

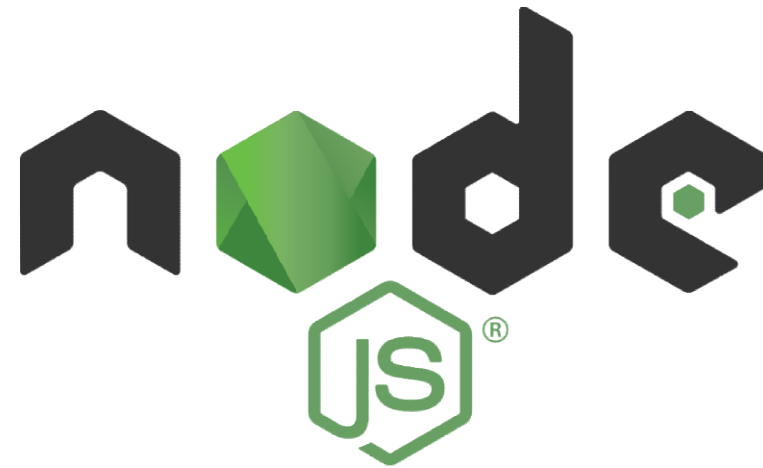
พัฒนา Modern Web Application ด้วย

Node.js

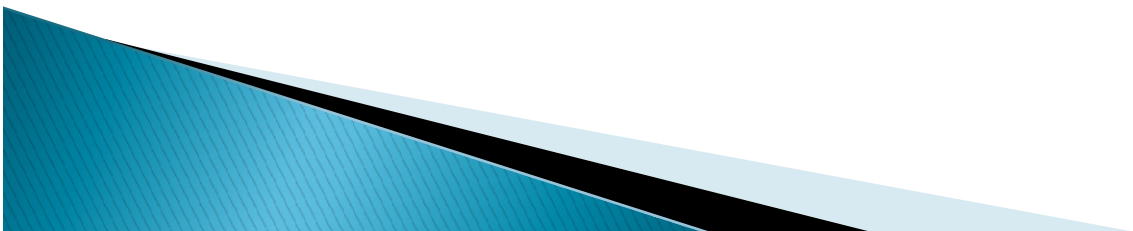


โค้ชเอก

[Codingthailand.com](http://Codingthailand.com)

