

# NODEJS SUMMARY

---

# NODE CONTENT

- ▶ What is Node?
- ▶ Your first Node Program
- ▶ Node Module System
- ▶ Path Module
- ▶ File System Module
- ▶ Events Module
- ▶ Http Modules
- ▶ **Node Package Manager**

# WHAT IS NODE

- ▶ A **Runtime environment** for executing JavaScript code
- ▶ Often use to build back-end services
- ▶ **Application Programming Interface**



# WHY USING NODEJS

Great for prototyping and agile development

Superfast and highly scalable

JavaScript everywhere

Cleaner and more consistent codebase

Large ecosystem of open-source libs

## NODE APP

**Built twice as fast** with fewer people

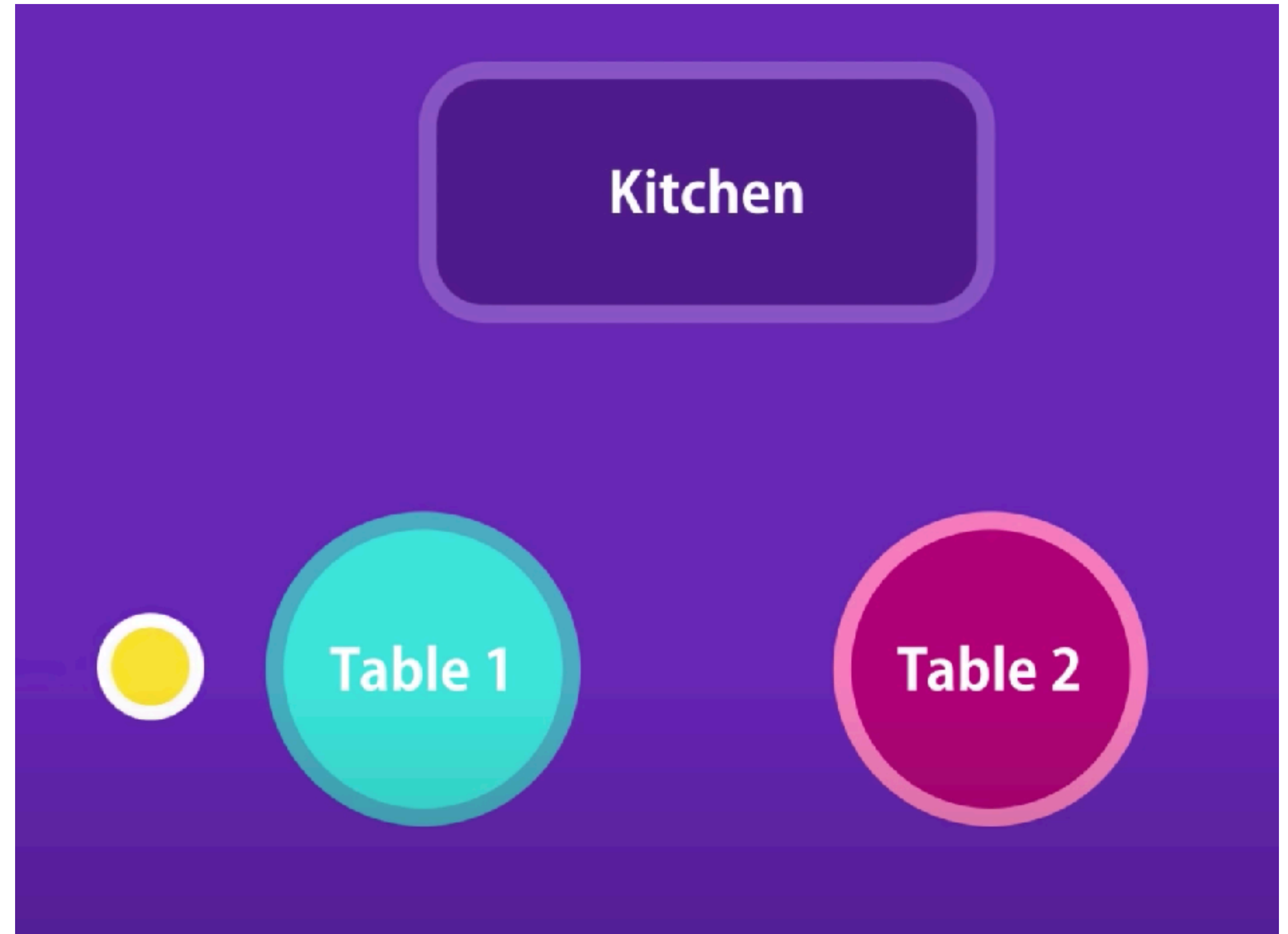
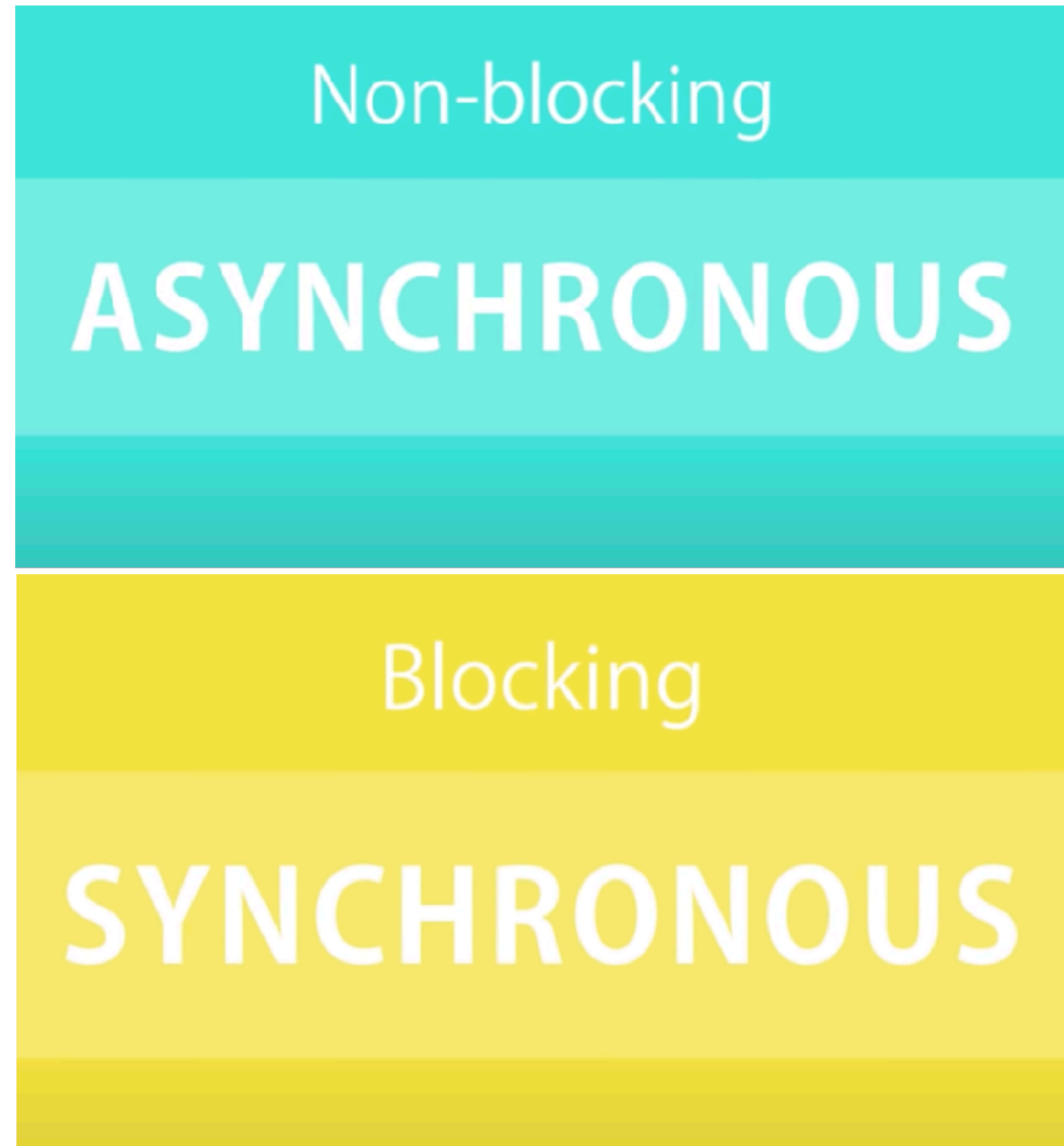
**33%** fewer lines of code

