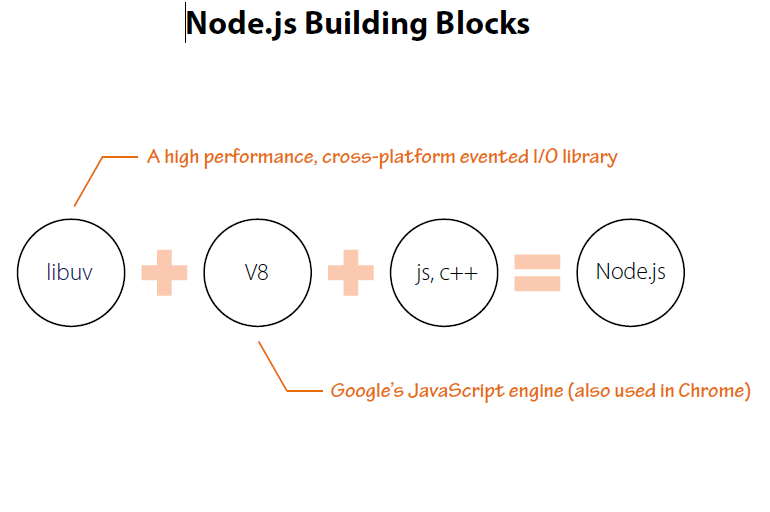
# **Node Tutorial**

**Chapter 1**

* An overview of Node.js
* Building and installing Node.js
* Developing for Node with Cloud9 IDE
* An introduction to Node’s event loop
* Writing code with callbacks

# An overview of Node.js



# Building and installing Node.js

[**http://nodejs.org/download/**](http://nodejs.org/download/)

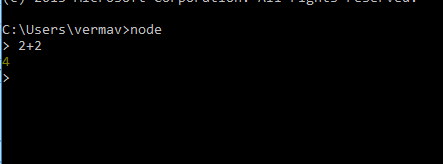
**Installers available for Windows & Mac OS X**

**Binaries available for Windows, Mac, Linux and SunOS** Also available via many Linux package managers

Tesing Node is installed or not

**Node –v or node –version**

User node in terminal



Now node is working fine

You can run a single file with node command

Node filename.js

Run code through nodemon

Install nodemon in your machine (nodemon will watch the files in the directory in which nodemon was started, and if any files change, nodemon will automatically restart your node application.)

