node

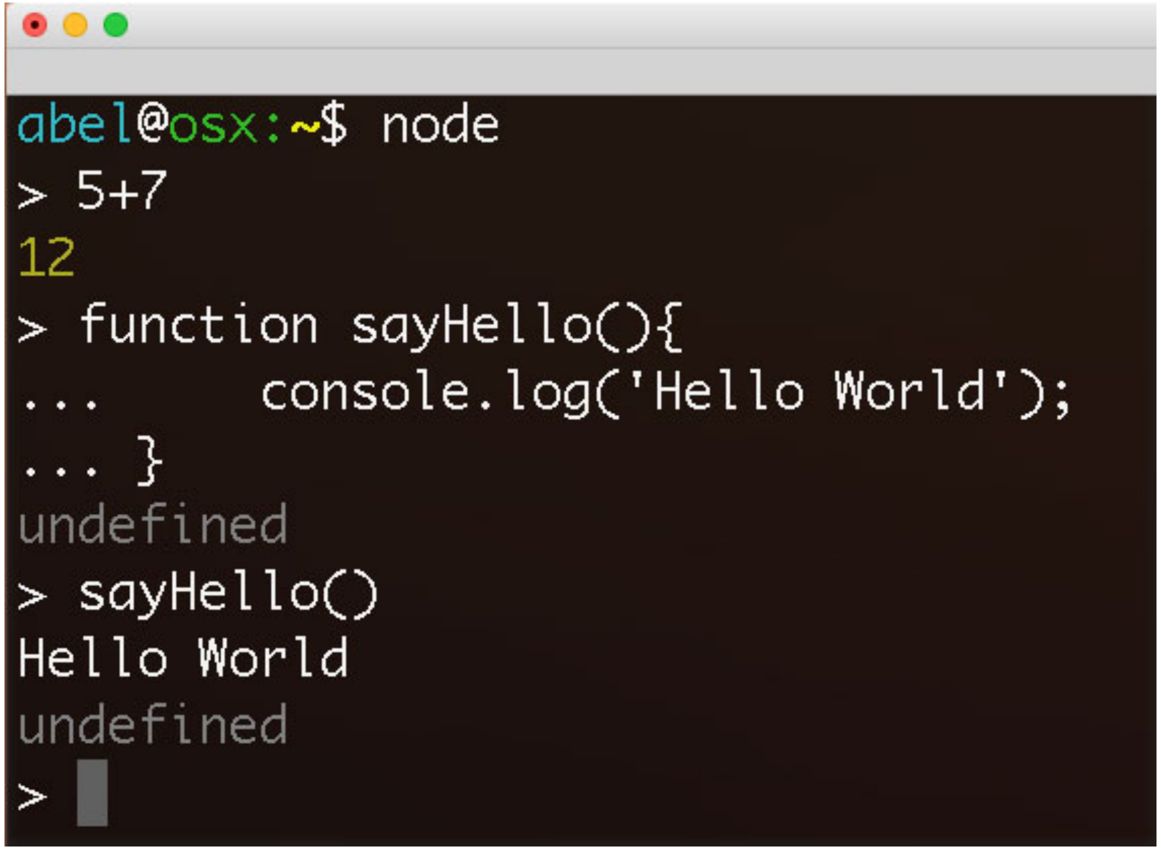
JS®

Node's goal is to provide an easy way to build scalable network programs

Node.JS

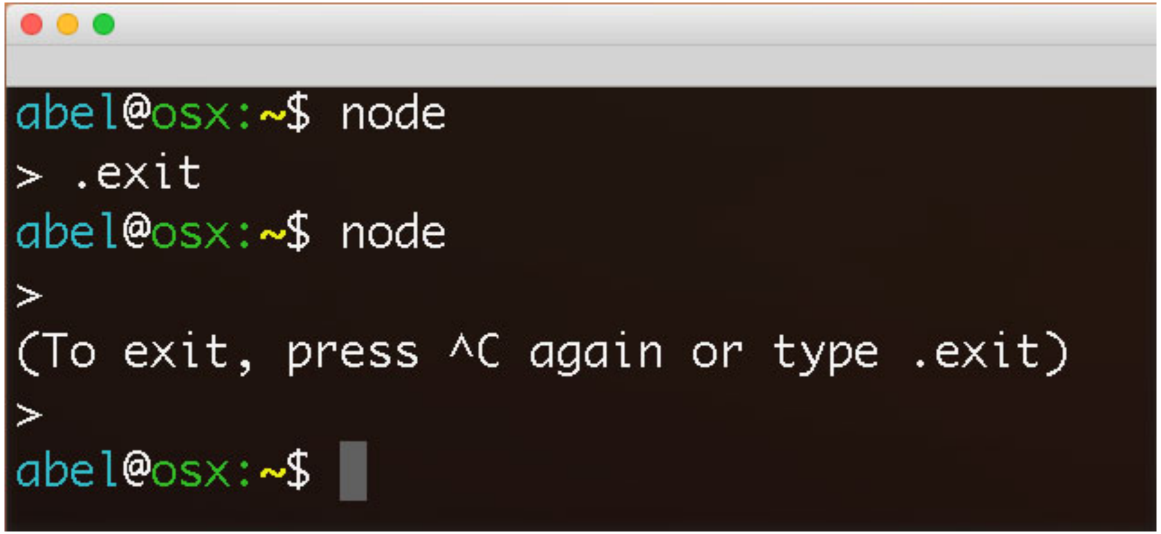
Node.js is an open source, cross-platform, JavaScript runtime built on Chrome's V8 JavaScript engine, for developing server and client side applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