**40%** fewer files

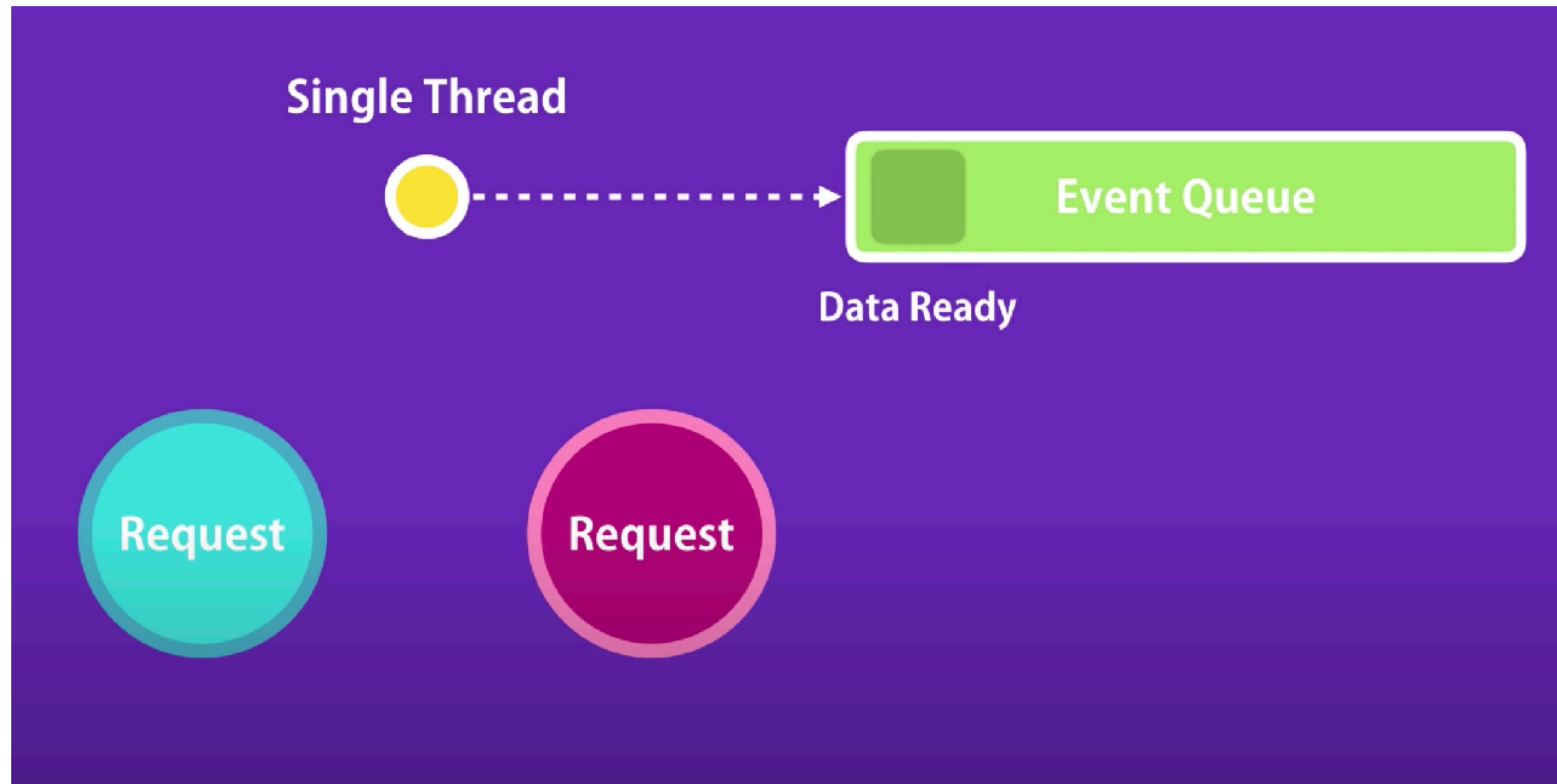
**2x** request/sec

**35%** faster response time

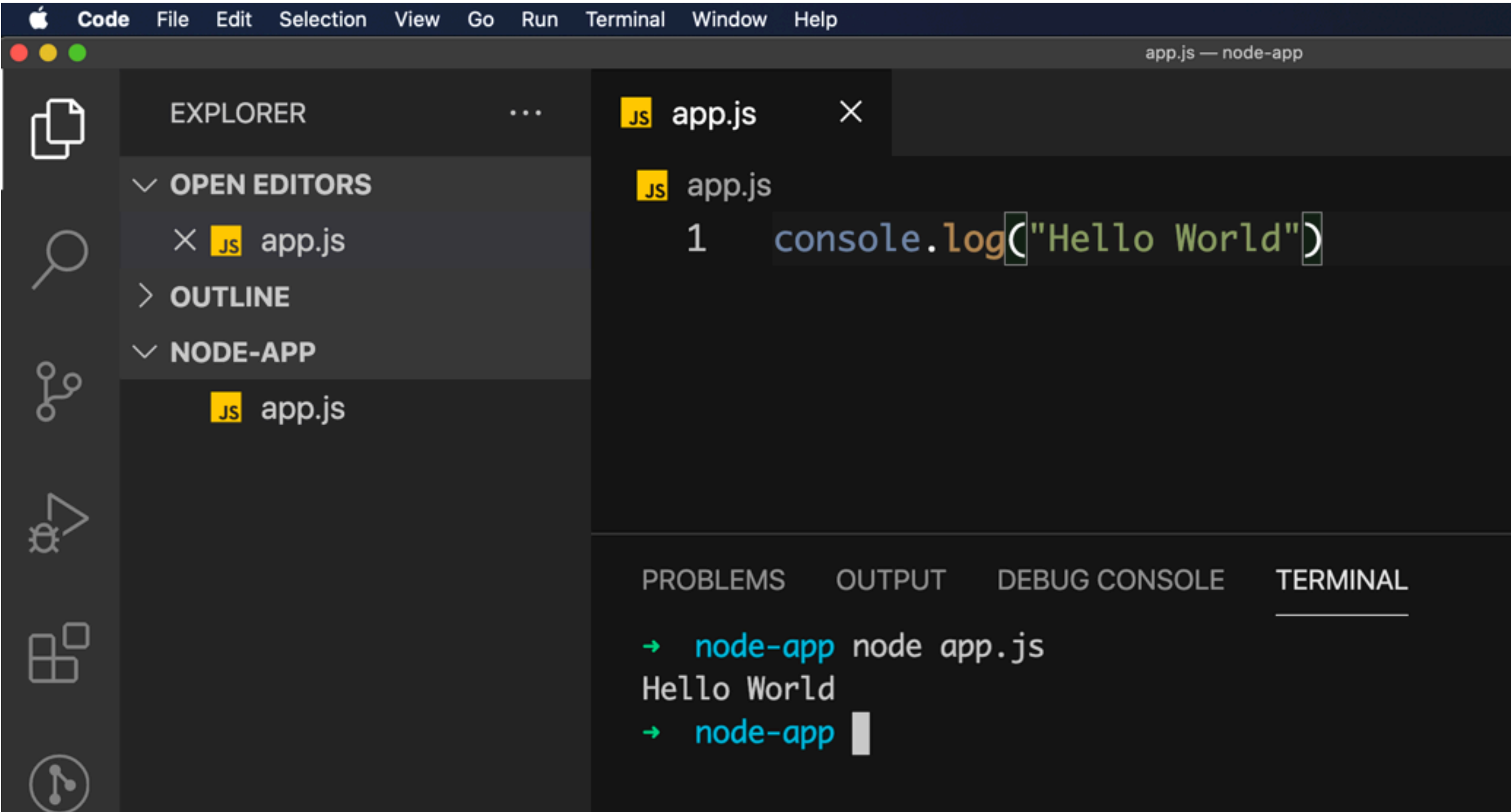
## HOW NODE WORKS?