**npm install -g nodemon**

**nvm install 0.8.14**

**nvm use 0.6.19**

# Developing for Node with Vscode

### best vscode extensions angualr js

HTML CSS Class Completion

Bracket Pair Colorizer

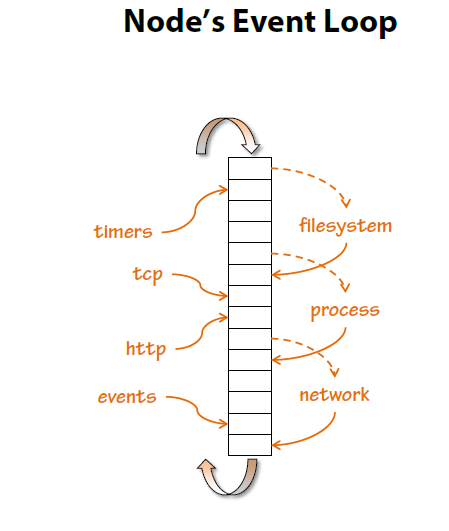
TypeScript

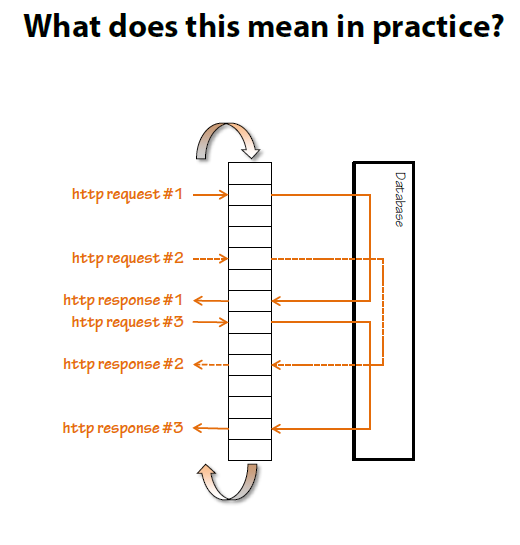
TSLint

TypeScript Hero

Node.js Modules Intellisense

# An introduction to Node’s event loop





# Writing code with callbacks

**A typical approach / blocking code**

var conn = getDbConnection(connectionString);

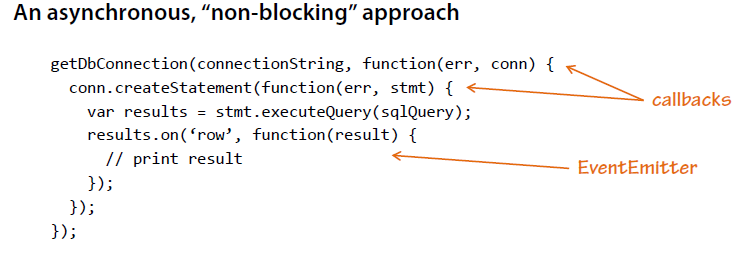
var stmt = conn.createStatement();

var results = stmt.executeQuery(sqlQuery);

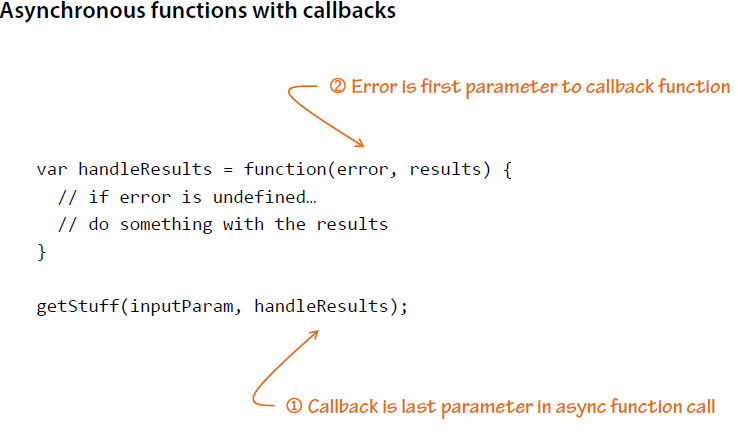
for (var i=0; i<results.length; i++) {

// print results[i];

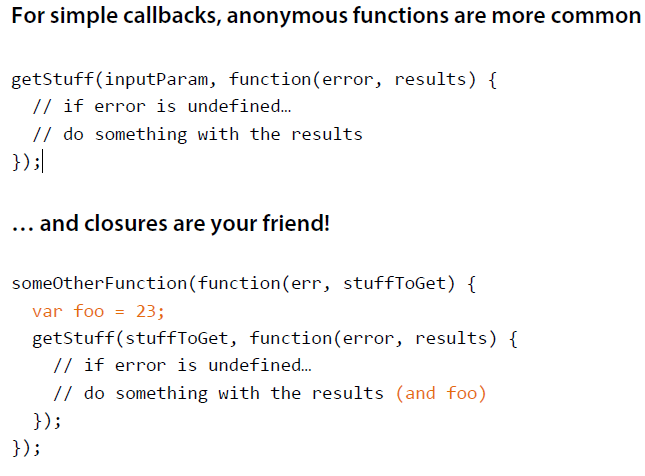
}



**Coding for asynchrony with callbacks**



**Anonymous Functions and Closures**



**“Christmas tree” effect!**

asyncFunction1(inputParam, function(err, results1) {

asyncFunction2(results1, function (err, results2) {

asyncFunction3(results2, function (err, results3) {

asyncFunction4(results3, function (err, results4) {

asyncFunction5(results4, function (err, results5) {

// and so on…

});

});

});

});

});