Console Use



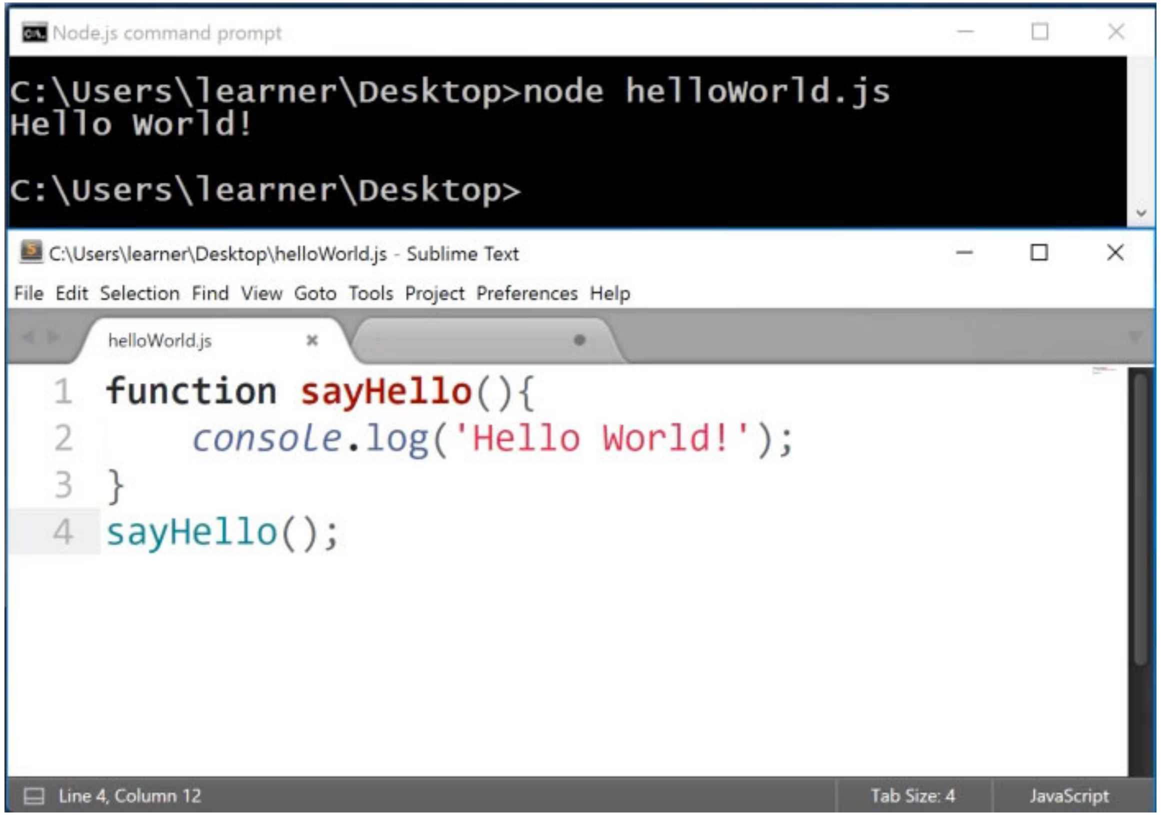
```
abel@osx:~$ node
> 5+7
12
> function sayHello(){
...       console.log('Hello World');
... }
undefined
> sayHello()
Hello World
undefined
> 
```

Exit Console



```
abel@osx:~$ node
> .exit
abel@osx:~$ node
>
(To exit, press ^C again or type .exit)
>
abel@osx:~$
```

