Node.js BASICS

LEARN THE BASICS OF NODE

Node.js Basics

Rick Hernandez

This book is for sale at http://leanpub.com/nodejsbasics

This version was published on 2016-03-25

* * * * *

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

* * * * *

Table of Contents

Preface

I want to take this time to say thank you and **congratulate** you for taking the first step to learning how to use Node.js. The wide adoption that Node.js has obtain over the years has been astonishing, it's a great tool for new programmers that want to start using JavaScript outside of the browser. This book is created to help you get started using Node.js in your specific operating system, along with helping you keep Node.js up to date with the latest releases.

Just like most tools this book will show clear data of when to use Node.js and who is using Node.js and how they are using it. The purpose of the book is to get you comfortable enough for you to be able to take this knowledge of Node.js and start learning to program with JavaScript or to dig deeper into more in depth parts of Node.js.

One reason or another you want to learn Node.js and this is the book that will welcome you into the world of Node.js and JavaScript.

About the Author

Rick H

Rick Hernandez works as a software developer with small to mid-size companies. He has been responsible for evaluating business requests to determine feasibility, identifying options, and recommend solutions for software development enhancements. He has assisted with interpreting customer requirements into conceptual design specifications, and has developed interfaces/prototypes and maintained software solutions. He also has written over 300+ technical publications published at Code With Intent. Rick has a deep passion and curiosity to help others succeed.

Say Hello

My intention with this book is to get you up and going as quickly as possible with Node.js. If for what ever reason you might have a question, comment or suggestion about the book I would encourage you to take the time and write in to rick@codewithintent.com.

Who Is This Book For?

This book is created for new JavaScript programmers in mind. No programming experience is required, **you do not need to know JavaScript nor any of the web languages such as HTML/CSS**. The purpose of this book is to help you get started using Node.js. While learning to use Node.js we will explore other questions such as:

- Why was Node.js created?
- Why you need to know Node.js
- What companies are using Node.js
- Working with the event loop
- Understanding how npm works
- Versioning in Node.js
- Why are Node.js releases named after elements in the Periodic Table?
- How to install and configure Node.js in Windows, OSX, and Linux
- Running a simple application through the Command Line

Below you will find the most common questions that I get about the book.

- Will you cover how to program with JavaScript?

No, this book is a prerequisite for the Programming Fundamentals series at JSecademy.

- Does the book cover advance topics such as streams, web servers, async programming, custom modules, IO read and write, setting up databases, or testing Node.js code?

No, the book is created for new programmers in mind. Topics that will be covered are Node.js basics (History, and Application), the Event Loop, npm, setup and install of Node.js in (Windows, OSX, and Linux), and executing a simple application.

Prerequisites

Before getting started with this book you need to have a clear understanding of the following tools to enable you to be successful with this book.

- Modern Development Environments

If for what ever reason you are not completely comfortable with the above requirement, I have provided for you some resources to get you going before getting started with this book.

- “Hello” Environments | Learn Modern Development Environments

Throughout this series, you will learn the in’s and out’s of multiple development environments, text editors (Vi, Vim, Atom, Sublime, Brackets), understand how to work with local, virtual and private clouds (VirtualBox, Vagrant, Cloud9), you will also learn how to automate the most common task using the command line. But most importantly you will have a good understanding of what to look for when you are creating your *“ideal”* development environment.

Code Conventions

All of the code samples are released under the MIT License. You are free to fork any part of the code and use it for any of your projects.

You can find the entire list of code samples in the following repository. Download Code

Tips Conventions

Pro Tip

Pro Tips contain information that is well established within the professional development group of programmers.

Suggestion

A suggestion is an opinionated or suggestive comment toward a specific topic.

Warning

Warnings are well established problems or common errors.

Introduction to NodeJS

In 2008 Google released the first version of the V8 Javascript Engine and for the very first time developers would be able to write JavaScript code that could be executed on the server side and client side. Allowing for the development of applications only developed in Javascript.

According to the main site for Node.js it defines it's self as follows.

> Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices. src

Ryan Dahl is the original contributor to the Node.js project, he was working on this very same idea since 2006. What idea? What he wanted was to create a web server that could handle multiple connections at the same time and be able to respond accordingly.

In 2006 he started to create this project first by using ruby and improving it with C but he hit multiple dead ends due to the constraints that the ruby interpreter had at that time. When he heard about the V8 runtime, he saw the opportunity and took it.

Ryan left his job and got to work on Node.js for 6 straight months. He requested a slot for JSConf and he got his chance to present the Node.js project with only **400** lines of JavaScript code he created an IRC server for his demo being powered by Node.js and amazingly enough it worked. All of the audience was astonished and has since then experienced significant growth, popularity and adoption.

If you would like to know more about the story behind node, I would recommend you to check out Ryans talk.

Basics

JavaScript also known as ECMAScript is a cross-platform, object-oriented dynamic programming language. It is a small and lightweight language. Inside a host environment, JavaScript can be connected to the objects of its environment to provide programmatic control over them asynchronously.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as logical operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects.

Node.js utilizes the JavaScript language to execute programs. In most common cases you will find Node.js on a server, executing some type of JavaScript code.

Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

In other words `Node.js` is a set of libraries for JavaScript allowing it to be used outside of the browser. It is primarily focused on creating simple, easy to build network clients and servers.