# HOW NODE WORKS?



# HELLO WORLD



# GLOBAL OBJECT

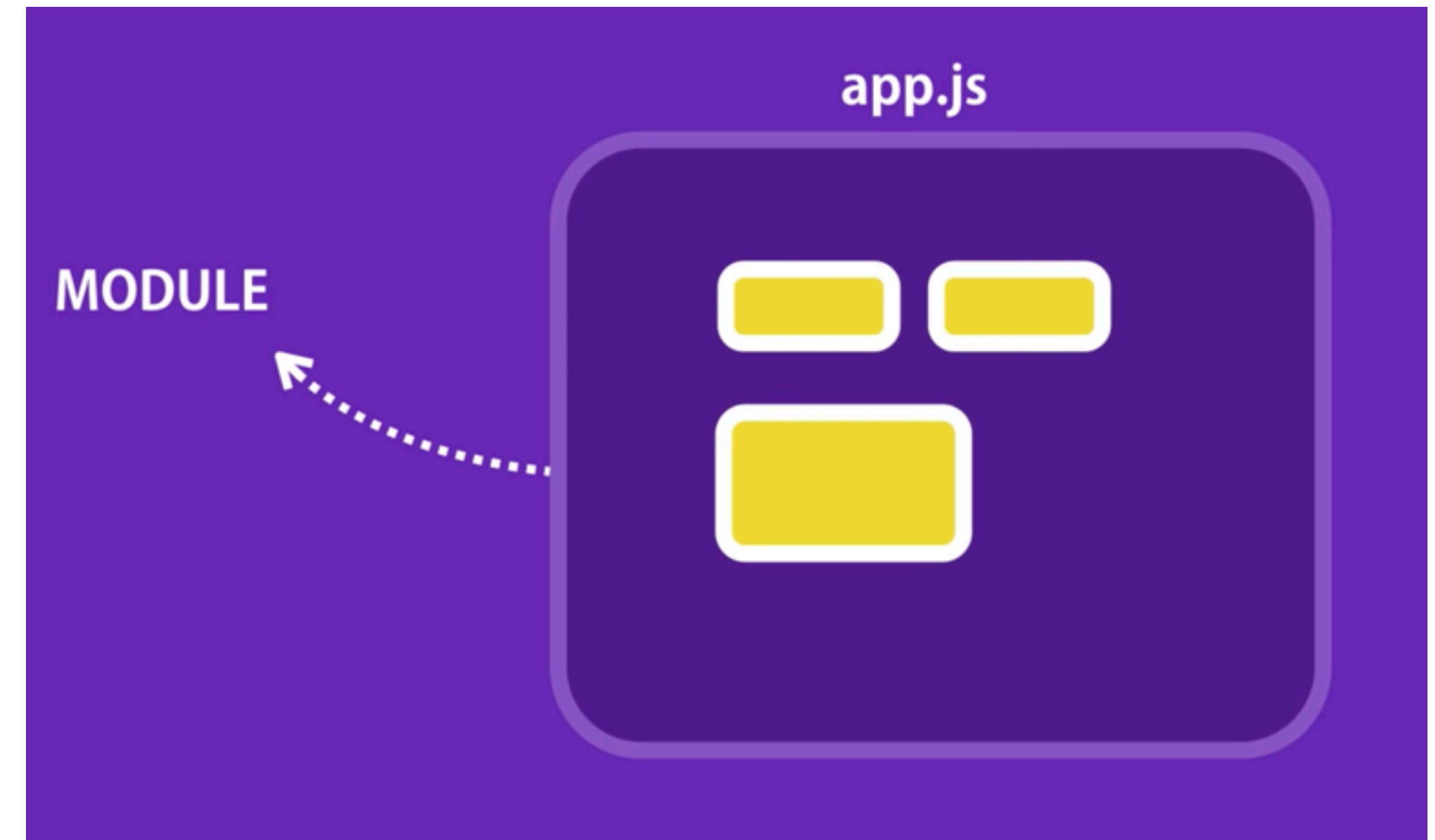
```
Js app.js ×  
  
Js app.js  
1 console.log("Hello World");  
2 console.log(window); // ReferenceError  
3 // window is not defined  
4  
5 setTimeout();  
6 clearTimeout();  
7 setInterval();  
8 clearInterval();  
9 console.log(global);
```

```
Js app.js ×  
  
Js app.js > ...  
1 var globalMessage = "Hello";  
2 console.log(global.globalMessage);  
  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
  
→ node-app node app.js  
undefined  
→ node-app
```