Run File

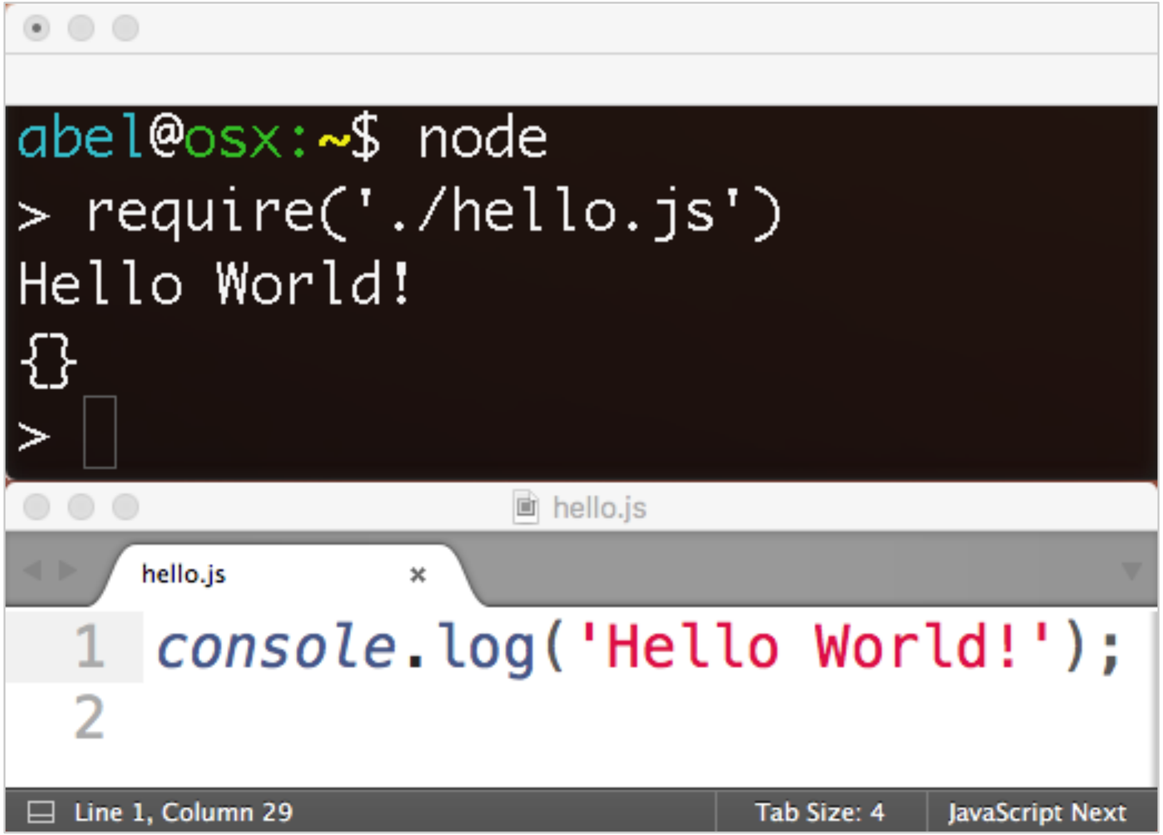

Node.js command prompt
C:\Users\learner\Desktop\helloWorld.js - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
helloWorld.js
Line 4, Column 12
Tab Size: 4
JavaScript

```
C:\Users\learner\Desktop>node helloWorld.js
Hello World!

C:\Users\learner\Desktop>
```

```javascript
function sayHello(){
    console.log('Hello World!');
}
sayHello();
```

Load File



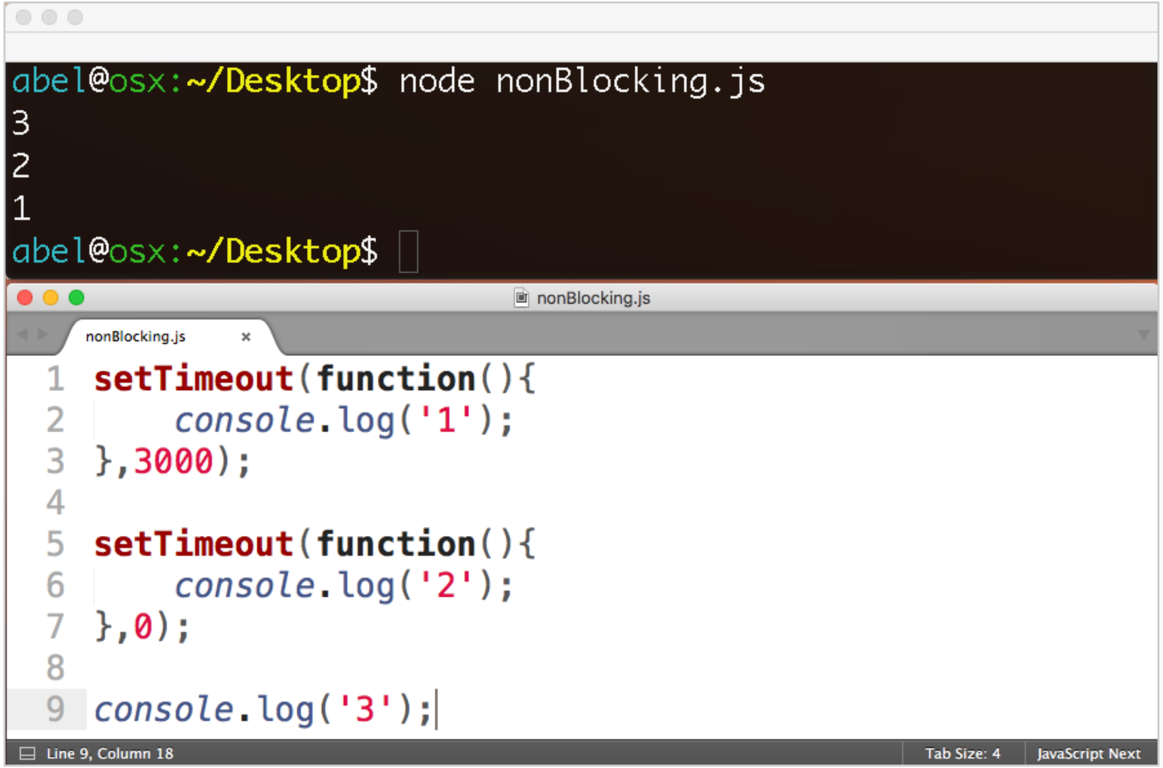
```
abel@osx:~$ node
> require('./hello.js')
Hello World!
{}
> 
```

hello.js

hello.js

```
console.log('Hello World!');
```

Non-Blocking



```
abel@osx:~/Desktop$ node nonBlocking.js
3
2
1
abel@osx:~/Desktop$
```