In 2016, Node.js is definitely moving away from the original idea of just serving web pages. You can find Node.js in some of the hottest startups, to projects that are preparing for the internet of things.

Here is a list of some of the benefits that the Node.js provides for you.

- Asynchronously programming I/O with an event loop
- Database Support (NoSQL, SQL)
- Concurrent connections, great for realtime applications
- Active development/community
- Easily work with JSON
- Motivates you to write unit test

Sometimes it takes others to take the leap of faith with a new technology, here is a short list of companies developing software applications using Node.js.

Companies

LinkedIn

When LinkedIn went to rebuild their Mobile application they decided to use Node.js for their Mobile application server which acts as a REST endpoint for Mobile devices. Their engineering blog talks about the Mobile stack. src

Nextflix

Netflix started to move there platform over to Node.js, some of the reason for this was in order to be able to more quickly develop and deploy varies parts of the software. src

Paypal

PayPal began migrating some of their Java, and C++ code to Node.js. They began to create some small applications to test Node.js and after seeing an increase in development time and quality, they started moving their entire site to Node.js. src

Uber

Uber's fundamental service of letting anyone have a private driver is being powered by Node.js due to the data intensive realtime services required by the application. src

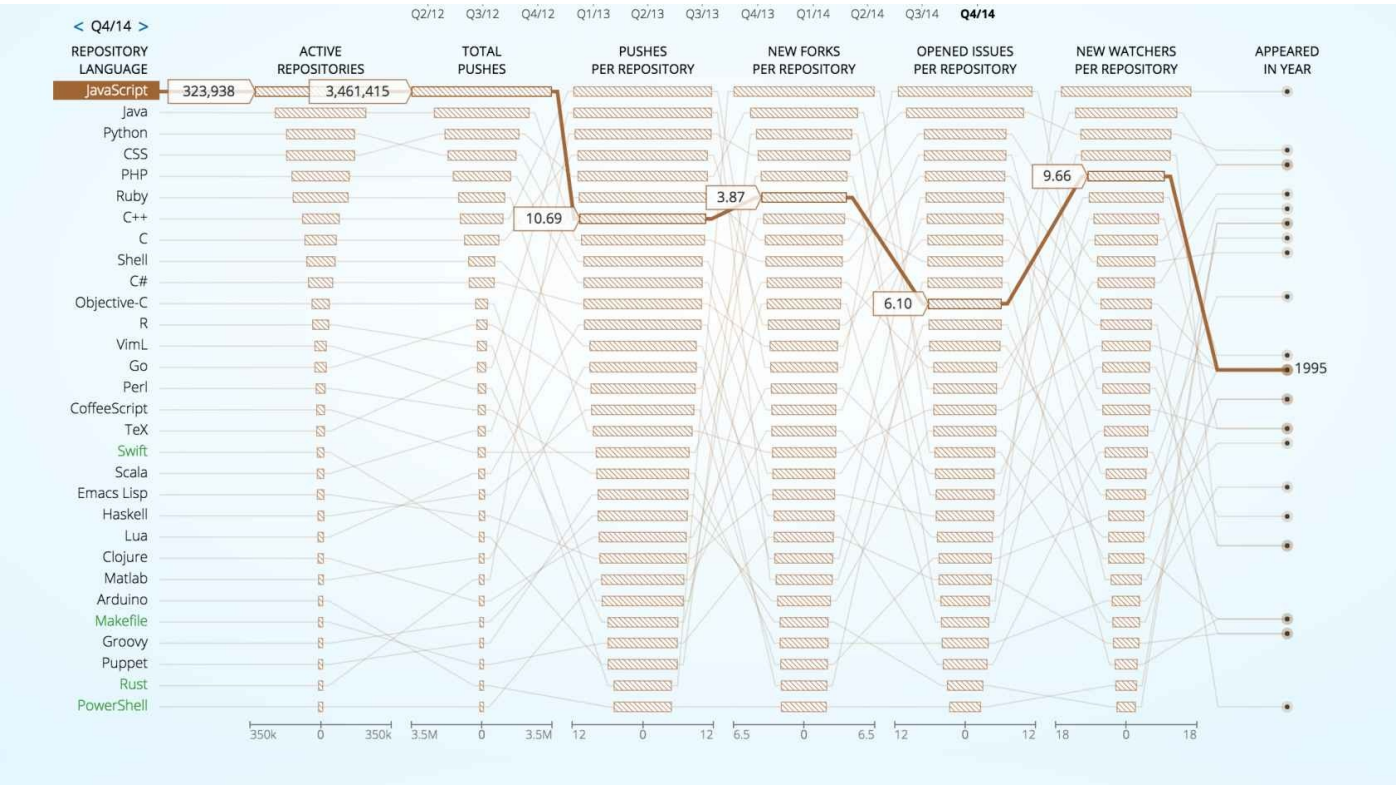
Walmart

Walmart needed to start migrating off of the Java platform they started to slowly implement a proxy with node and started to redirect request depending on what phase of the migration that they where in. This allowed them to still have legacy code as they started to migrate over services to Node.js. src

