

[Tutorials ▼](#)[Exercises ▼](#)[Services ▼](#)[Sign Up](#)[Log in](#)[CSS](#)[JAVASCRIPT](#)[SQL](#)[PYTHON](#)[JAVA](#)[PHP](#)[HOW TO](#)[W3.CSS](#)[C](#)

Node.js Events

[< Previous](#)[Next >](#)

Node.js is perfect for event-driven applications.

Events in Node.js

Every action on a computer is an event. Like when a connection is made or a file is opened.

Objects in Node.js can fire events, like the `readStream` object fires events when opening and closing a file:

Example

[Get your own Node.js Server](#)

```
var fs = require('fs');
var rs = fs.createReadStream('./demofile.txt');
rs.on('open', function () {
  console.log('The file is open');
});
```

[Run example »](#)

Events Module

[Tutorials](#)▼[Exercises](#)▼[Services](#)▼[Sign Up](#)[Log in](#)[CSS](#)[JAVASCRIPT](#)[SQL](#)[PYTHON](#)[JAVA](#)[PHP](#)[HOW TO](#)[W3.CSS](#)[C](#)

event properties and methods are an instance of an EventEmitter object. To be able to access these properties and methods, create an EventEmitter object:

```
var events = require('events');  
var EventEmitter = new events.EventEmitter();
```

The EventEmitter Object

You can assign event handlers to your own events with the EventEmitter object.

In the example below we have created a function that will be executed when a "scream" event is fired.

To fire an event, use the `emit()` method.

Example

```
var events = require('events');  
var EventEmitter = new events.EventEmitter();  
  
//Create an event handler:  
var myEventHandler = function () {  
  console.log('I hear a scream!');  
}  
  
//Assign the event handler to an event:  
eventEmitter.on('scream', myEventHandler);  
  
//Fire the 'scream' event:  
eventEmitter.emit('scream');
```

[Run example »](#)