nonBlocking.js

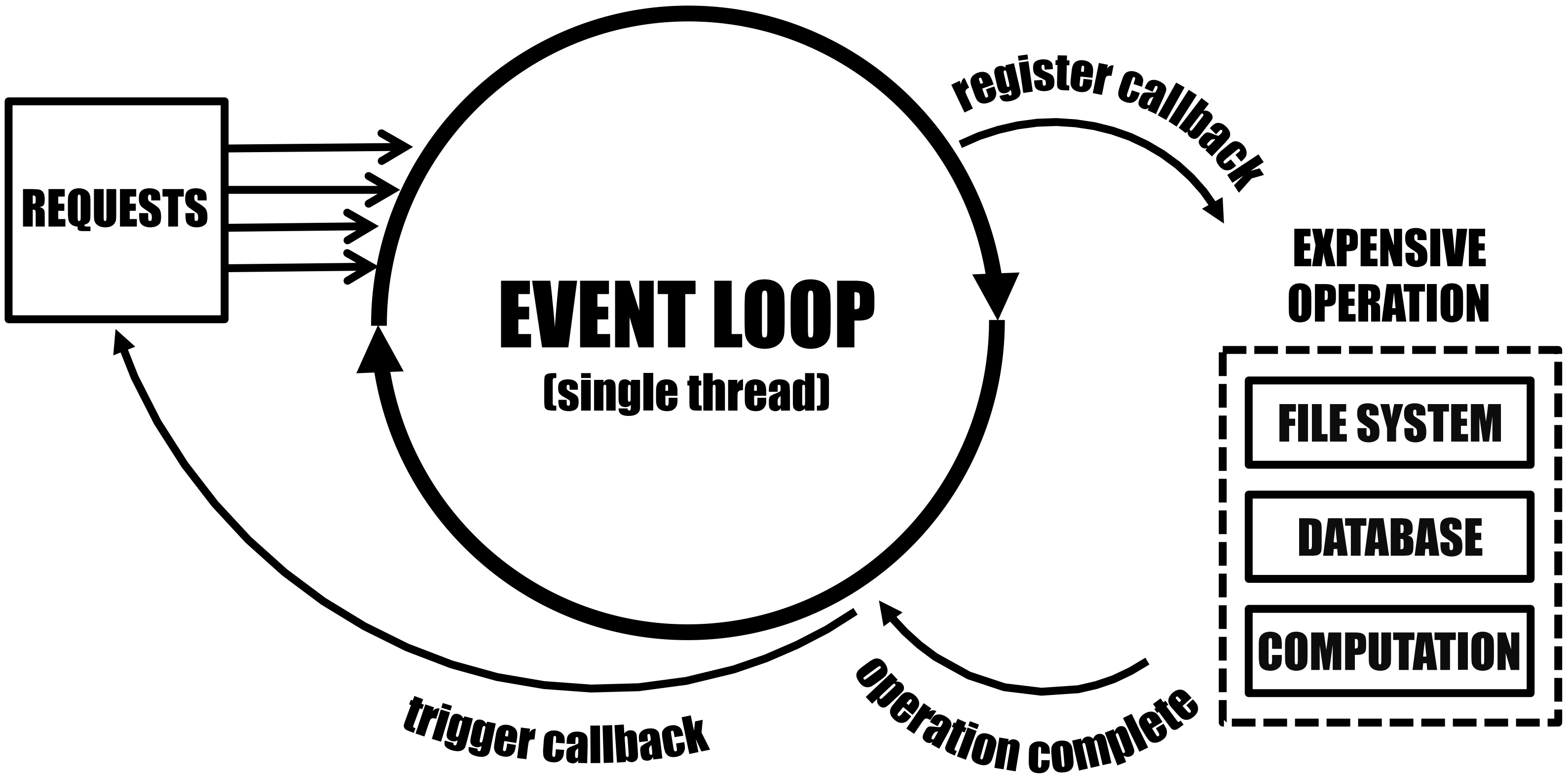
```
setTimeout(function(){
    console.log('1');
},3000);

setTimeout(function(){
    console.log('2');
},0);

console.log('3');
```

Line 9, Column 18 Tab Size: 4 JavaScript Next

Coca-Cola


REQUESTS
EVENT LOOP
(single thread)
register callback
EXPENSIVE OPERATION
FILE SYSTEM
DATABASE
COMPUTATION
operation complete
trigger callback

Architectural Shift

1.old) web server with some application logic
1.new) app that can connect and collaborate