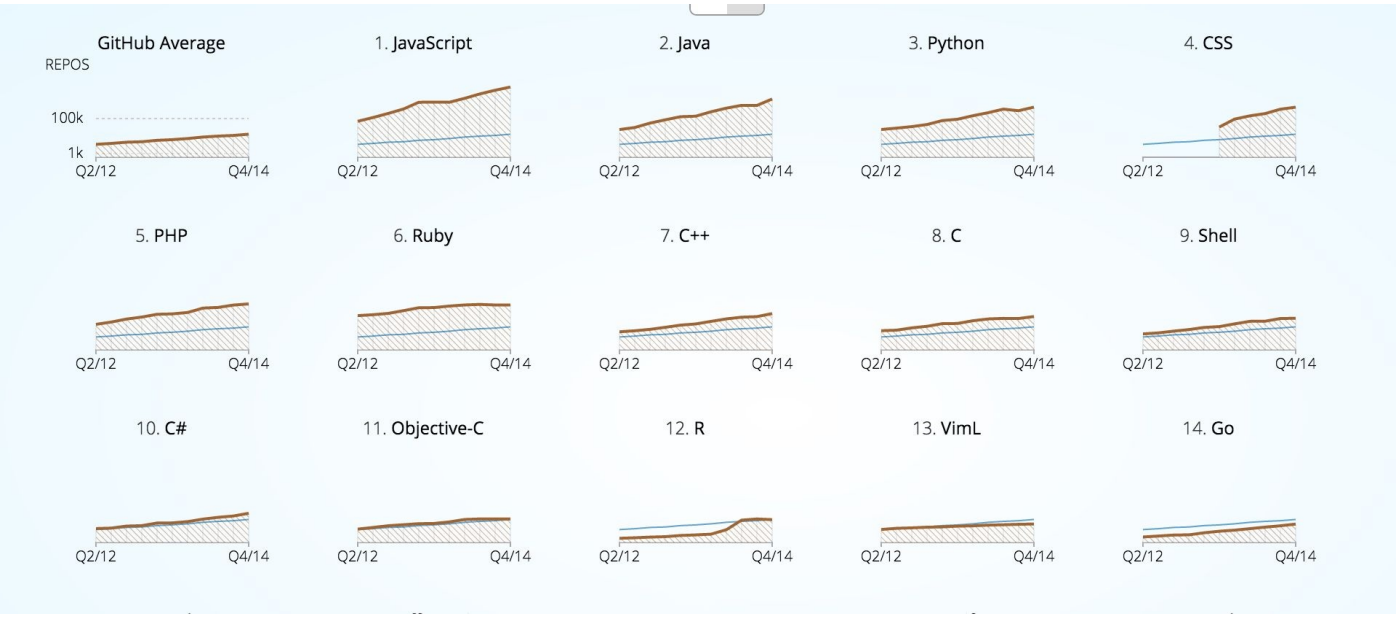
You can find a more extensive list of companies, and applications using Node.js in production.

How viable is the Node.js?

When exploring a new technology it's always a good idea to take a look around and see what others are doing with that exact technology. Githut provides a visual of the complexity of programming languages across 2 million active repositories hosted on GitHub. The number one most active language is **JavaScript** with Java not falling that far behind.



JavaScript 1



JavaScript 2

Pro Tip

Node.js will typically be referred to as *node*, you find evidence of this by the command that you use to run Node.js. Through out the book you will see it referred to NodeJS, Node.js or node.

Event Loop

Node.js is a single threaded system by design. If you don't know what threads or what even a single one will do. No problem. Just keep the following idea in mind when working with Node.js

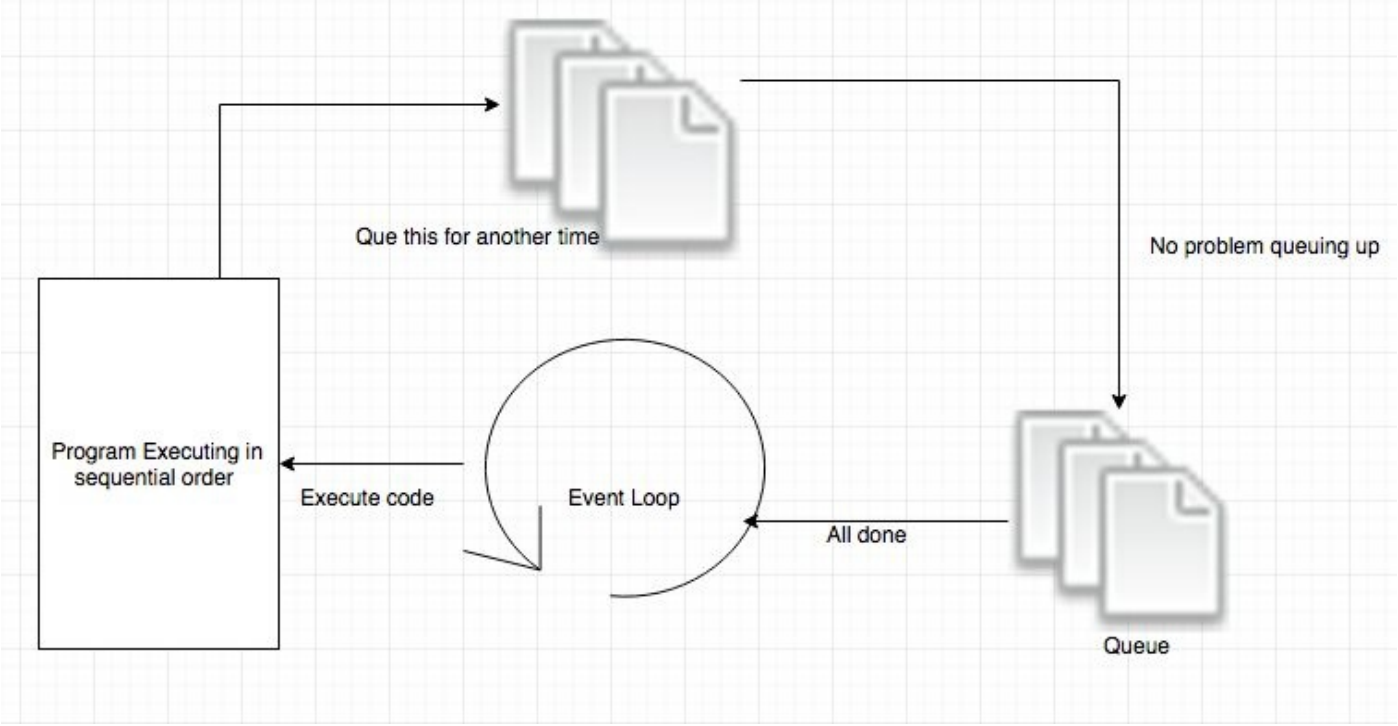
> Node.js can only do one thing at any given time.