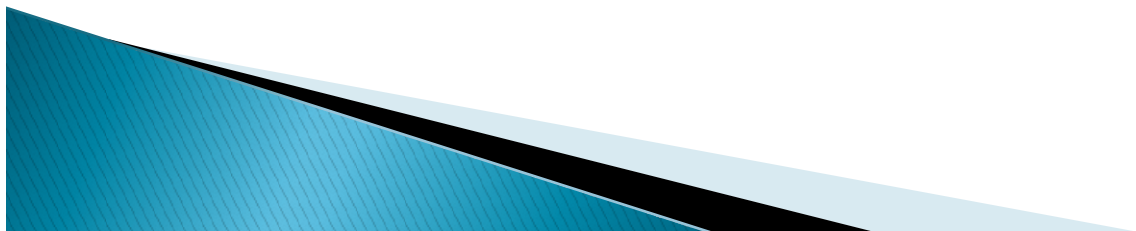


Node.js

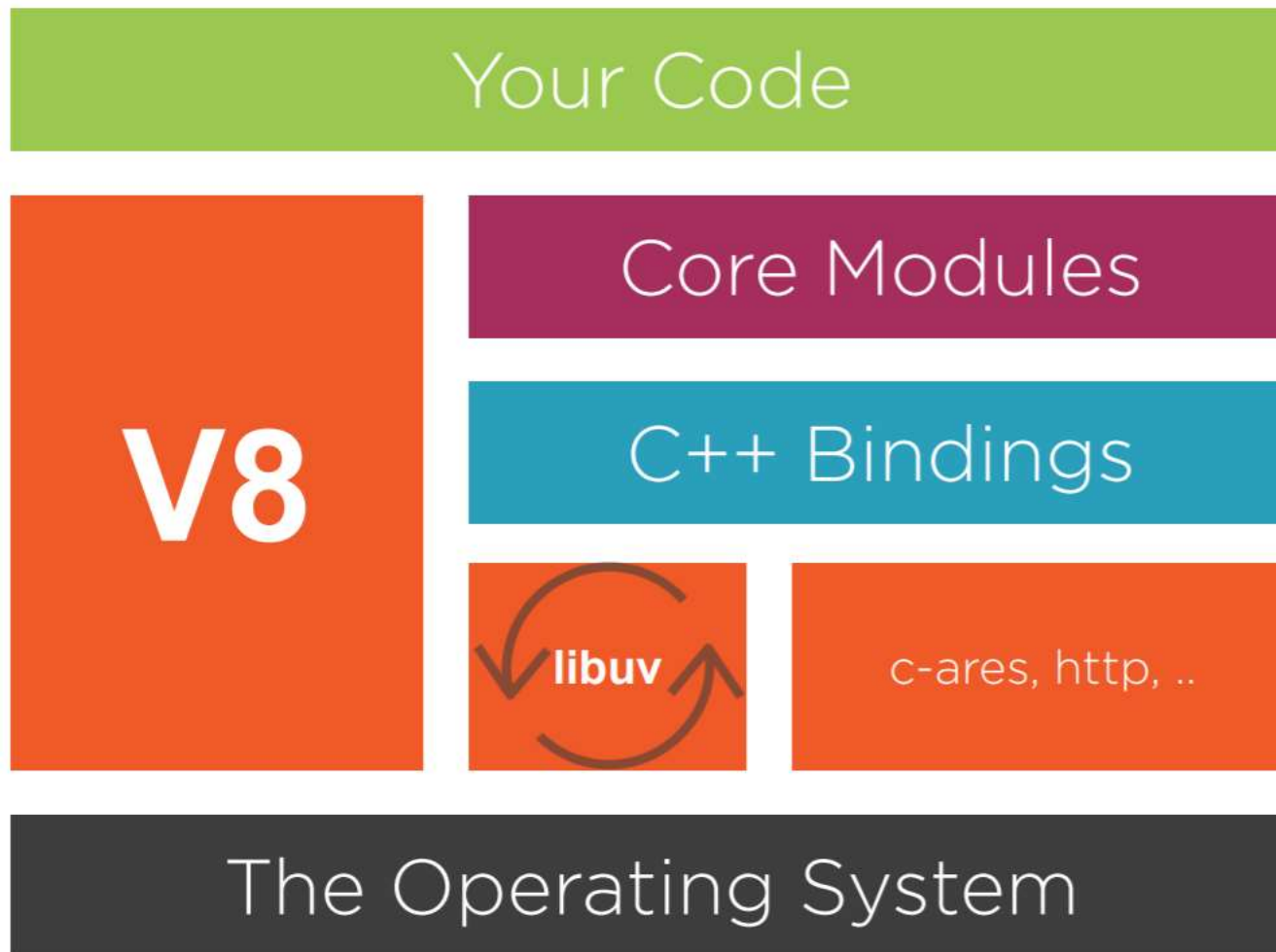


# What is Node?

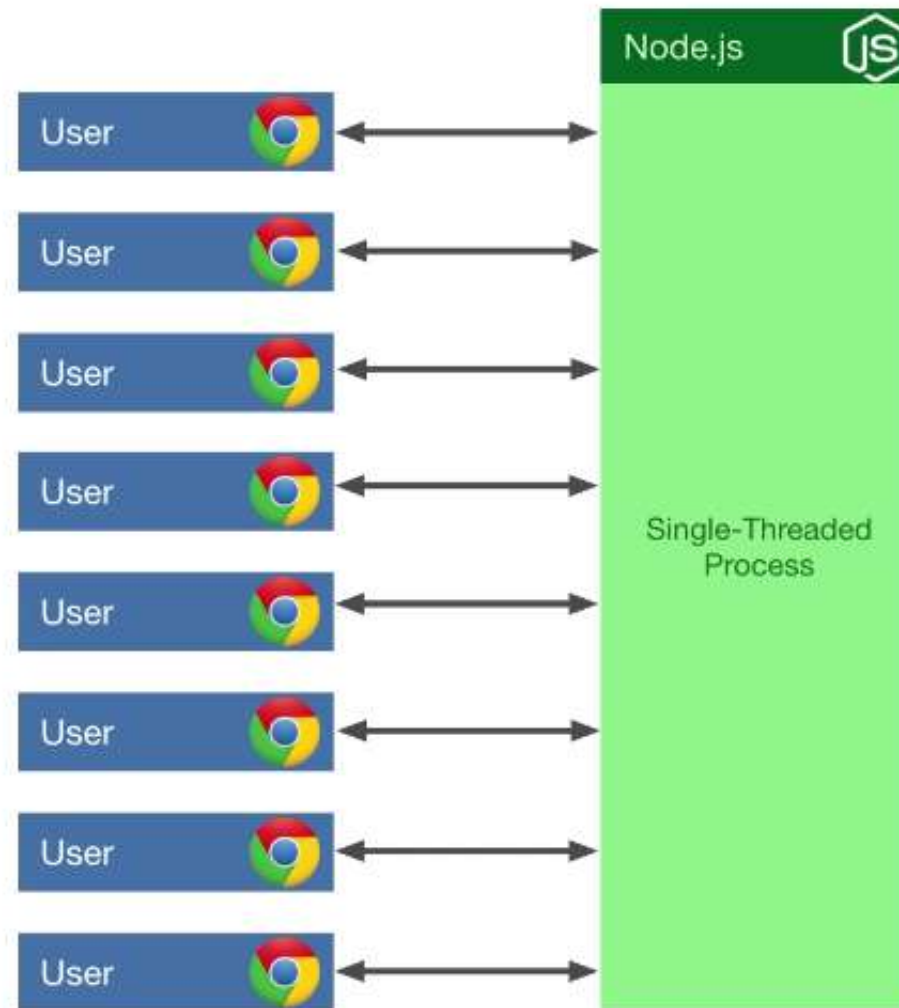
- ▶ Having been first published by Ryan Dahl in 2009