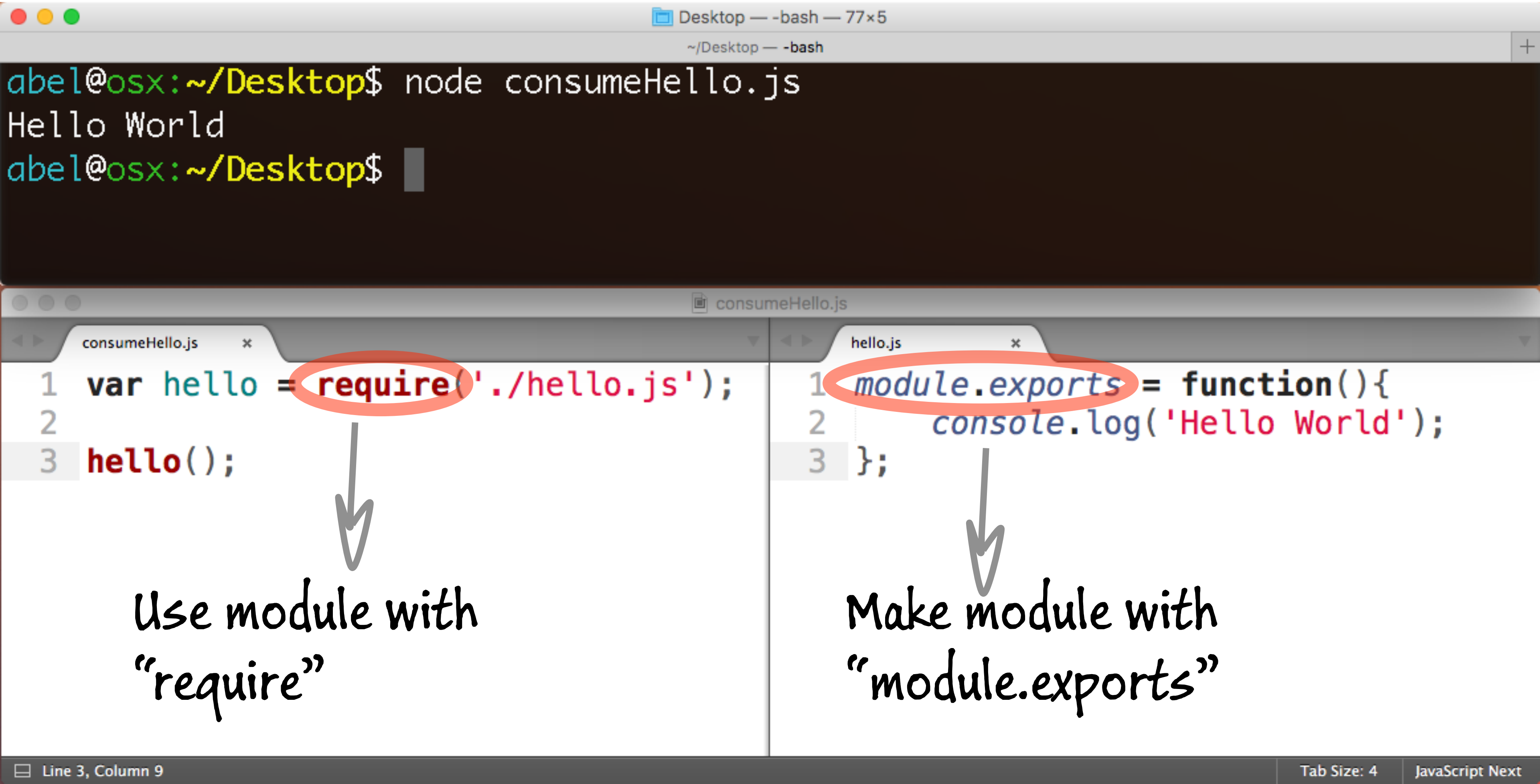
2.old) stateful
2.new) stateless

Architectural Shift

1.old) blocking
1.new) non-blocking

2.old) process per request
2.new) single process

MODULES



Desktop — -bash — 77×5

~/Desktop — -bash

```
abel@osx:~/Desktop$ node consumeHello.js
Hello World
abel@osx:~/Desktop$
```

consumeHello.js

consumeHello.js

```
var hello = require('./hello.js');

hello();
```

hello.js

```
module.exports = function(){
    console.log('Hello World');
};
```

Use module with "require"

Make module with "module.exports"

Line 3, Column 9 Tab Size: 4 JavaScript Next

What is npm?

Package manager. Installs, publishes and manages node programs

https://www.npmjs.com

What is npm?

400,000+ Modules

What is npm?

261 Million

downloads in the past day

What is npm?

1.9 Billion

downloads in the past week

What is npm?

7.8 Billion

downloads per month

What is a Module?