# MODULES

- ▶ Every file in Node Project are a module. ( mean all is private )
- ▶ If you want to use the variable, function, class in a module, you need to make it public.



# MODULES

- ▶ Every file in node project have an object named **module**
- ▶ **Module** is not global object

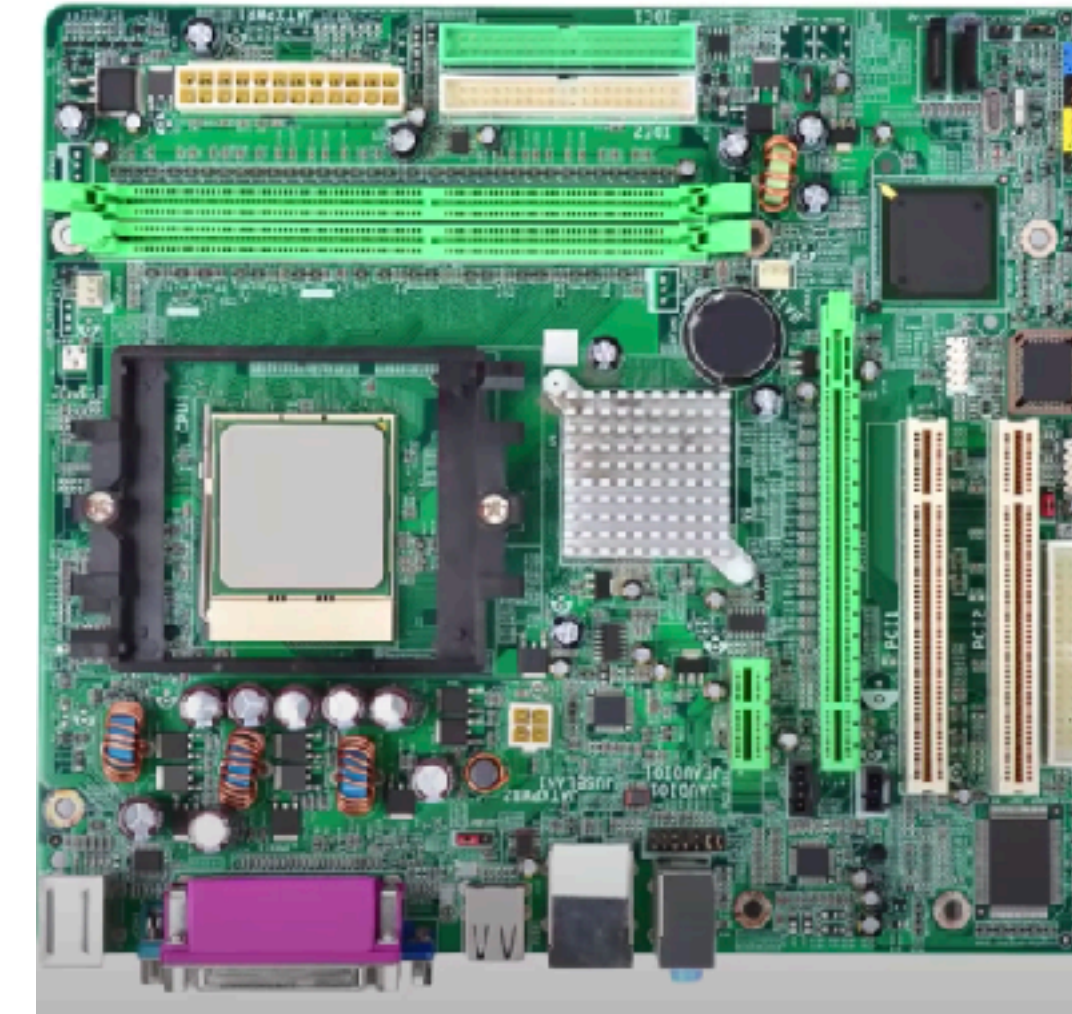
```
js app.js
1 console.log(module);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: zsh