This means that you can not do two things at the same time, one has to happen before or after another.

Synchronized Programming: Each of the lines of the program have to execute one after another. This means you can easily run into the problem of blocking your entire program if you are doing something that requires the use of the entire thread.

Unsynchronized or Asynchronous Programming: None of the lines of the program need to run in a specific order. This means that your program can run at different time.

Why is time important?: Time is important to both sync and async code, this is due to the nature of how the code is executing.



Event Loop

Now the reason why time matters to Node.js is because node uses both sync and async functionality when executing code. If you create a program that executes in sequential order all of your code will have to execute line by line. If you create a program that executes in unsynchronized manner then the code can run at different times and not be dependent on other parts of the program.

What's better async or sync? This will depend on what type of program you are writing, Node.js provides you with the functionality to write both at the same time.

npm

npm logo

> Contrary to the belief of many, “npm” is not in fact an abbreviation for “Node Package Manager”. It is a recursive bacronymic abbreviation for “npm is not an acronym”. (If it was “ninaa”, then it would be an acronym, and thus incorrectly named.) – Issac Z. Schlueter

npm is a package manager that runs through the command line or on the web. npm is used as a dependency manager, that allows programmers such as your self to create packages, share packages, and install packages from other programmers. As of this writing their are a total of *254,425* packages and growing at a rate of *407* per day.

Understanding packages: Packages are a set of functionality that is encapsulated into a specific package. This functionality can be a wide range of things from calculating precise number calculations to integrating another functionality that someone else has built.

Finding packages: To find packages with npm run the following command or search the index of npm

```
1 npm search numbers
```

Installing a packages: Once you have found the package that you want to use with in your application you can use the following command to install the package dependency for your program.

```
1 npm install numbers
```

Once the package is installed you can start using it within your program.

Remove Packages: To remove packages simple run the following command replacing the value of the name with the package that you want to remove.

```
1 rm -R node_modules/numbers
```

Node.js setup and Install

In this section, we will actually take a look at how to install, uninstall, and update Node.js in your specific operating system.

By the end of the chapter you will be able to:

1. Configure Node.js in Windows
    - Install Node.js
    - Update Node.js
    - Remove Node.js
2. Configure Node.js in OSX
    - Install Node.js
    - Update Node.js
    - Remove Node.js
3. Configure Node.js in Linux
    - Install Node.js
    - Update Node.js
    - Remove Node.js

Node.js is currently supporting the three main operating systems which are Windows, OSX, and Linux with a pre-built installer. Depending on what operating system you are using you can jump to that section and ignore any of the other sections.

Pro Tip

Node.js is typically deployed in Linux machines.

Understanding Node.js Versions

Node.js uses semantic versioning to control the versions of Node.js. Semantic versioning works with the following pattern.

MAJOR.MINOR.PATCH

Now let's take a look at one of the builds for Node.js.

Node.js Build

```
1 v5.9.0
```

1. `5` denotes that it's a major release
2. `9` denotes that it's a minor release
3. `0` denotes that a patch was applied.

The last part is to take a look at what stable, lts, and nightly are used when looking at Node.js.