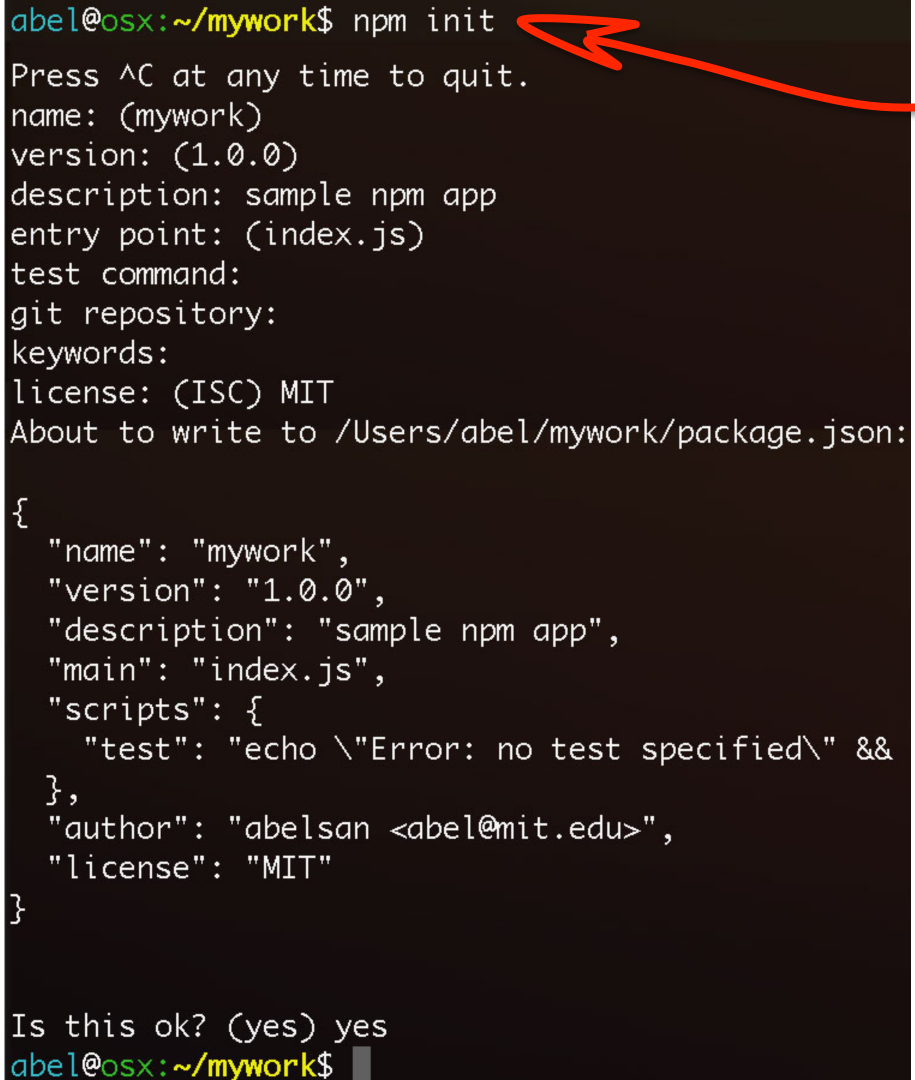
A Module is some JavaScript paired with a package.json file

JS + JSON

Advantages

- Small pieces, loosely joined
- Leverage external packages
- Leverage internal packages
- Facilitate collaboration
- Packages are discoverable in npm

Creating node/npm apps

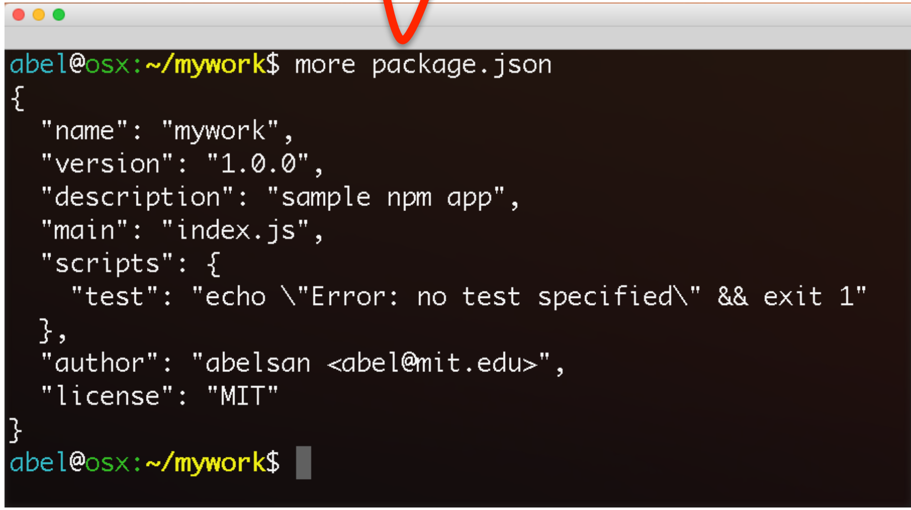


```
abel@osx:~/mywork$ npm init
Press ^C at any time to quit.
name: (mywork)
version: (1.0.0)
description: sample npm app
entry point: (index.js)
test command:
git repository:
keywords:
license: (ISC) MIT
About to write to /Users/abel/mywork/package.json:

{
  "name": "mywork",
  "version": "1.0.0",
  "description": "sample npm app",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" &&
  },
  "author": "abelsan <abel@mit.edu>",
  "license": "MIT"
}


Is this ok? (yes) yes
abel@osx:~/mywork$
```

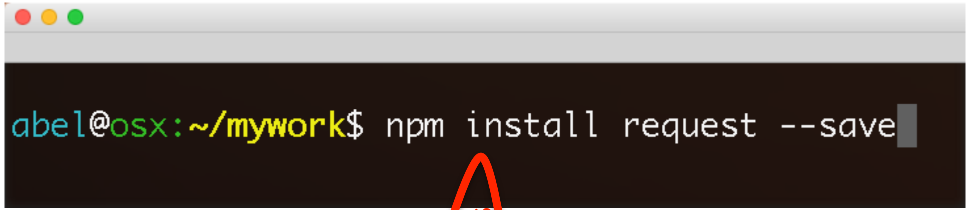
"npm init" generates the configuration file package.json



```
abel@osx:~/mywork$ more package.json
{
  "name": "mywork",
  "version": "1.0.0",
  "description": "sample npm app",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "abelsan <abel@mit.edu>",
  "license": "MIT"
}
abel@osx:~/mywork$
```