- ▶ The open source software that allows the common web language of JavaScript to run outside the browser.
- ▶ Node.js is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#)
- ▶ Node.js uses an event-driven, non-blocking I/O model
- ▶ <https://nodejs.org/en/>



# สถาปัตยกรรมของ Node.js

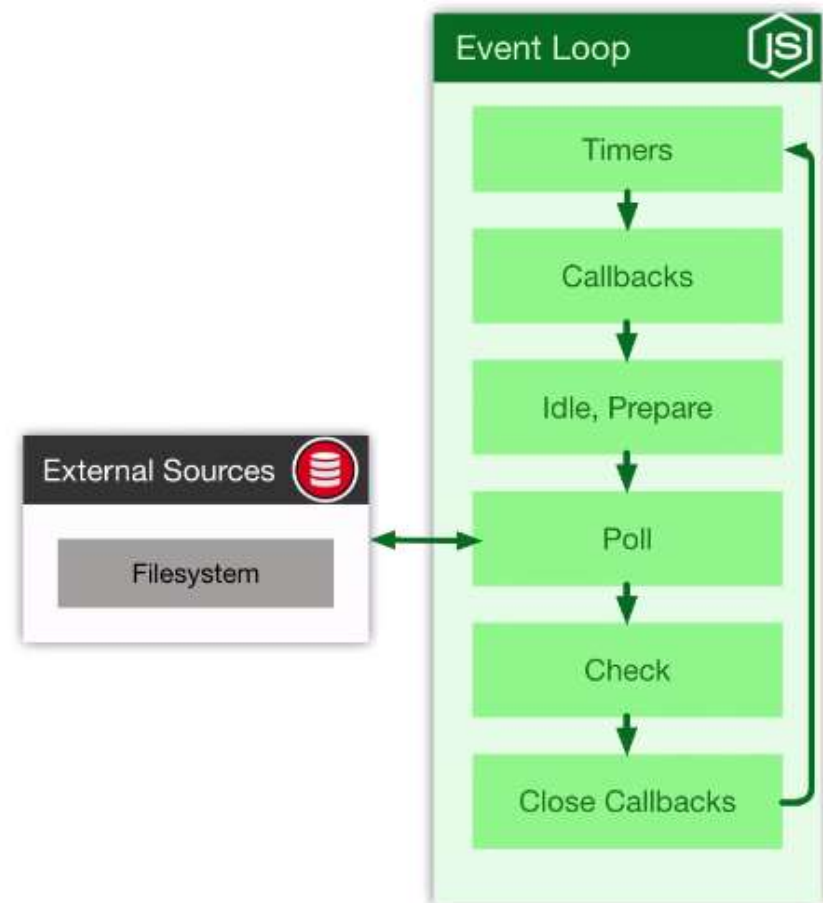
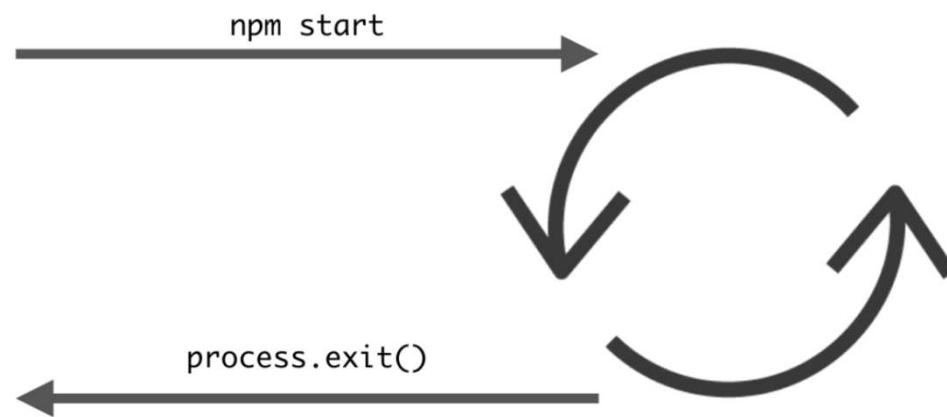