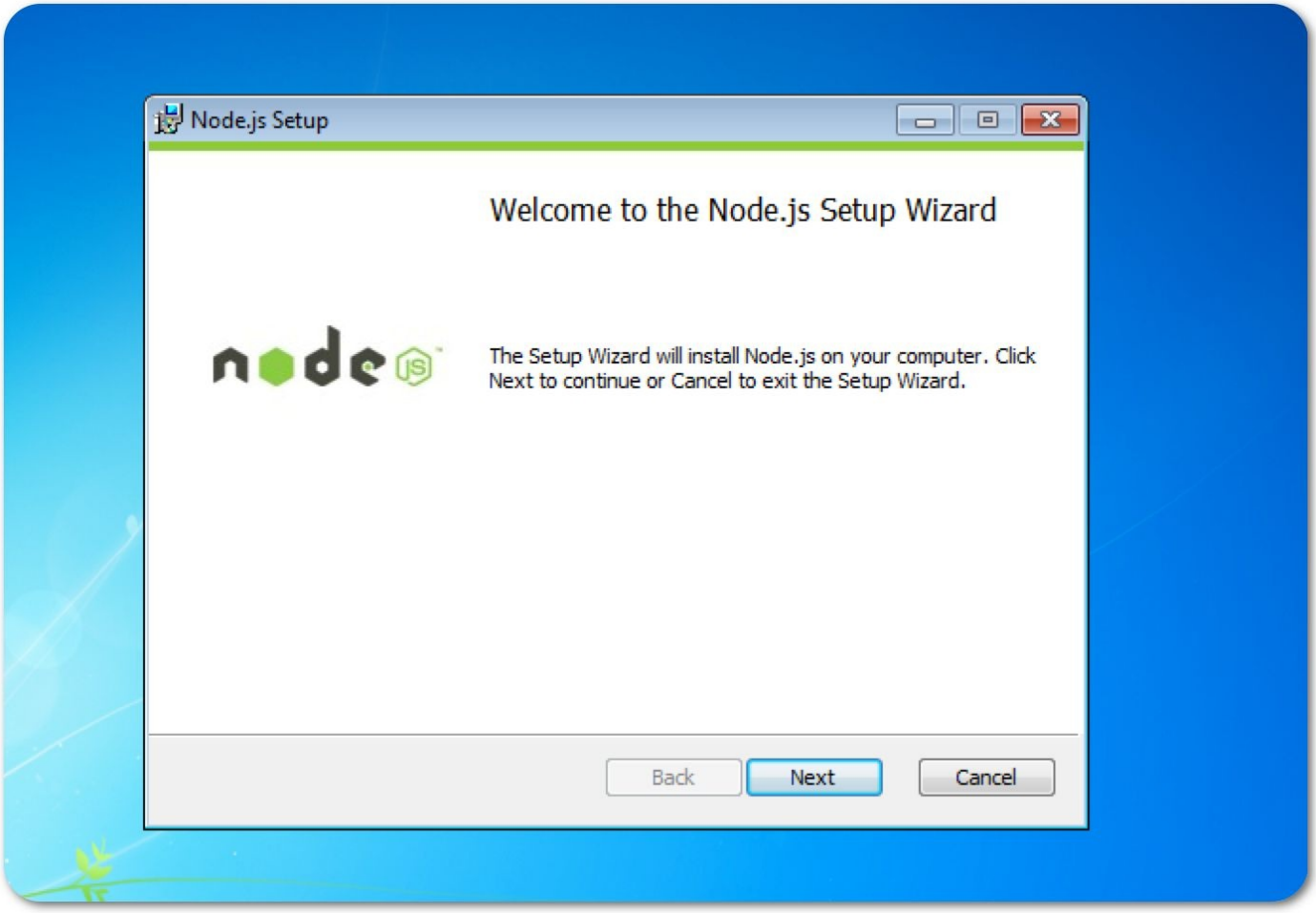
1. `stable`: Software releases have been verified by core team members as a stable release from development.
2. `lts`: Software releases that have LTS(Long-term Support) are typically supported longer then the stable releases
3. `nightly`: Software builds that are created every 24 hours.

Now each of this builds mean something very specific for Node.js you can find out what that is by visiting the Release Types of node.

What happened to node versions `0.y.z`? If you have tried to use node previously to this book you might have encountered a version of node in the `0.y.z` version. This is mostly due to a team of developers that decided to fork off to new project known as io.js. Both of this projects are now merged into a single project known as Node.js as of `Sept. 8, 2015`. You can find a complete release notes, where each version and date released are documented.

Why are Node.js releases named after elements in the Periodic Table? All releases that are under LTS support are assigned a *codename* from the Periodic Table by convention.

Windows



Node.js on windows

Installing Node.js

Start by visiting the main site for node at [NodeJS.org](), and click on the `stable` download button this will initiate the download for a file name `node-v5.9.0-x86.msi`, double click on the `.msi` file and this will begin the Node.js wizard follow the instructions on the prompt and wait for it to be installed.

Once node is installed you can verify that node is setup correctly by checking what version of node is installed with the following command. Open up the command line and use the following command.

```
1 node -v
```

