Node.js
Events

Events

- Core of Node.js->asynchronous programming

- In some cases we have to do something when something happens.. Ie pass data around the app when that data is obtained…

- Events allow us to do so…

- Every action on a computer can be characterized as an event
  - ➢When a file opens, when a connection is established…

Events

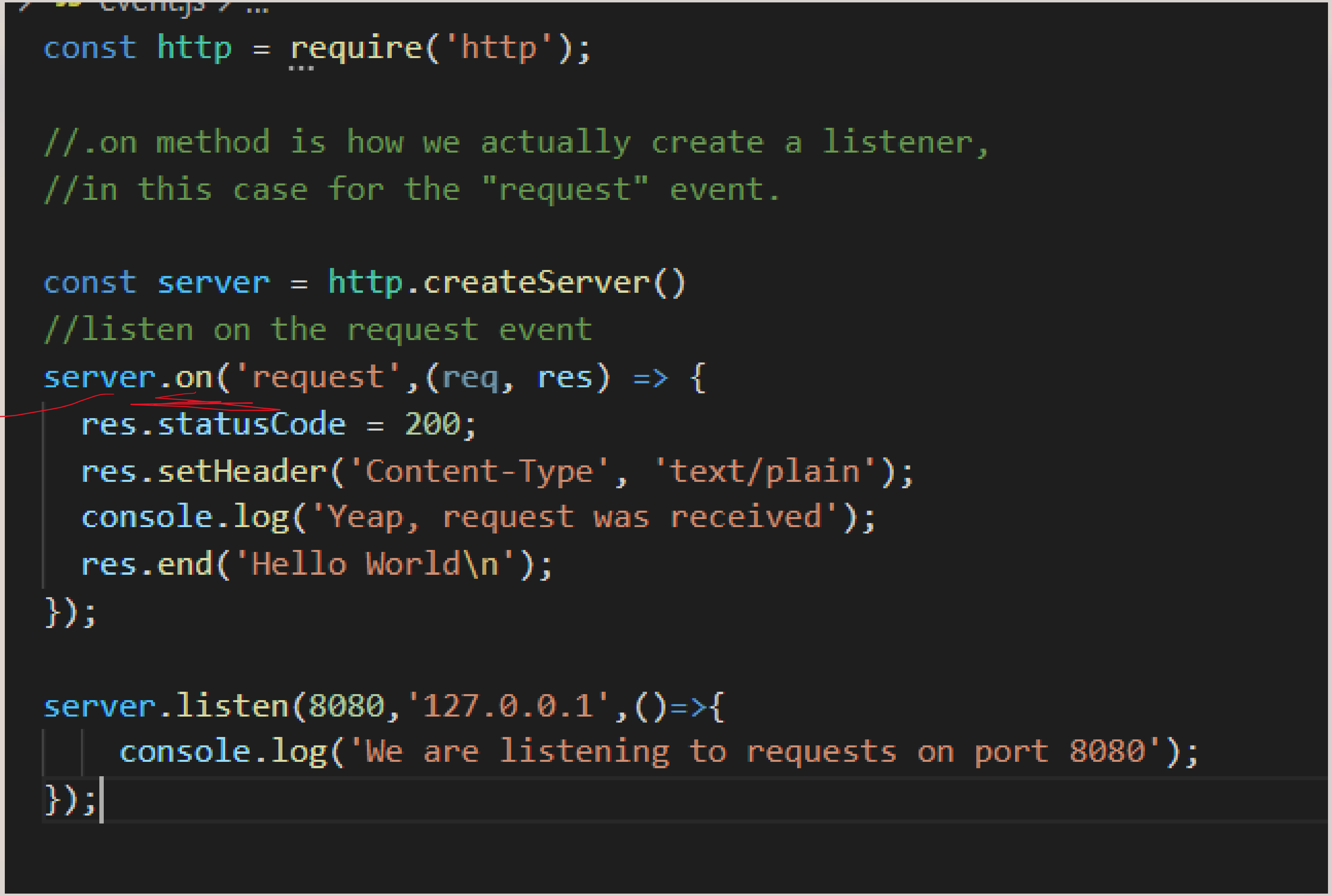
- Objects in Node.js can fire events, we can “listen” to these events

- For example receiving an **HTTP request** on our server or a **file finishing** to **read**, all these will **emit events** &event loop will then pick up these events

Events

- So Emitting, listening to, and handling events in a Node.js app is possible

- 2 types of events in Node JS
    - ➢build-in events
    - ➢custom events



```
> event.js > ...
const http = require('http');

//.on method is how we actually create a listener,
//in this case for the "request" event.

const server = http.createServer()
//listen on the request event
server.on('request',(req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  console.log('Yeap, request was received');
  res.end('Hello World\n');
});

server.listen(8080,'127.0.0.1',()=>{
    console.log('We are listening to requests on port 8080');
});
```

Create a listener for request event

Built in node module functions-> many times emit their own events-> all we have to do is to listen to them.

