"HELLO" ENVIRONMENTS

Learn Modern Development Environments

Rick Hernandez

”Hello” Environments

Learn Modern Development Environments

Rick Hernandez

This book is for sale at http://leanpub.com/hello-dev-environments

This version was published on 2016-01-30

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.

Tweet This Book!

Please help Rick Hernandez by spreading the word about this book on Twitter!

The suggested hashtag for this book is #jsecademy.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

https://twitter.com/search?q=#jsecademy

Contents

Introduction

I want to take this time to say **thank you** and congratulate you for taking the first step to creating the perfect development environment. This book contains solid techniques on how to setup your ideal development environment. Throughout this book, 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.

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 200 technical publications published at Code With Intent[1]. Rick has a deep passion and curiosity to help others succeed.

Say Hello

My intention with this book is to get you up and running with your ideal development environment. If for whatever reason you might have a question, comment or suggestion about the book, I encourage you to take the time and write into Code With Intent[2].

[1] http://codewithintent.com

[2] http://codewithintent.com/contact/

JSecademy.com 30 Day Free Subscription Bonus

Deciding to take advantage of the 30 day subscription period of JSecademy is a great decision. Not only will you get step by step videos walking you through the entire book, you will also receive help from a community of people wanting to learn more about Javascript and software development. To get started, follow the simple steps below.

1. Sign up for JSecademy[3]
2. Redeem Your Promotion[4]
3. Enjoy a 30 Day Subscription

What is a Transaction ID? Depending on where you purchased this copy of the book, e.g., Amazon, Leanpub, etc., you should have received an `order number`. **This is the Transaction ID**. Transaction ID's typically take 24-48 hours to validate. Get started with this right away

Happy Learning

[3] https://www.jsecademy.com/register
[4] https://www.jsecademy.com/settings/promo

Who this book is for?

The purpose of this book is to help you learn to create your ideal development environment. In the process of this learning you will have to explore different topics and be willing to look into different editors and not just be stuck with your 1976 text editor.

If you have ever had any of the following questions and did not know how to answer them, then this book is for you.

> *"How can I setup my development environment for programming with JavaScript?"*
>
> *"What's all the buzz about Vagrant?"*
>
> *"How can I get my entire team setup using the same development environment?"*
>
> *"What's the best web browser?"*
>
> *"Why don't JavaScript developer show any love for Windows Developers?"*
>
> *"What is the difference between Cloud Environment and a Virtual Environment?"*
>
> *"I don't want to use Linux! Can you show me Windows?"*

By the end of the book you will be able to answer all of this questions.

I assure you that if you have perseverance, dedication, and creativity to finishing this book, you will be off to having an awesome development environment that all your coworkers will be jealous of.

Prerequisites before getting started

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

- Basic command line experience

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

- Bash Basics[5]

[5] https://www.youtube.com/watch?v=x73WTEltyHU

- The UNIX Operating System[6]

Software and Hardware are constantly evolving this book uses the following operating system's as a baseline.

Software

- OSX 10.10 or higher
- Ubuntu 12.04 or higher
- Windows 7 or higher
- Modern Browser (Firefox, Chrome, Safari)

Hardware

- Minimum of 2GB of free memory
- 4GB of free hard drive space (optional)

This book will only focus on the mentioned operating systems, if you are running a earlier version of one the mentioned operating systems, this guide will need to be modified to work with that operating system. I would recommend you to just read the abstracts of every chapter and get an idea for what you need and then move on.

Code Samples

The book contains a variety of code samples and snippets that are used to illustrate a specific part of the book. None of the code builds upon each other, feel free to fork any part of the code and use it for your own setup.

All of the snippets used throughout the book are released under the MIT License[7].

The complete list of the snippets can be found on github[8].

Book Conventions

Pro Tip

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

[6] https://www.youtube.com/watch?v=tc4ROCJYbm0
[7] http://opensource.org/licenses/MIT
[8] https://gist.github.com/rick4470

Suggestion

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

Warning

Warnings are well established problems or common errors.

Code Examples

```
1 sudo apt-get update
2 sudo apt-get upgrade
```

What to expect out of the book?

This book is used as an educational mean, to convey one main message.

> **Setting up the perfect Development Environment**

This book is **NOT** here to teach you how to use this tools, some of this tools take years to master. Each of the tools that I cover, have a wealth of resources where you can learn more about them.

I will not be covering shortcuts, the command line, system administration, or macros. This book is to help you understand what tools you need to make you a more productive developer, by helping you automate some of the systems that help you create software.

Pro Tip

Remember: The key to being a **GREAT** programmer is being able to learn new skills.