Result `v5.9.0`

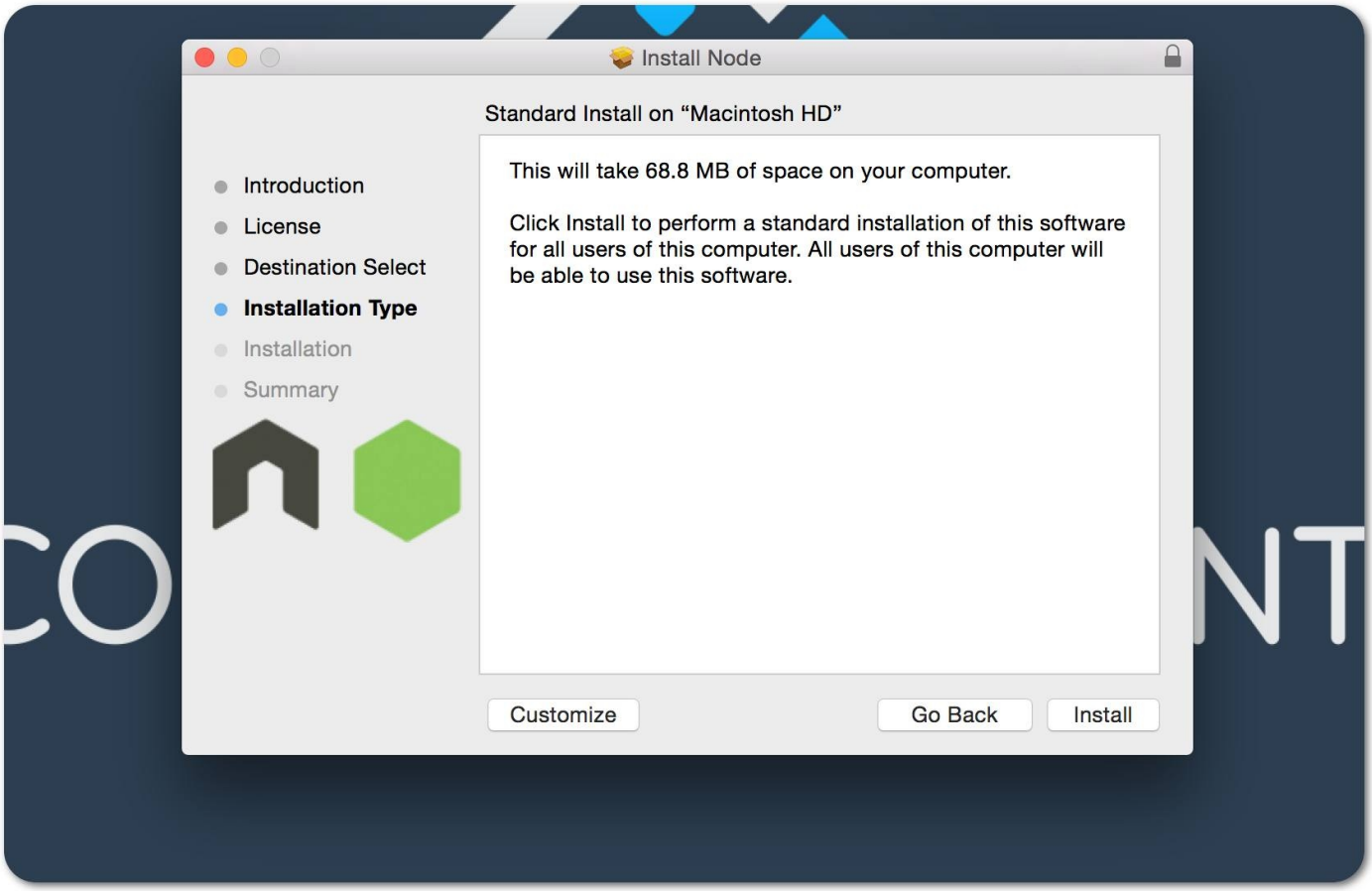
Updating Node.js

In windows there is no package manager. This means the process of updating node is simple enough, you uninstall the program and download and install the new version form the main NodeJS.org web site and you will be back up to the latest version of Node.js.

Removing Node.js

To remove Node.js from windows you will have to use the built in *"Programs and Features"* window to uninstall the node program.

OSX



Node.js on OSX

Installing Node.js

Start by visiting the main site for NodeJS.org, and click on the `stable` download button this will initiate the download for a file name `node-v5.9.0.pkg`, double click on the `.pkg` file and this will begin the Node.js wizard follow the instructions on the prompt and wait for it to be installed.

If you are using Homebrew, to install packages you use the following command to install node.

```
1 brew install node
```

After the install is complete it will install node at `/usr/local/bin/node` and npm at `/usr/local/bin/npm` and now you can double check if this directories are on your path by using `echo $PATH` on your terminal, if it’s not their make sure you at to this file `vim /etc/paths` and now you will have a working version of Node.js.

Once node is installed you can verify that node is setup correctly by checking what version of node is installed with the following command. Open up the terminal and use the following command.

```
1 node -v
```

Result `v5.9.0`

Updating Node.js

To update node you have two options. 1. If you installed node with a binary installer, then you will need to uninstall node from your computer using the command below that removes node with the `rm` command this will permanently delete node from your computer. Then you will have to go out and download and install the latest version of node.js. 2. If you used Homebrew to install node you can update node by running the following command. `brew upgrade node`

