

# Event Arguments



**Mosh Hamedani**  
[programmingwithmosh.com](https://programmingwithmosh.com)

ActivitiesVisual Studio Code

Sun 15:38 •

0B/s ↓0B/s ↑🔴🟡🟢🔌📶🔊🗑️

app.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

> OPEN EDITORS

▼ APPLICATIONS

> 02. Global Object

> 03. Modules

> 04. Creating a Module

> 05. Loading a Module

> 06. Module Wrapper Function

> 07. Path Module

> 08. OS Module

> 09. File System Module

> 10. Events Module

> 11. Event Arguments

JS app.js

> 12. Extending Event Emitter

> 13. HTTP Module

> 14. Node Core Recap

> OUTLINE

11. Event Arguments > JS app.js > ...

```
1  const EventEmitter = require('events');
2  const emitter = new EventEmitter();
3
4  // Register a listener
5  emitter.on('messageLogged',function(arg1,arg2){// e, eventArg
6  console.log('Listener called', arg1, arg2);
7  });
8
9  // Raise an event
10 emitter.emit('messageLogged', 1, 'http://');
11 //emitter.emit('messageLogged', { id : 1,url : 'http://'});
12
```

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

6: bash

dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments\$ node app.js
Listener called 1 http://
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments\$

Ln 8, Col 1Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style🔍🔔

ActivitiesVisual Studio Code

Sun 15:390B/s ↓0B/s ↑

app.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

02. Global Object

03. Modules

04. Creating a Module

05. Loading a Module

06. Module Wrapper Function

07. Path Module

08. OS Module

09. File System Module

10. Events Module

11. Event Arguments

JS app.js

12. Extending Event Emitter

13. HTTP Module

14. Node Core Recap

11. Event Arguments > JS app.js > ...

```
1  const EventEmitter = require('events');
2  const emitter = new EventEmitter();
3
4  // // Register a listener
5  // emitter.on('messageLogged',function(arg1,arg2){// e, eventArg
6  // console.log('Listener called', arg1, arg2);
7  // });
8  // Raise an event
9  // emitter.emit('messageLogged', 1, 'http://');
10
11 // Register a listener
12 emitter.on('messageLogged',function(arg){// e, eventArg
13 console.log('Listener called', arg);
14 });
15 // Raise and event
16 emitter.emit('messageLogged', { id : 1,url : 'http://'});
17
```

DEBUG CONSOLEPROBLEMSOUTPUTTERMINAL

6: bash

dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments\$ node app.js
Listener called 1 http://
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments\$ node app.js
Listener called { id: 1, url: 'http://' }
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments\$

OUTLINE

master\* 0 0 javascript | app.jsLn 17, Col 2Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style

ActivitiesVisual Studio Code

Sun 15:430B/s ↓0B/s ↑

app.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

- 02. Global Object
- 03. Modules
- 04. Creating a Module
- 05. Loading a Module
- 06. Module Wrapper Function
- 07. Path Module
- 08. OS Module
- 09. File System Module
- 10. Events Module
- 11. Event Arguments
- JS app.js
- 12. Extending Event Emitter
- 13. HTTP Module
- 14. Node Core Recap

JS app.js 11. Event Arguments

JS app.js 08. OS Module

11. Event Arguments > JS app.js > ...

```
10
11 // // Register a listener
12 // emitter.on('messageLogged',function(arg){// e, eventArg
13 // console.log('Listener called', arg);
14 // });
15 // // Raise and event
16 // emitter.emit('messageLogged', { id : 1,url : 'http://'});
17
18 // Register a listener
19 // ECMAScript is a scripting-language specification standardized by Ecma International.
20 // It was created to standardize JavaScript to help foster multiple independent implementations.
21 emitter.on('messageLogged', (arg) => { // From ES6 we can write 'Arrow Function'
22 |   console.log('Listener called', arg);
23 | });
24 // Raise and event
25 emitter.emit('messageLogged', { id: 1, url: 'http://' });
```

DEBUG CONSOLE

PROBLEMS 1

OUTPUT

TERMINAL

6: bash

```
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments$ node app.js
Listener called 1 http://
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments$ node app.js
Listener called { id: 1, url: 'http://' }
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments$ node app.js
Listener called { id: 1, url: 'http://' }
dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments$
```

OUTLINE

master\*

0 0 0 1

✓ javascript | ✓ app.js

Ln 25, Col 58

Spaces: 4

UTF-8

LF

Babel JavaScript

JavaScript Standard Style

ActivitiesVisual Studio Code

Sun 16:02 •

0B/s ↓0B/s ↑

app.js - Applications - Visual Studio Code

FileEditSelectionViewGoRunTerminalHelp

EXPLORER

OPEN EDITORS

APPLICATIONS

02. Global Object

03. Modules

04. Creating a Module

05. Loading a Module

06. Module Wrapper Function

07. Path Module

08. OS Module

09. File System Module

10. Events Module

11. Event Arguments

JS app.js

12. Extending Event Emitter

13. HTTP Module

14. Node Core Recap

OUTLINE

JS app.js 11. Event Arguments ×JS app.js 08. OS Module

11. Event Arguments > JS app.js > ...

15 // // Raise and event

16 // emitter.emit('messageLogged', { id : 1,url : 'http://'});

17

18 // Register a listener

19 // ECMAScript is a scripting-language specification standardized by Ecma International.

20 // It was created to standardize JavaScript to help foster multiple independent implementations.

21 // emitter.on('messageLogged', (arg) => { // From ES6 we can write 'Arrow Function'

22 // console.log('Listener called', arg);

23 // });

24 // // Raise and event

25 // emitter.emit('messageLogged', { id: 1, url: 'http://' });

26

27 // Raise : logging (date : message)

28 emitter.on('logging', (arg) => {

29 console.log(`Output : \${arg.message}`);

30 console.log('Output :',arg);

31 });

32 emitter.emit('logging', {message : 'Work harder'});

DEBUG CONSOLEPROBLEMS 1OUTPUTTERMINAL

6: bash

dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments\$ node app.js

Output : Work harder

Output : { message: 'Work harder' }

dandy@My-Ubuntu:~/Keep/Node.js/02. Node Module System/Applications/11. Event Arguments\$

master\*0 0 0 1✓ javascript | ✓ app.js

Ln 32, Col 52Spaces: 4UTF-8LFBabel JavaScriptJavaScript Standard Style