```
→ node-app node app.js
Module {
  id: '.',
  path: '/Users/Kame/Desktop/jsbasic/node-app',
  exports: {},
  parent: null,
  filename: '/Users/Kame/Desktop/jsbasic/node-app/app.js',
  loaded: false,
  children: [],
  paths: [
    '/Users/Kame/Desktop/jsbasic/node-app/node_modules',
    '/Users/Kame/Desktop/jsbasic/node_modules',
    '/Users/Kame/Desktop/node_modules',
    '/Users/Kame/node_modules',
    '/Users/node_modules',
    '/node_modules'
  ]
}
```

# CREATING A MODULE

```
1 function log(message) {  
2     console.log(message);  
3 }  
4 function log(message) {  
5     console.warn(message);  
6 }  
7  
8 module.exports.log = log;  
9 module.exports.warn = warn;
```



Implementation Detail



Public Interface



# LOADING A MODULE

```
JS app.js × JS logger.js

JS app.js > ...
1  const logger = require("./logger");
2  console.log(logger);
3  logger.log("Hell noo. Why you call me");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: zsh

```
→ node-app node app.js
{ log: [Function: log] }
Hell noo. Why you call me
→ node-app
```

```
JS app.js × JS logger.js

JS app.js > ...
1  const {log, warn} = require("./logger");
2  log("Hell noo. Why you call me");
3  warn("And me, also?")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: zsh

```
→ node-app node app.js
Hell noo. Why you call me
And me, also?
→ node-app
```

# PATH MODULE

- ▶ `path.extname(path)`
- ▶ `path.dirname(path)`
- ▶ `path.format(pathObject)`
- ▶ `path.parse(path)`
- ▶ `path.resolve([...paths])`

# FILE SYSTEM MODULE

```
JS app.js ×
JS app.js > ...
1  const fs = require('fs');
2  const files = fs.readdirSync("./");
3  console.log(files);
4  fs.readdir("./", function (err, files) {
5      if (err) console.error("error", err);
6      else console.log("files", files);
7  })

const searchImages = function (path) {
  readdirSync(path).forEach(function (item) {
    let stat = statSync('' + path + '/' + item);
    if (stat.isDirectory()) {
      return searchImages('' + path + '/' + item);
    }
    else if (['jpg', 'png', 'jpeg'].indexOf(item.split('.').pop()) > -1) {
      return imageUrl.push({ path: '' + path + '/' + item, status: 'raw' });
    }
  });
};