Removing Node.js

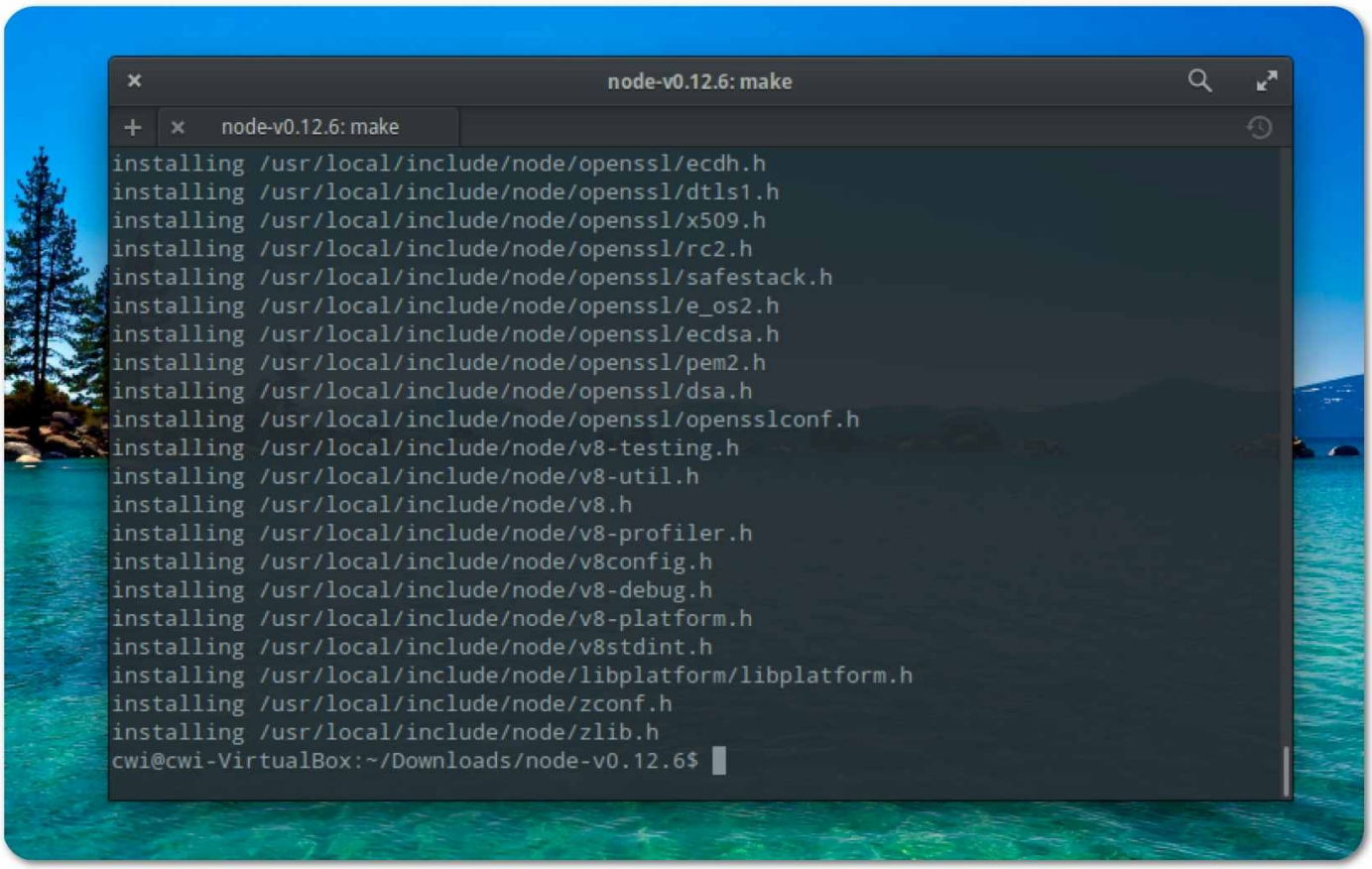
If you have installed Node.js with the binary executable from Nodejs.org you might find out that the application is not found in the Applications folder, this is because node is installed in a different directory. If you want to remove all the components from Node.js in OSX run the following command.

```
1 sudo rm -rf /usr/local/{bin/{node,npm},lib/node_modules/npm,lib/node,share/man/*\
2 /node.*}
```

If you have used Homebrew to install node then use the following command to remove node.

```
1 brew uninstall node
```

Linux



Node.js on Linux

Installing Node.js

Start by visiting the main site for Node.js and click on the `stable` download button this will initiate the download for a file name `node-v5.9.0.tar.gz`, to install node run the following commands.

```
1 tar -xf node-v5.9.0-linux-x64.tar.xz
2 sudo mv node-v5.9.0-linux-x64/bin/* /usr/local/bin/
3 sudo mv node-v5.9.0-linux-x64/lib/node_modules/ /usr/local/lib/
```

Now you must be asking why I did not just use `sudo apt-get install nodejs`, well you see the packages for ubuntu or Debian tend not to get updated as frequently as you would like them to. This could cause errors in your programs. If you install it manually you will remove most of the headaches of having to deal with older versions of Node.js.

After the move command is complete it will install node at `/usr/local/bin/node` and npm at `/usr/local/bin/npm` and now you can double check if this directories are on your path variable by using `echo $PATH` on your terminal, if it's not their make sure you at to this file `vim /etc/paths` and now you will have a working version of Node.js.

Once node is installed you can verify that node is setup correctly by checking what version of node is installed with the following command. Open up the terminal and use the following command.

```
1 node -v
```

Result `v5.9.0`

Updating Node.js

If you would like to update Node.js you will have to either version off the versions of node that you want to keep on your machine or remove them. If you end up removing the versions of node then you just simply delete the binaries fund in the removing node sectio and then install the new version of node from a fresh download.

If you decide you want to work with different versions of Node.js, I would recommend you to section off each of the node verisons in you /usr/local/bin/ and path variable set correctly.