# สถาปัตยกรรมของ Node.js

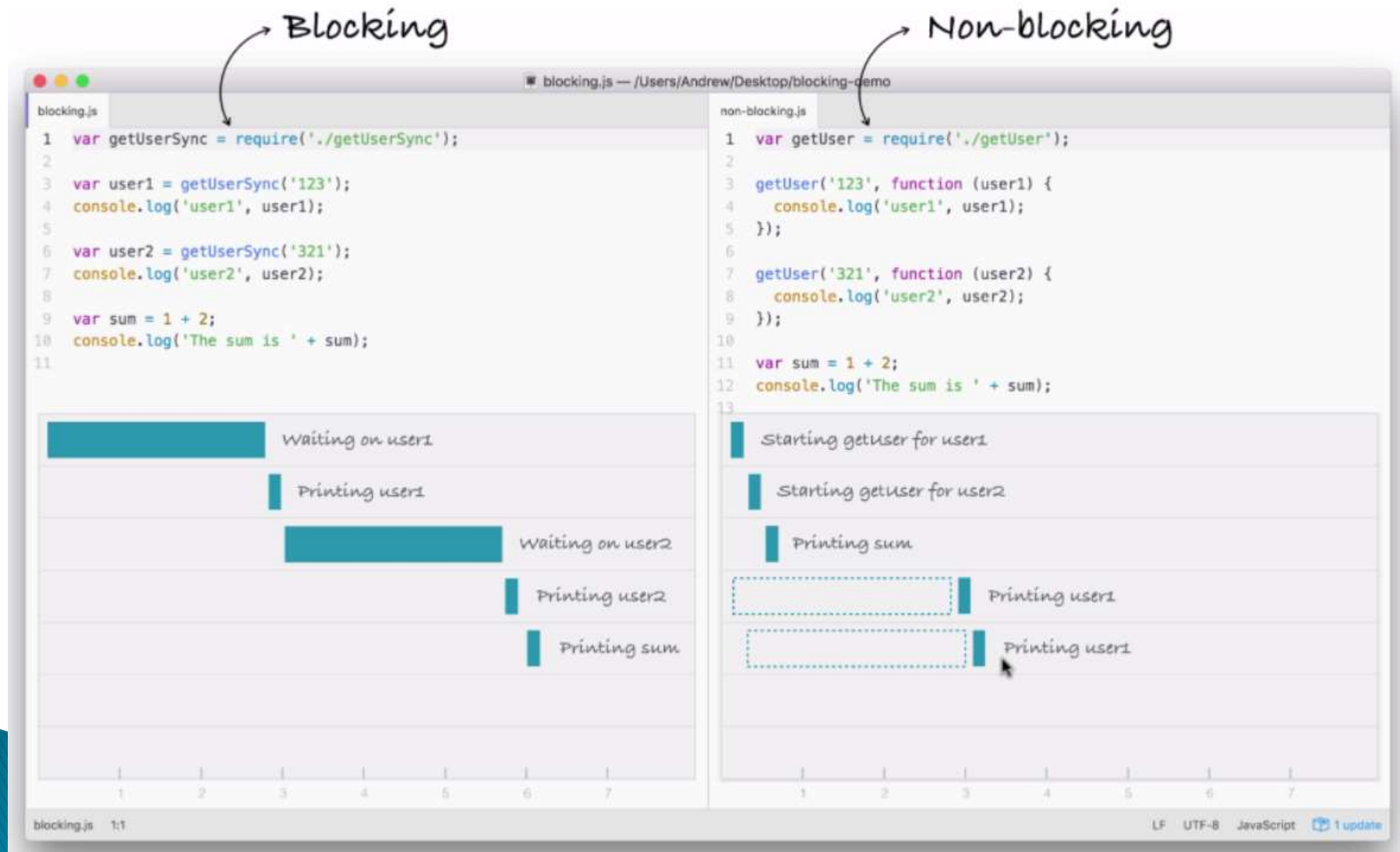


# The Event Loop

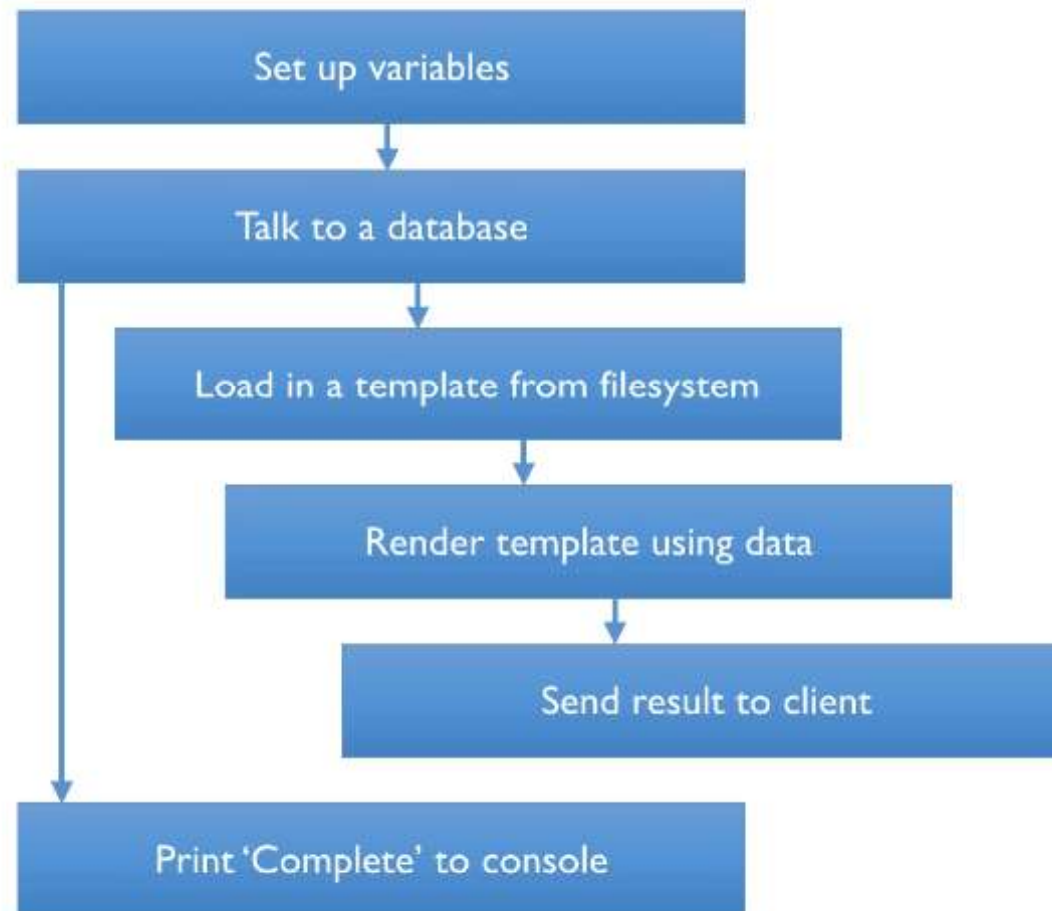
- ▶ A loop that picks events from the event queue and push their callbacks to the call stack