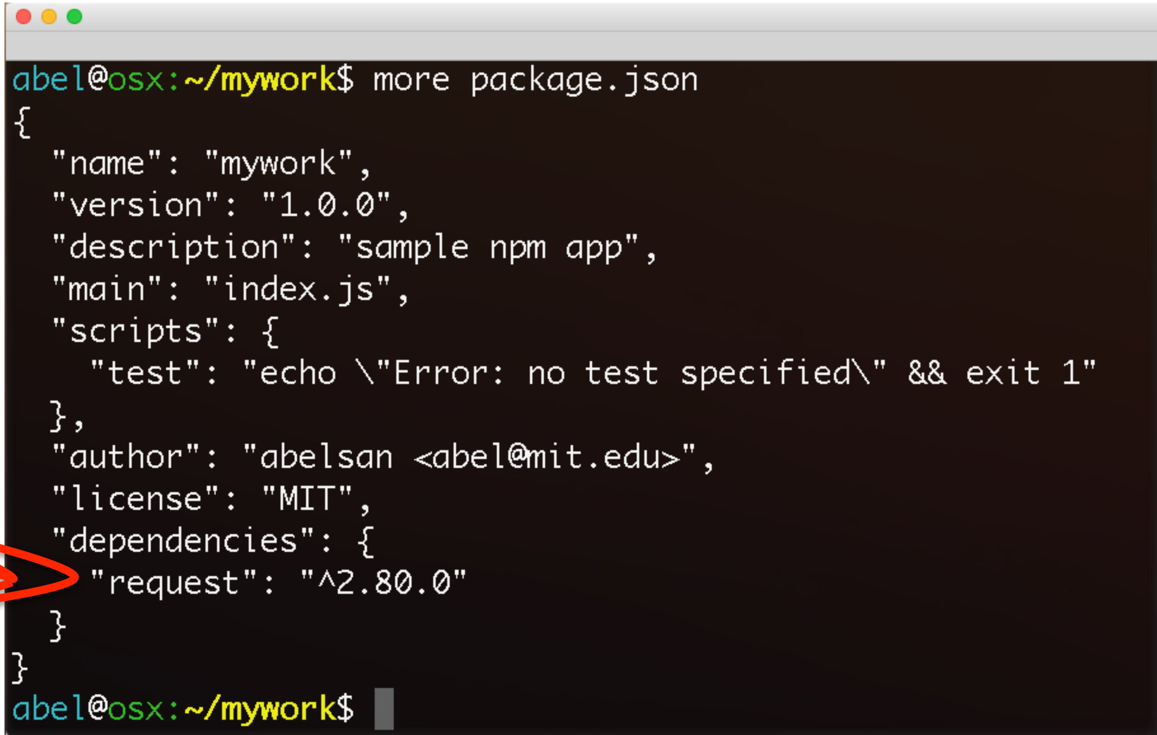
npm

adding npm packages to your application



```
abel@osx:~/mywork$ npm install request --save
```

Local installation of request package. Dependency added to package.json

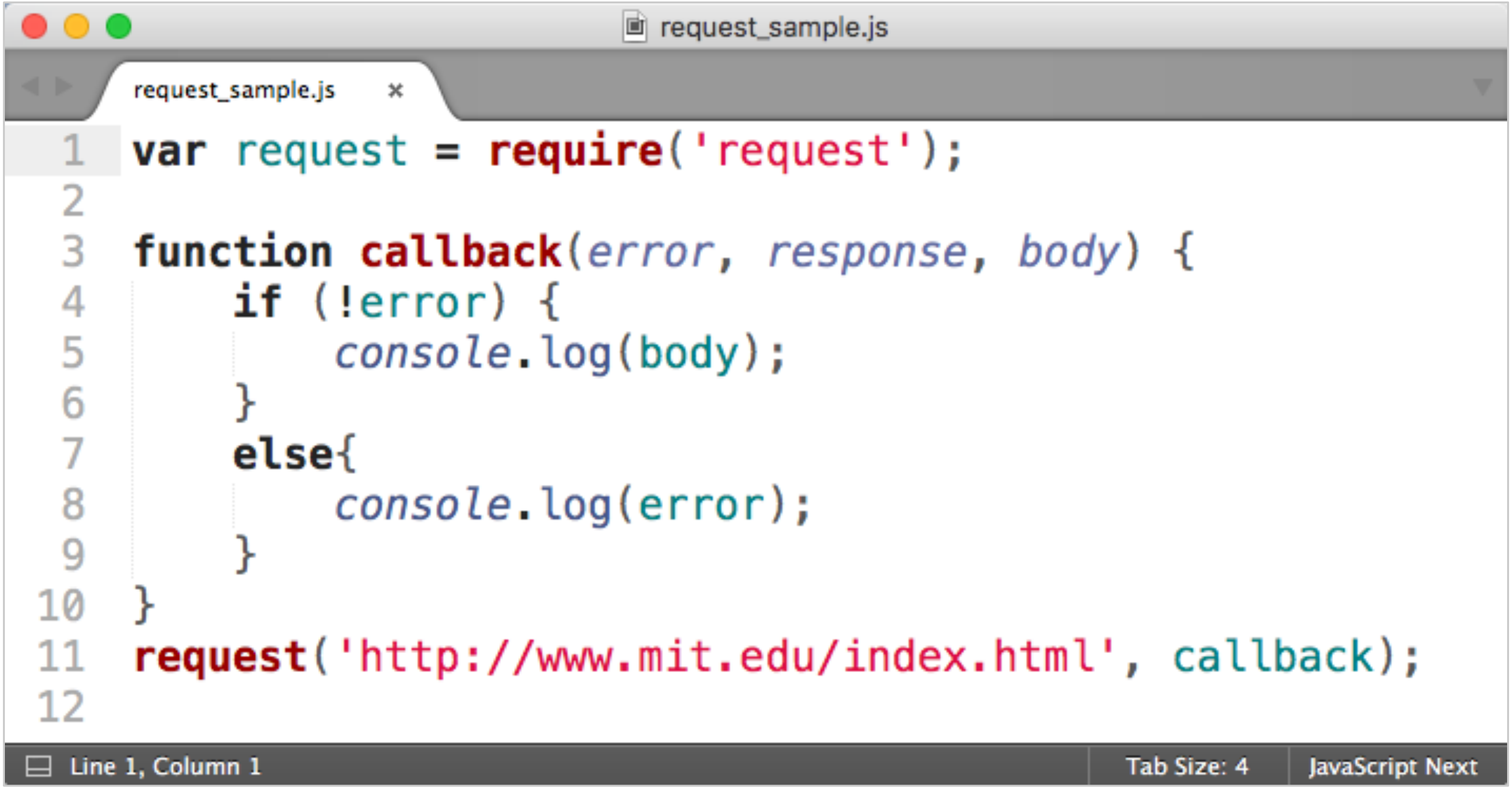


```
abel@osx:~/mywork$ more package.json
{
  "name": "mywork",
  "version": "1.0.0",
  "description": "sample npm app",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "abelsan <abel@mit.edu>",
  "license": "MIT",
  "dependencies": {
    "request": "^2.80.0"
  }
}
abel@osx:~/mywork$
```