Removing Node.js

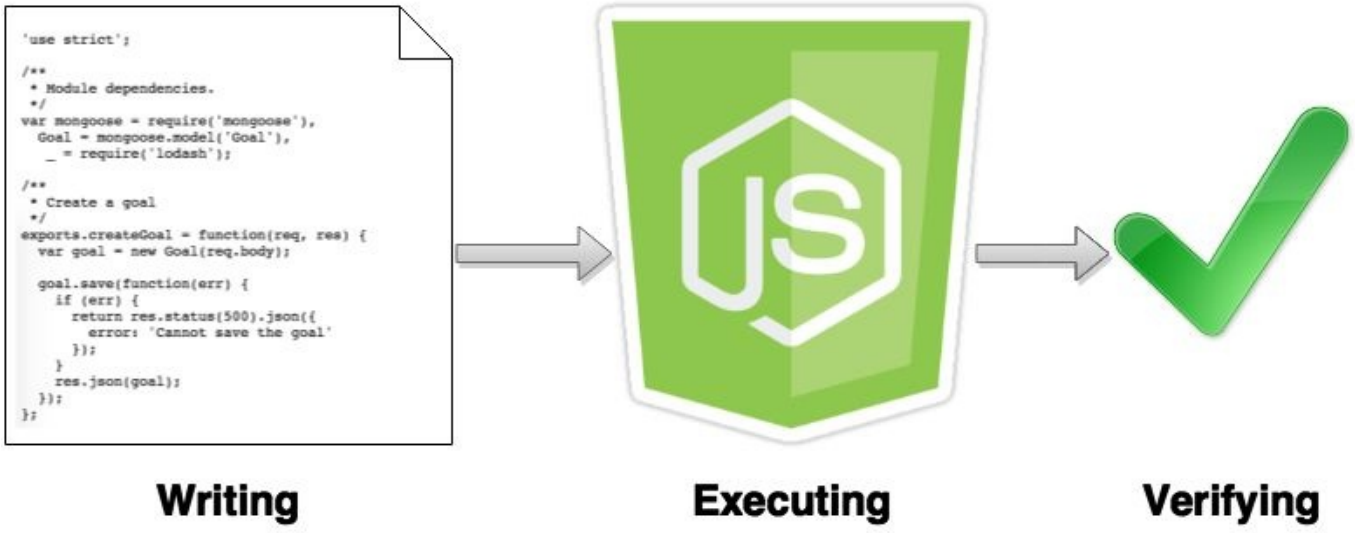
To remove node from your operating system run the following command. This will delete the node binaries files. If your linux distribution can not find node with the following command you will have to find out where you have installed node and remove it from that directory.

```
1 sudo rm /usr/local/bin/{node,npm}
```

Programming Workflow

In order for you to remain focus on the problem that you are solving it's a good idea to create all your JavaScript programs that you will be creating with no UI, while you learn to program. With Node.js you can create an application and run it with out a browser. This will allow you to quickly verify that the code that you write is working accordingly.

The work flow for this is a simple and easy one to follow.



Workflow

1. Write the code in a file with an extension of `.js`, e.g., `app.js`.
2. Let Node.js know about your file by executing it. `javascript node app.js`
3. Node.js will terminate after your script has completed executing.

Pro Tip

You can give the file any extension or no extension. The reason why .js extension are typically used is because will provide your text editor information to apply a specific format to a specific file type.

```
// Node will run both files
node app.js // Preferred Extension
node app
```

Sample Application

Now that you understand the core concepts of Node.js let's now create a simple application and execute it with Node.js with the terminal.

```
touch app.js
echo "console.log('(7 + 5) / 2 is ?', (7+5)/2)" >> app.js
node app.js
```

Results `(7 + 5) / 2 is ? 6`

Working with REPL

```
1 node
```

`node` opens the interactive shell (also called REPL) a perfect place to test simple one liners in JavaScript, to get out of it simply press `Ctrl + C` twice.

```
1 node
2 > console.log('(7 + 5) / 2 is ?', (7+5)/2);
```

Prints to the console and evaluates the expression

Results `(7 + 5) / 2 is ? 6`

Give some of the following commands a try with node.

OPTIONS

```
 1 -v, --version          print nodes version
 2 -e, --eval script      evaluate script
 3 -p, --print            print result of --eval
 4 -i, --interactive      always enter the REPL even if stdin does not appear to be\
 5  a terminal
 6 --no-deprecation       silence deprecation warnings
 7 --trace-deprecation    show stack traces on deprecations
 8 --throw-deprecation    throw errors on deprecations
 9 --v8-options           print v8 command line options
10 --max-stack-size=val   set max v8 stack size (bytes
11 --enable-ssl2          enable ssl2 in crypto and https modules
12 --enable-ssl3          enable ssl3 in crypto and https modules
```

You can find the above material plus all of the `V8 OPTIONS` that are available on github or if you are on a *nix system just use the man pages.

```
1 man node
```

Moving Forward

You are well on your way to becoming a **GREAT** programmer. You have successfully completed this book. This is a **BIG DEAL!** You need to celebrate. This is only the beginning of something great, if you keep working on developing your Node.js skills you will soon enough be creating your own applications using Node.js.

I want to hear about your success with Node.js. I welcome you to write into rick@codewithintent.com.

New Node.js programmers from all around the world need your help! Can you please go onto Amazon.com, JSecademy.com, or wherever you purchased this book and leave your feedback about the book. This will help new programmers decide if this book is right for them.

Leave a review:

- Amazon
- JSecademy

Thank you for giving me the opportunity to show you around the world of Node.js.

Best wishes,

Rick Hernandez.