# Blocking vs. Non-blocking



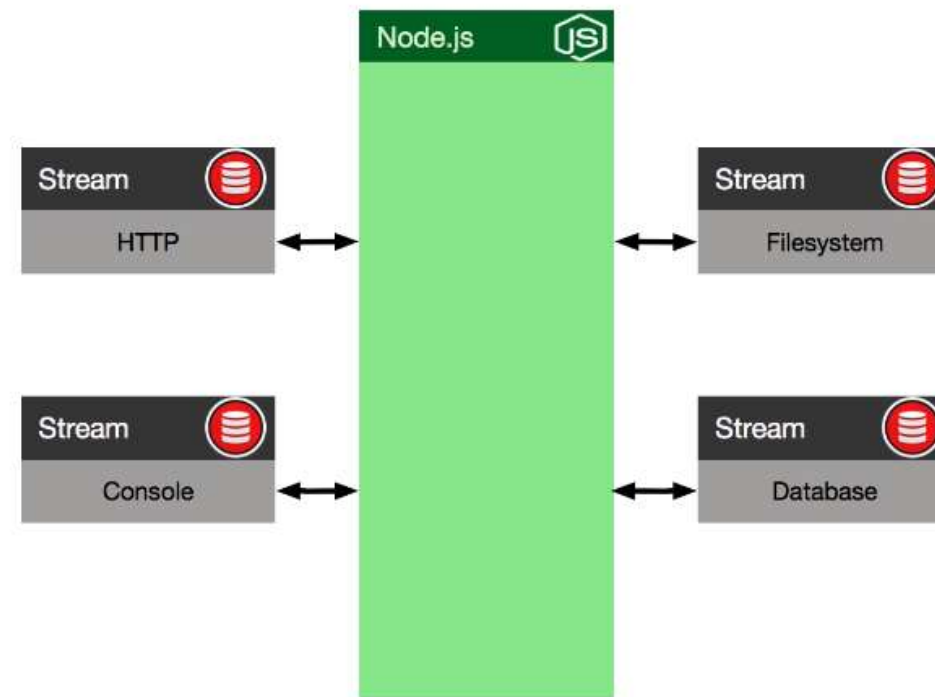
# Non-blocking





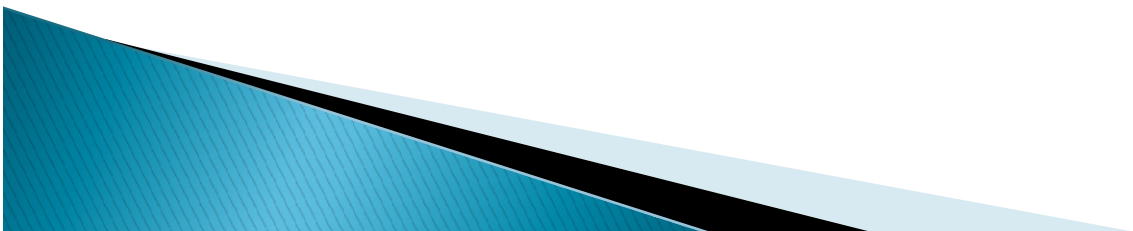
# Node CLI and Global Object

- ▶ Global Object
- ▶ ลองพิมพ์ global. (กด tab 2 ครั้ง)
- ▶ Process
- ▶ OS
- ▶ Path
- ▶ Http / Https
- ▶ Crypto
- ▶ File System
- ▶ Url



# Filesystem Access

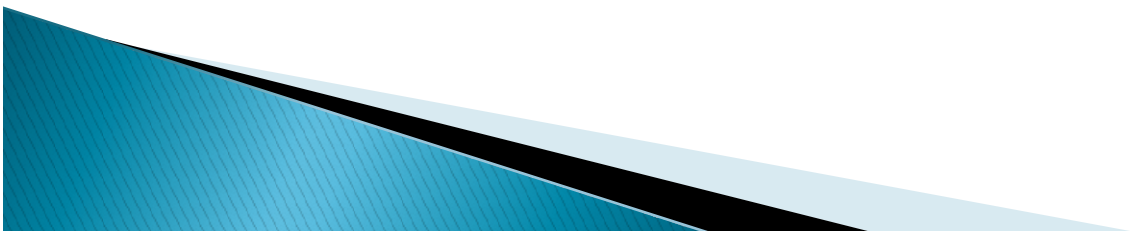
```
fs.writeFileSync(path.join(__dirname, 'user.json'), JSON.stringify(student) , (err, data) => {  
  if (err) {  
    console.error(err);  
    return;  
  }  
});  
  
fs.readFile(path.join(__dirname, 'user.json'), (err, data) => {  
  if (err) {  
    console.log(err);  
    return;  
  }  
  const users = JSON.parse(data);  
  for (const user of users) {  
    console.log(user.id);  
    console.log(user.name);  
  }  
});
```