Check events for http module
https://nodejs.org/api/http.html#http_class_http_clientrequest

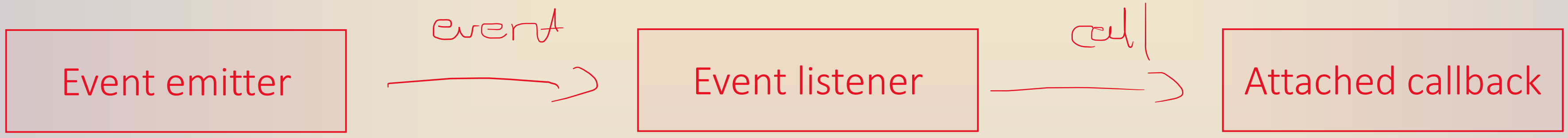
events module

- *events* module -> allows us to **create & handle *custom* events** in Node.js.

- Event emitters (Nodejs objects, instances of the **EventEmitter** class)-> emit events when something important happens in the app(ie request hitting server)
  - ➢event listeners pick these events
  - ➢ listeners trigger callbacks attached to them

- If there are more than one listener for an event, they run synchronously

events module



Event emitter

Event listener

Attached callback

- If there are more than one listeners for an event, they run synchronously

events module

- Node.js has a built-in module: "Events", that allows as to create-> fire-> listen for- our own events

- Syntax

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

- All event **properties and methods** are an **instance of an EventEmitter object**.
- Thus,
  1) We *require* **events module**
  2) We create an **EventEmitter object:** to access event properties and methods

events module

Listening events syntax

eventEmitter.addListener(event, listener)
eventEmitter.on(event, listener)

} pretty much the same

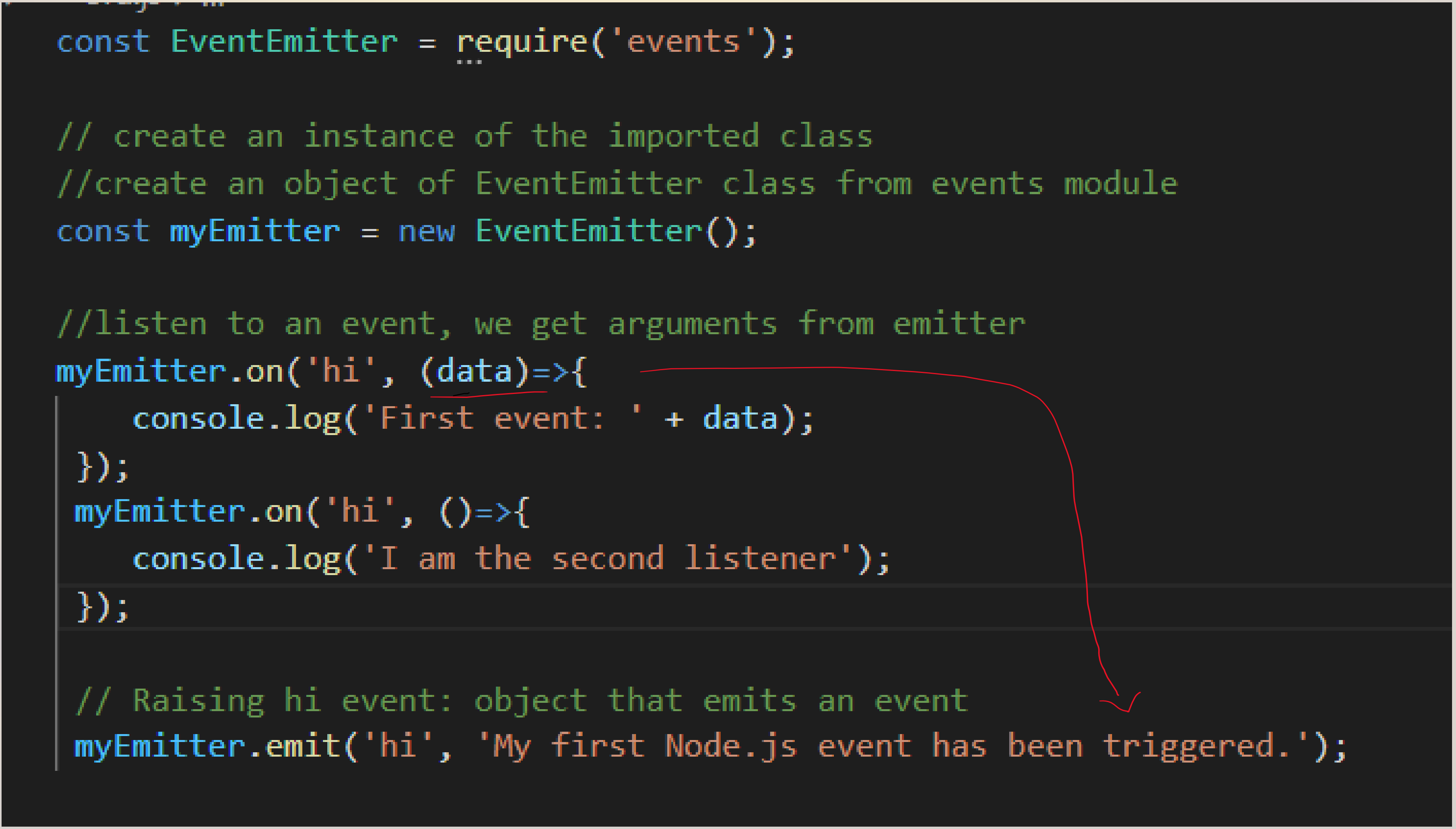
- Emitting events Syntax:

eventEmitter.emit(event, arg1, arg2, ...)

events module

- **Event listener code** is a **callback** function that takes a parameter for the data and handles it

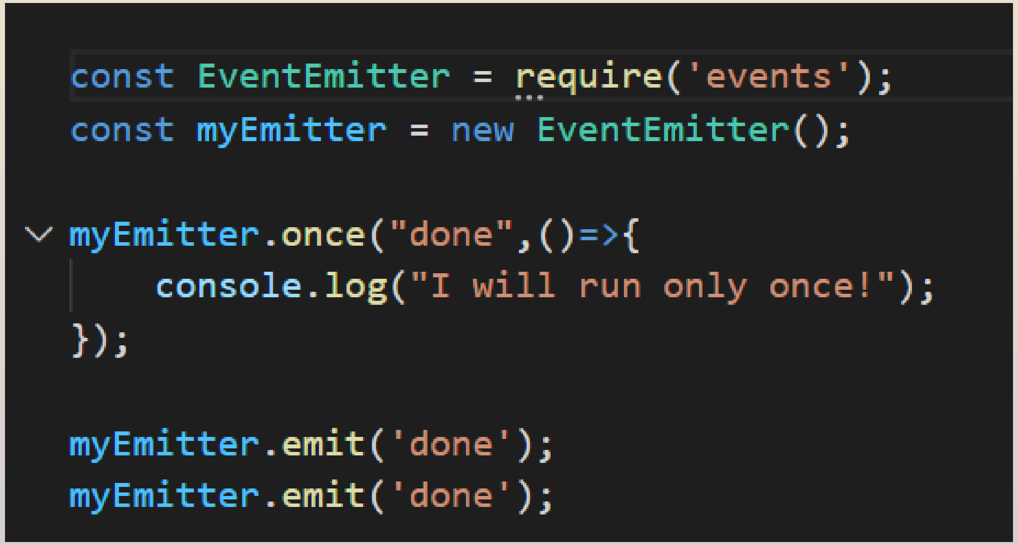
- An argument passed in the **event** is shared between all listeners.



```
const EventEmitter = require('events');

// create an instance of the imported class
//create an object of EventEmitter class from events module
const myEmitter = new EventEmitter();

//listen to an event, we get arguments from emitter
myEmitter.on('hi', (data)=>{
    console.log('First event: ' + data);
 });
 myEmitter.on('hi', ()=>{
    console.log('I am the second listener');
 });

 // Raising hi event: object that emits an event
 myEmitter.emit('hi', 'My first Node.js event has been triggered.');
```

