The Node Event emitter

node js have a bunch of inbuilt code modules which we can use in our applications

One of that code modules in node js is called the "events module".

events

when we require a code module, we take module name not path, because it is a code module built into node js, node will recognize that and it will go out & fetch that module for us.

```
* events
var stuff = require('events')
```

code module

whatever returned on the module exports of this module is going to be added in this events variable

One of the things returned on that module.exports property in this module is event emitter.

event emitter (var events = require('events'))
↓
custom event

we can use event emitter in nodejs to create custom events & then react to those events when they are emitted.

↗ indicates emits the events

myEmitter.on('someEvent', function(msg))
eventemitter is used to trigger an event .on is used to add a call back function that's going to be executed when the event is triggered.

③ util is another core module which allows us to do various things.
one of the things that the util module allows us to inherit certain things from objects built into nodejs or other objects.

forEach method

is used to call function for each element in an array.

Reading & writing files (fs)

To read & write files on our computer we need one of the nodejs core modules that module is called FS.

FS module is another core module

- fs.readFileSync() is a method that is used to go out and read the file

fs.readFileSync('readMe.txt', 'utf8');

(filename; 'utf8' is used to convert binary into text format)

(Synchronous)

reading

```
var fs = require('fs');
var readMe = fs.readFileSync('readMe.txt', 'utf8');
console.log(readMe);
```

Synchronous means after completing fully this operation it will go down

readMe.txt

Yay, you read me!

Reading & writting:

```
val fs = require('fs');

val readMe = ______ same as above

fs.writeFileSync('writeMe.txt', readMe);
```

when we run this.

writeMe.txt file will be created & data in readMe file will be in the writeMe.txt file.

```
fs.readFile('readMe.txt', 'utf8', function (err, data) {
    console.log(data);
}
```

(err: gives err if any problem in ...; data: ready data)

(*) Call back function when we use asynchronous method, to fire when the process is completed.

Asynchronous (mean first do below operations & come to this method and completes)

```
val fs = require('fs');
fs.readFile('readMe.txt', 'utf8', function(err, data)
{
    fs.writeFile('writeMe.txt', data);
});
```

(x) console.log('text');

04/, bet

Yay, you read me.