# Operating System (OS)

```
const os = require('os');

console.log("Hostname: " + os.hostname());           // prometheus
console.log("OS type: " + os.type());                 // Linux
console.log("OS platform: " + os.platform());         // linux
console.log("OS release: " + os.release());           // 3.13.0-52-generic
console.log("OS uptime: " +
    (os.uptime()/60/60/24).toFixed(1) + " days");    // 80.3 days
console.log("CPU architecture: " + os.arch());        // x64
console.log("Number of CPUs: " + os.cpus().length);   // 1
console.log("Total memory: " +
    (os.totalmem()/1e6).toFixed(1) + " MB");          // 1042.3 MB
console.log("Free memory: " +
    (os.freemem()/1e6).toFixed(1) + " MB");           // 195.8 MB
```



# Web Servers

```
const http = require('http');

const server = http.createServer(function(req, res) {
  console.log(`${req.method} ${req.url}`);
  res.end('Hello world!');
});

const port = 8080;
server.listen(port, function() {
  // you can pass a callback to listen that lets you know
  // the server has started
  console.log(`server started on port ${port}`);
});
```



# รูปแบบการเรียกใช้ Core Modules, File Modules, and npm Modules

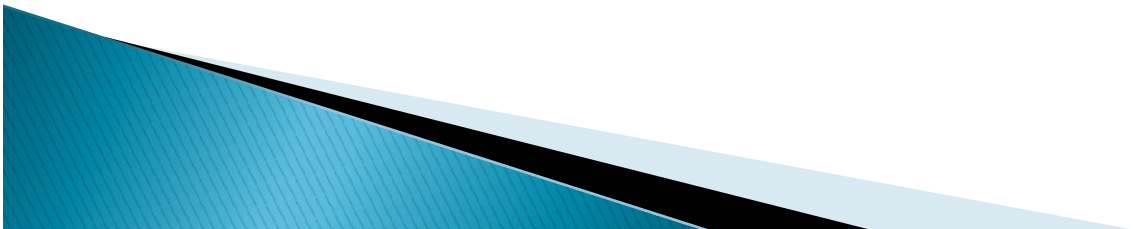
Type	String passed to require	Examples
<b>Core</b>	Doesn't start with /, ./, or ../	<code>require('fs')</code> <code>require('os')</code> <code>require('http')</code> <code>require('child_process')</code>
<b>File</b>	Starts with /, ./, or ../	<code>require('./debug.js')</code> <code>require('/full/path/to/module.js')</code> <code>require('../a.js')</code> <code>require('../../a.js')</code>
<b>npm</b>	Not a core module and doesn't start with /, ./, or ../	<code>require('debug')</code> <code>require('express')</code> <code>require('chalk')</code> <code>require('koa')</code> <code>require('q')</code>



## ตัวอย่างการสร้างไฟล์ Modules ของเราเอง

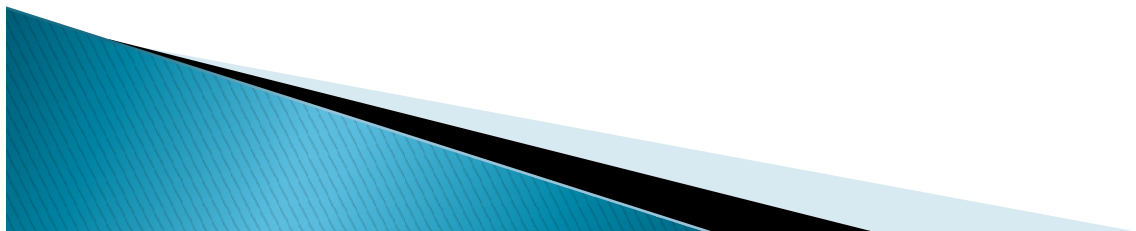
```
function calculate(a, x, n) {  
    if(x === 1) return a*n;  
    return a*(1 - Math.pow(x, n))/(1 - x);  
}  
  
module.exports = calculate;
```

```
const amanda_calculate = require('./amanda.js');  
const tyler_calculate = require('./tyler.js');  
  
console.log(amanda_calculate(1, 2, 5)); // Logs 31  
console.log(tyler_calculate(2)); // Logs 33.510321638291124
```



# Debugging Node.js Apps

- ▶ <https://nodejs.org/en/docs/inspector/>
- ▶ `node -inspect-brk index.js`
- ▶ ทดลองเขียนโค้ดเพื่อ debug





The end