Questions and Answers

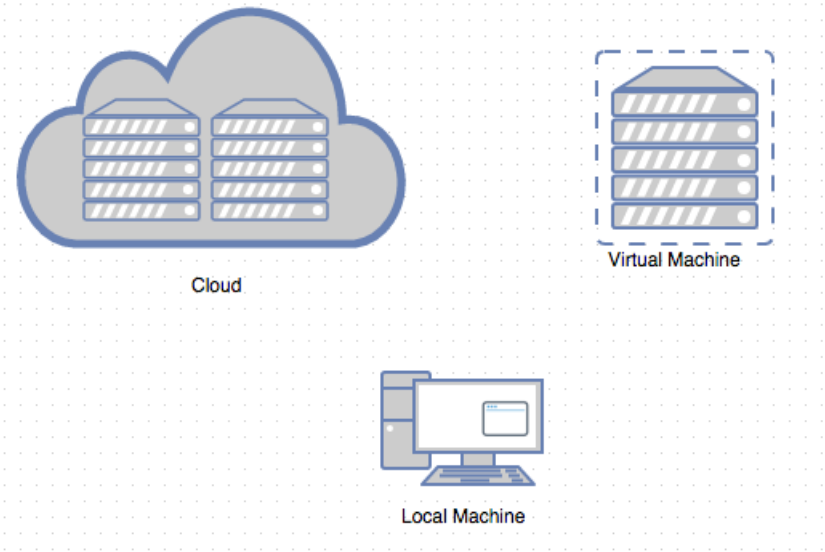
Is this book language agnostic? No, this book was design with one specific developer in mind, a JavaScript developer. If you are using another language such as C++, Java, Go, Perl etc this book is not for you.

Will you cover IDE's, for example Netbeans, WebStorm etc? Nope.

Does this book have a bias towards an operating system? Yes! This book does have a bias for ***nix** systems. Why? for me is due to my background in Computer Science and the bias from my instructors at my University, for others it could be a number of different reasons.

Will you cover how to use the command line? Nope.

Development Environments



Environment Types

The Development Environment that you decide to work on will dictate a large portion of you overall happiness as a programmer. From choosing the right platform to the right text editor, could be one of the reasons why you choose to keep working as a programmer or decide to move on to the next great thing. Depending on your current needs and budget this will vary to everyone's circumstances.

What is a Development Environment? Computer programmers require a set of specific tools when writing a software application. Your development environment is used to create software.

In this chapter you will be learning about the three different types of development environments that are the most common when working with JavaScript.

1. Local Environment
2. Virtual Environment
3. Cloud Environment

Remember depending on your specific circumstances, you will tend to lean towards a specific setup. Take for example the local environment, if you just want to get started with programming then this is the fastest way to do so.

If you are working with a group of teammates, you might have to look into a virtual or cloud environment. Now power up your desktop, or laptop and let's get started setting up your development environment.

The Basics

What is a Local Environment?: Typically, this will be the environment that is included with your operating system.

What is a Virtual Environment?: This Goes one step further then the local environment by installing a piece of software on the local environment to be able to virtualize instances of a machine onto the local machine.

What is a Cloud Environment?: This goes even further than the virtual environment, by completely eliminating the hardware and just provides you access to the machine over an SSH connection in order for you to configure your specific environment.

What is the difference between Cloud Environment and a Virtual Environment? Virtual and cloud environments are completely different concepts although cloud environments typically use some type of virtualization to provide you with a private environment.

What do Local Environments, Virtual Environments, and Cloud Environments all have in common? Environment Variables, which are a set of dynamic values that affect the way a program will behave on your computer. All of this variables are configured and managed by your operating system. The main reason why programmers use them is to be able to have a dynamic value that is able to be referenced.

Below you will find the list of the most common environment variables along with the location of where you can configure this variables values.

OSX

Variable	Description
$PATH	Semicolon delimited list of directories
$HOME	The path to the home directory

Linux

Variable	Description
$PATH	Semicolon delimited list of directories
$HOME	The path to the home directory

Windows

Variable	Description
%PATH%	Semicolon delimited list of directories

Windows

Variable	Description
%HOMEDIR%	The path to the home directory

Pro Tip

Remember: Environment variables can have scope and a value, just like any other variable.

Variable Location

Depending on what operating system that you are in the location of this variables will differ. The following locations are for global variables.

OSX: The default location for custom environmental variables in OSX the `.bash_profile` file.