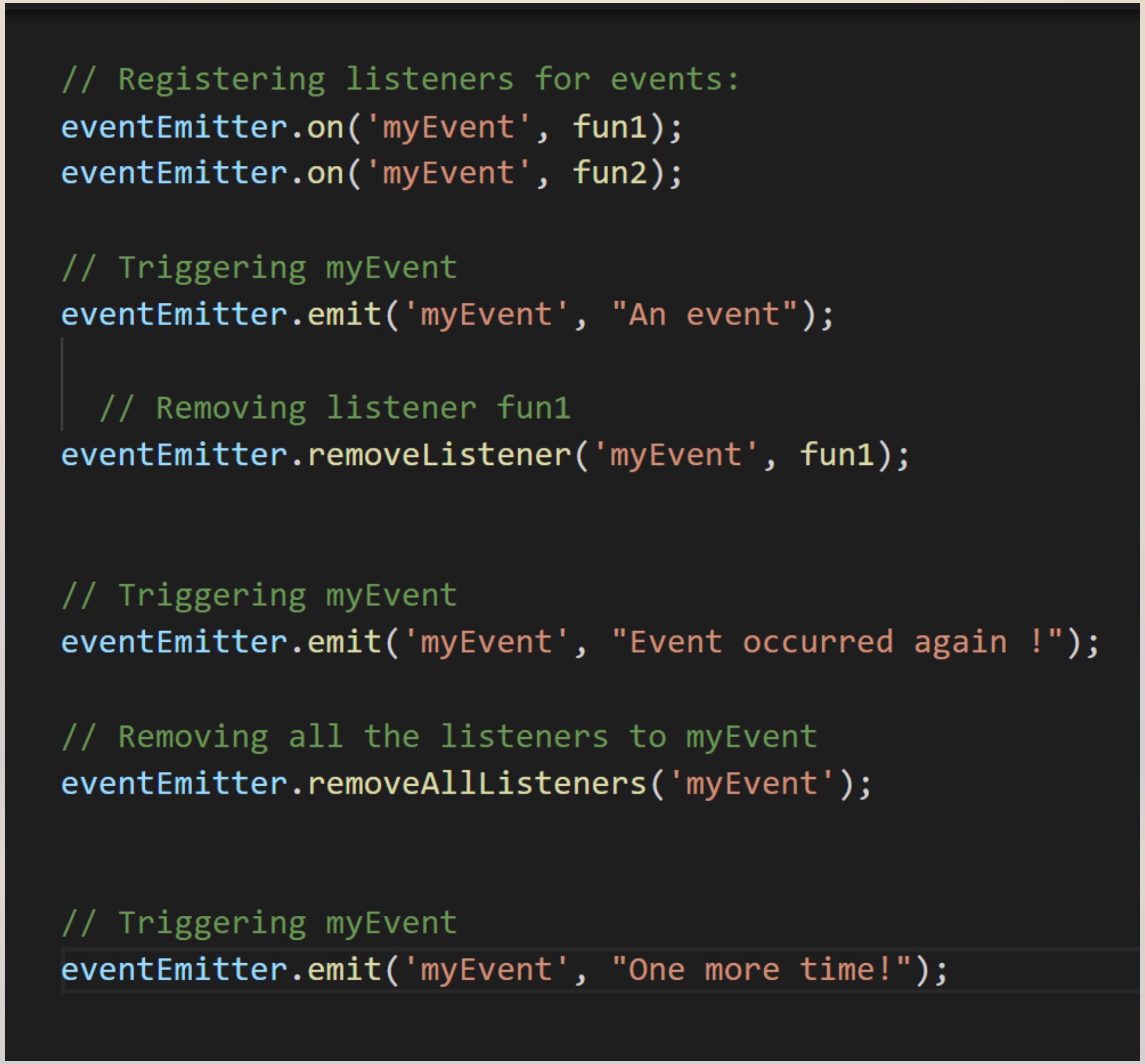
Some EventEmitter methods	Description
on(event, listener)	It can also be called as an alias of emitter.addListener()
once(event, listener)	Adds an one-time listener for event .
emit(event, [arg1], [arg2], [...])	Raise specified events with the supplied arguments.
removeListener(event, listener)	Removes a listener from the listener array for the specified event
removeAllListeners([event])	Removes all listeners, or those of specified event.

What do we expect to see below?



```
const EventEmitter = require('events');
const myEmitter = new EventEmitter();

myEmitter.once("done",()=>{
    console.log("I will run only once!");
});

myEmitter.emit('done');
myEmitter.emit('done');
```

What do we expect to see below?



```
// Registering listeners for events:
eventEmitter.on('myEvent', fun1);
eventEmitter.on('myEvent', fun2);

// Triggering myEvent
eventEmitter.emit('myEvent', "An event");

  // Removing listener fun1
eventEmitter.removeListener('myEvent', fun1);


// Triggering myEvent
eventEmitter.emit('myEvent', "Event occurred again !");

// Removing all the listeners to myEvent
eventEmitter.removeAllListeners('myEvent');


// Triggering myEvent
eventEmitter.emit('myEvent', "One more time!");
```

To be continued…

- Check https://nodejs.org/api/events.html
- https://nodejs.org/api/events.html#class-eventemitter
- https://codeburst.io/basics-of-events-streams-and-pipe-in-node-js-b84578c2f1be
- https://www.digitalocean.com/community/tutorials/how-to-work-with-files-using-streams-in-node-js