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: zsh
→ node-app node app.js
[ 'app.js', 'assets', 'logger.js' ]
files [ 'app.js', 'assets', 'logger.js' ]
```

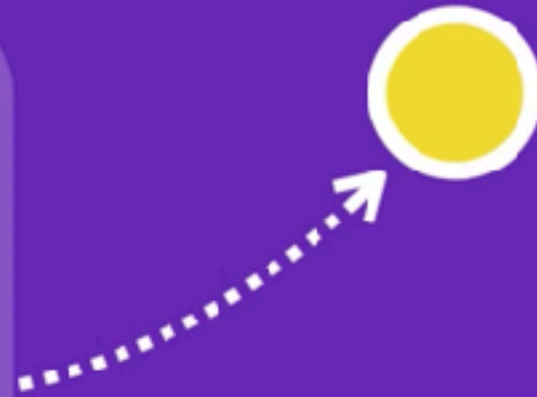
# EVENTS MODULE

## EVENT

A signal that something has happened

HTTP

Event: New Request



## Node.js v16.0.0 documentation

[Index](#) | [View on single page](#) | [View as JSON](#) | [View another version ▼](#) | [Edit on GitHub](#)

### ▼ Table of contents

#### ▪ Events

- Passing arguments and `this` to listeners
- Asynchronous vs. synchronous
- Handling events only once
- Error events
- Capture rejections of promises

#### ▪ Class: `EventEmitter`

- Event: `'newListener'`
- Event: `'removeListener'`
- `emitter.addListener(eventName, listener)`
- `emitter.emit(eventName[, ...args])`

# EVENTS MODULE

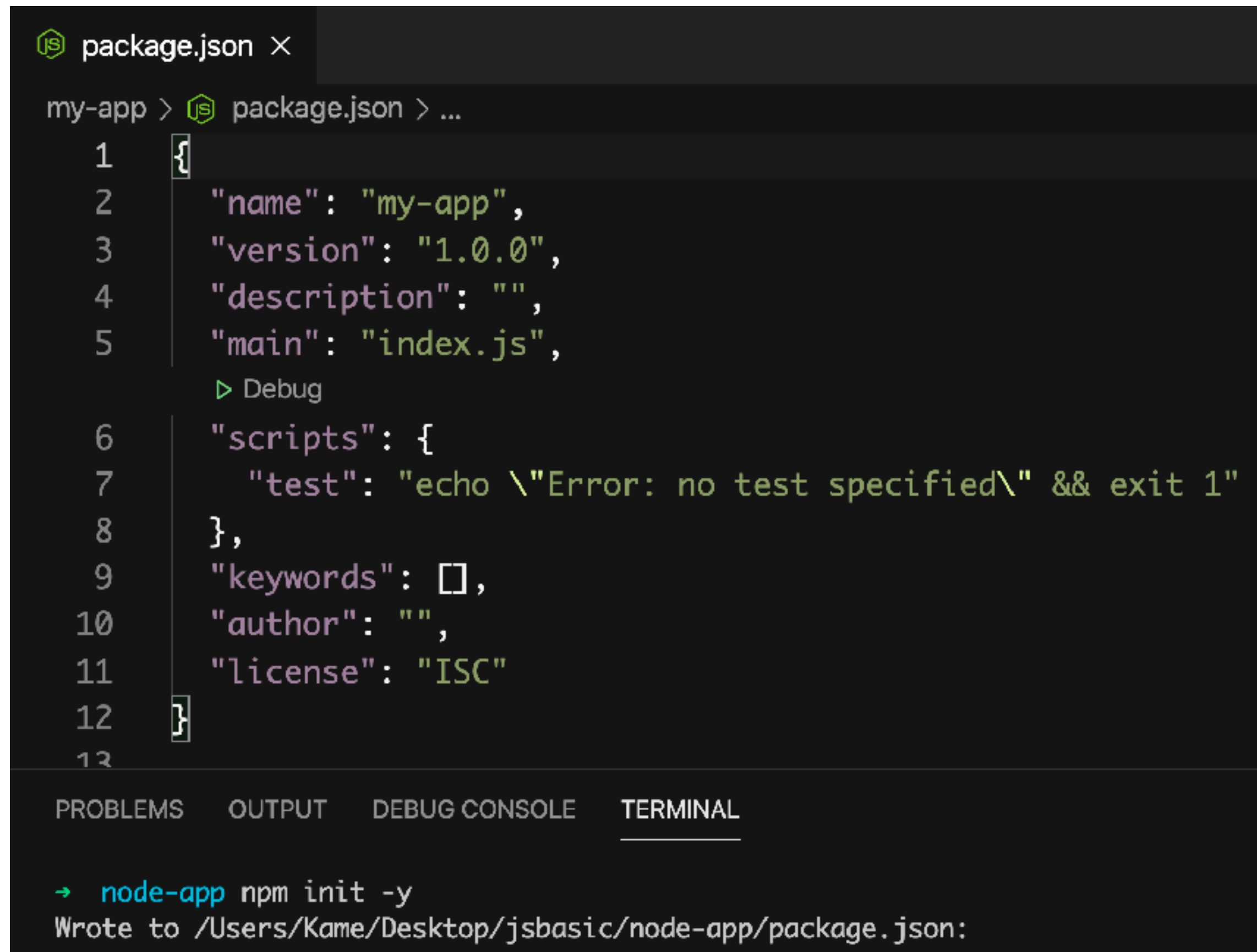
- ▶ `emitter.on(event, listener)`
- ▶ `emitter.emit(event, parameters)`
- ▶ `emitter.off(event, listener)`



# HTTP MODULE

```
1 const http = require("http");
2 const server = http.createServer();
3 server.on('connection', (socket) => {
4     console.log("new connection");
5 });
6 server.listen(3000);
7 console.log("listenning on port 3000");
8
```

# NODE PACKAGE MANAGER



The screenshot shows a VS Code editor with a file named `package.json` open. The file content is as follows:

```
1 {  
2   "name": "my-app",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"  
8   },  
9   "keywords": [],  
10  "author": "",  
11  "license": "ISC"  
12 }  
13
```

Below the editor, the **TERMINAL** tab is active, showing the command `node-app npm init -y` and its output: `Wrote to /Users/Kame/Desktop/jsbasic/node-app/package.json:`

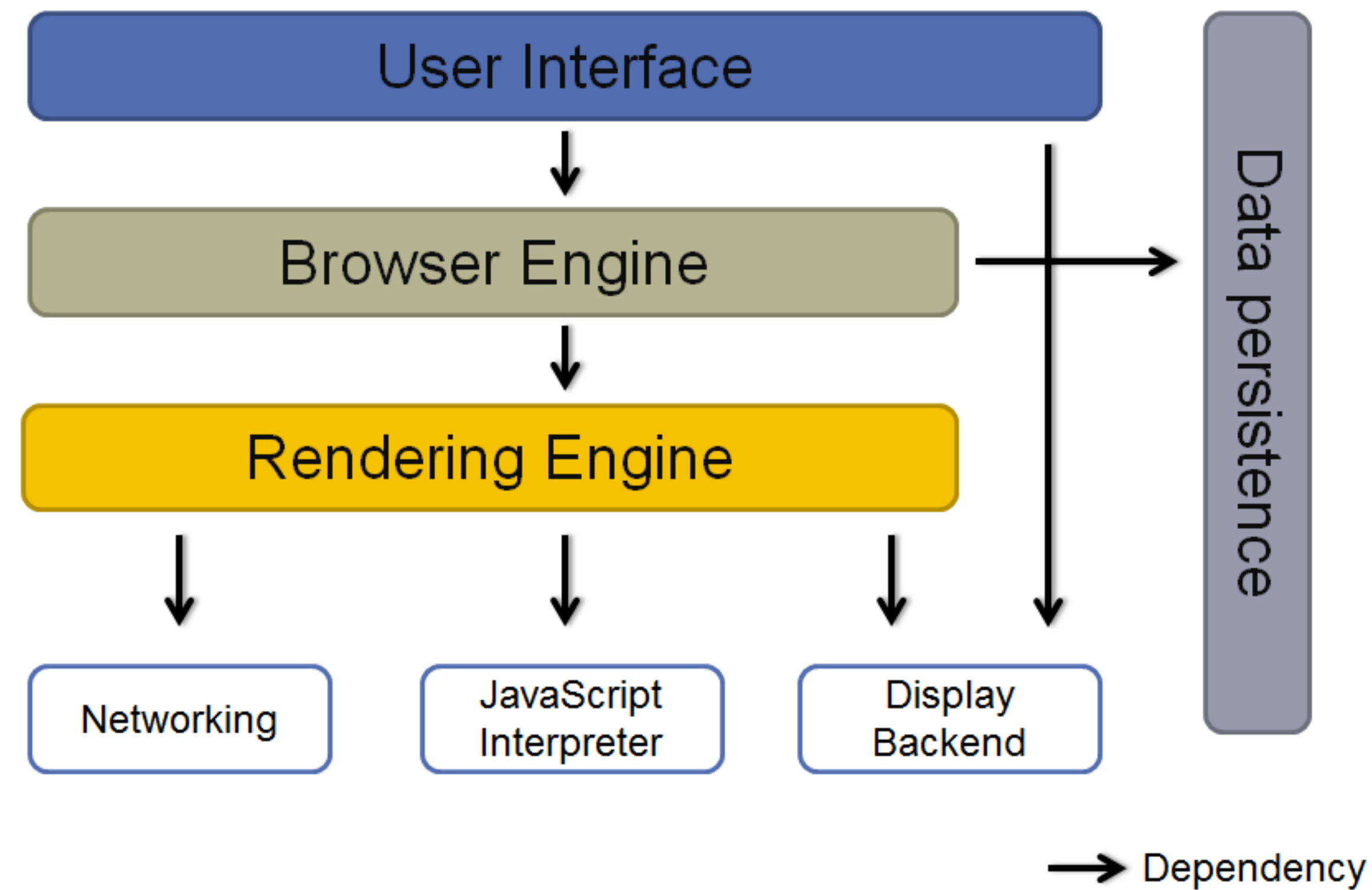
```
  "author": "Kame",  
  "dependencies": {  
    "hashmap": "^2.4.0",  
    "lodash": "^4.17.15",  
    "queue-fifo": "^0.2.6",  
    "socket.io-client": "^2.3.0",  
    "uuid": "^3.3.3"  
  },  
  "devDependencies": {  
    "@babel/core": "^7.8.3",  
    "@babel/preset-env": "^7.9.5",  
    "@babel/preset-typescript": "^7.8.3",  
    "@types/howler": "^2.1.2",  
    "@types/lodash": "^4.14.150",  
    "@types/node": "^13.13.0",
```

# ADDITIONAL

- ▶ Browser
- ▶ Develop Tools
- ▶ Command Line
- ▶ Rules
- ▶ Git

# BROWSER

## BROWSER ARCHITECTURE



# CHROME DEV TOOLS

- ▶ Inspector
- ▶ Console
- ▶ Debug
- ▶ Perfomance
- ▶ NetWork
- ▶ Memory
- ▶ SessionStorage, LocalStorage, Cookie

## SOME RULE IN DEVELOPMENT

- ▶ DRY - Don't repeat your self
- ▶ KISS - Keep it simple stupid
- ▶ Single Responsibility Principle