tech World

Third Listener
Hello Fionna, Welcome to Tech World
Hello Jack, Welcome to Tech World
Hello John, Welcome to Tech World
Hello Dave, Welcome to Tech World
```

By following these steps, you have successfully created various events in Node.js for passing arguments to listeners, switching to asynchronous mode, emitting error events, handling events, and calling event listeners.