Active Learning

- …

Request Basics



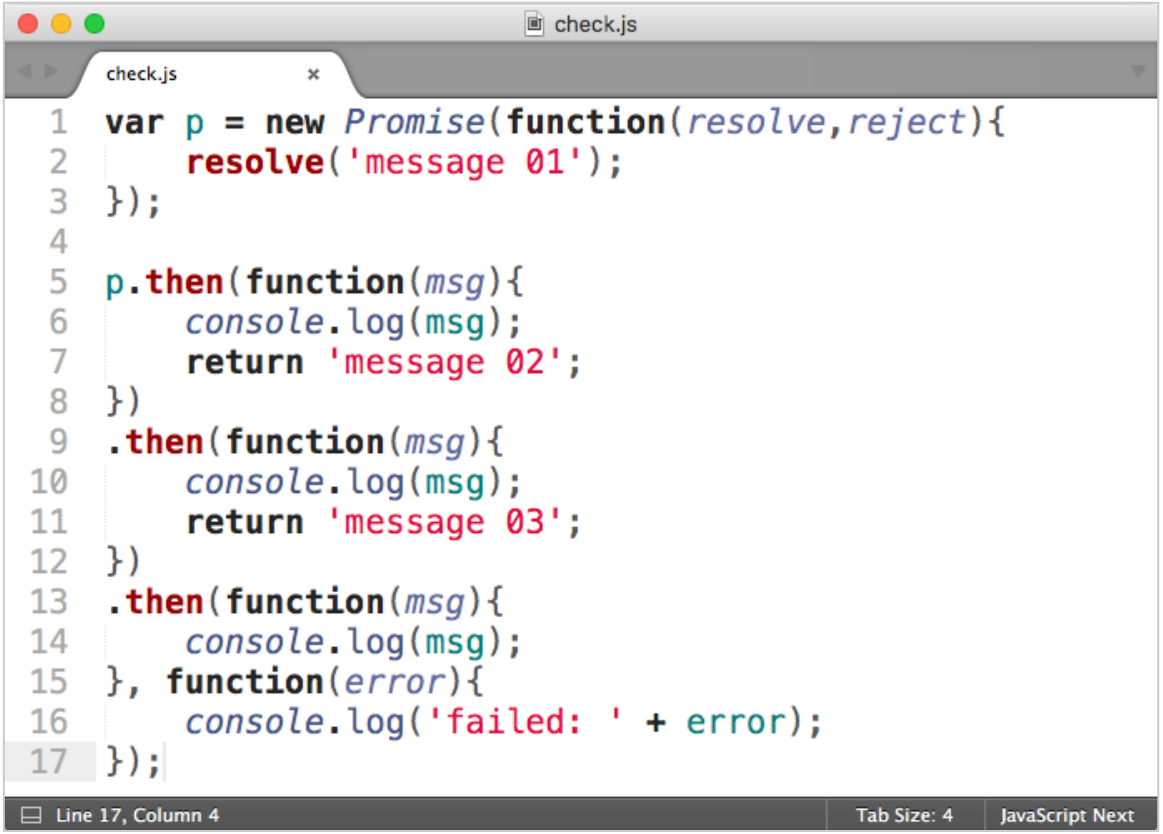
```javascript
var request = require('request');

function callback(error, response, body) {
	if (!error) {
		console.log(body);
	}
	else{
		console.log(error);
	}
}
request('http://www.mit.edu/index.html', callback);

```

Promises Basics

- Basics …


check.js

```
var p = new Promise(function(resolve,reject){
    resolve('message 01');
});

p.then(function(msg){
    console.log(msg);
    return 'message 02';
})
.then(function(msg){
    console.log(msg);
    return 'message 03';
})
.then(function(msg){
    console.log(msg);
}, function(error){
    console.log('failed: ' + error);
});
```

Line 17, Column 4   Tab Size: 4   JavaScript Next