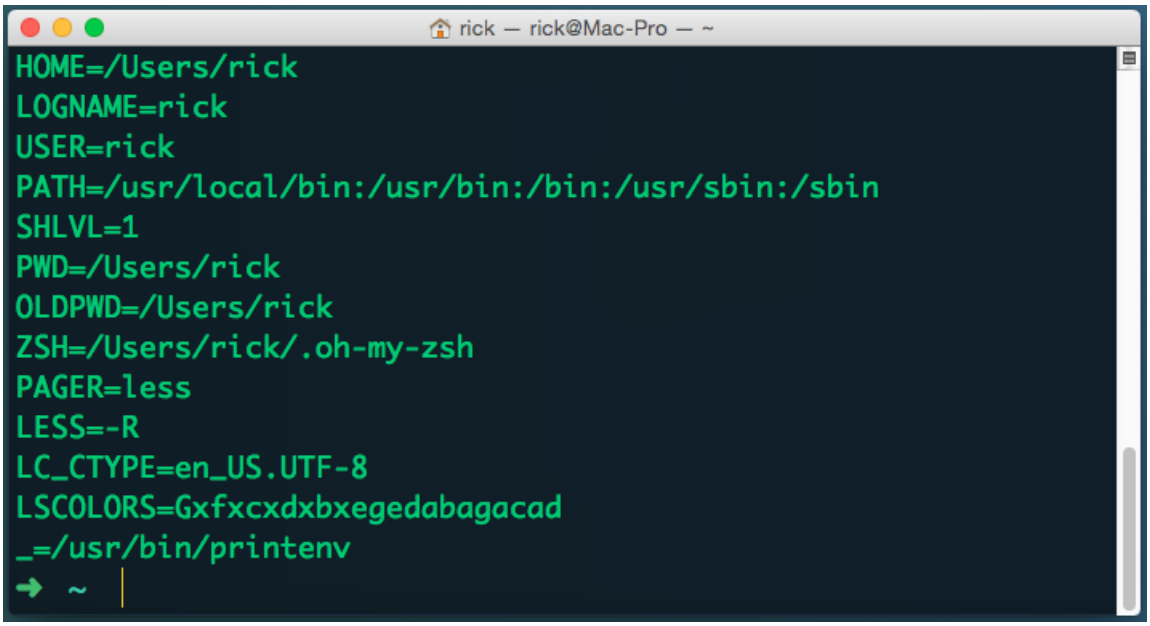
.bash_profile

```
1 $ ~/.bash_profile
```

To list all the variables that are being used by your system use the `printenv` command.

.printenv

```
1 $ ~ printenv
```



printenv

Linux: The default location for custom environmental variables in Linux is the `.bash_profile` file.

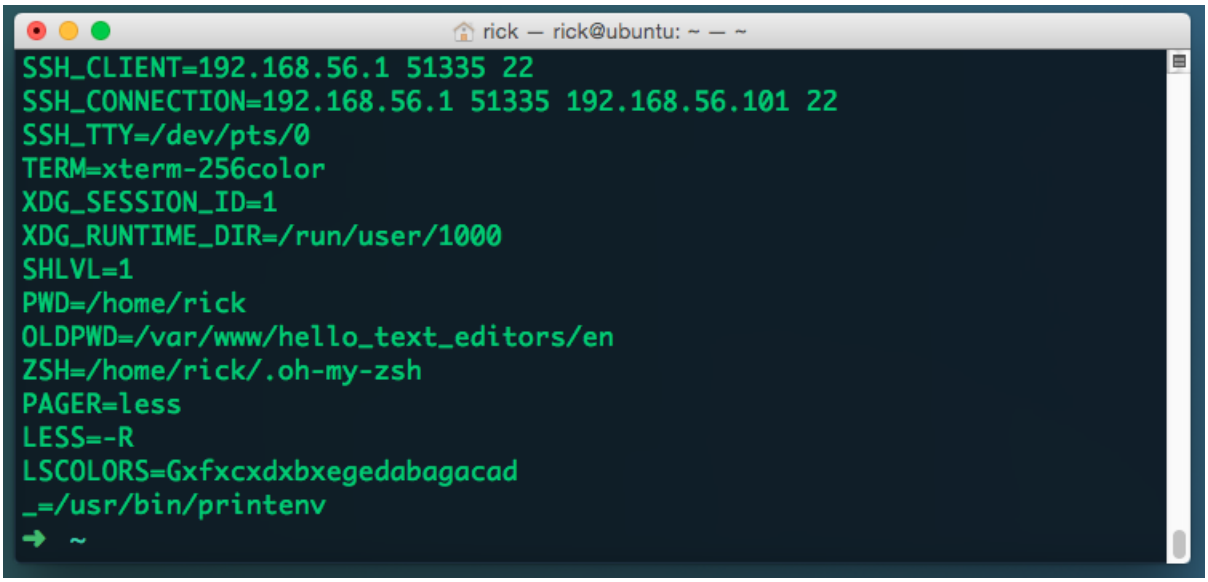
.bash_profile

```
~/.bash_profile
```

To list all the variables that are being used by your system use the `printenv` command.

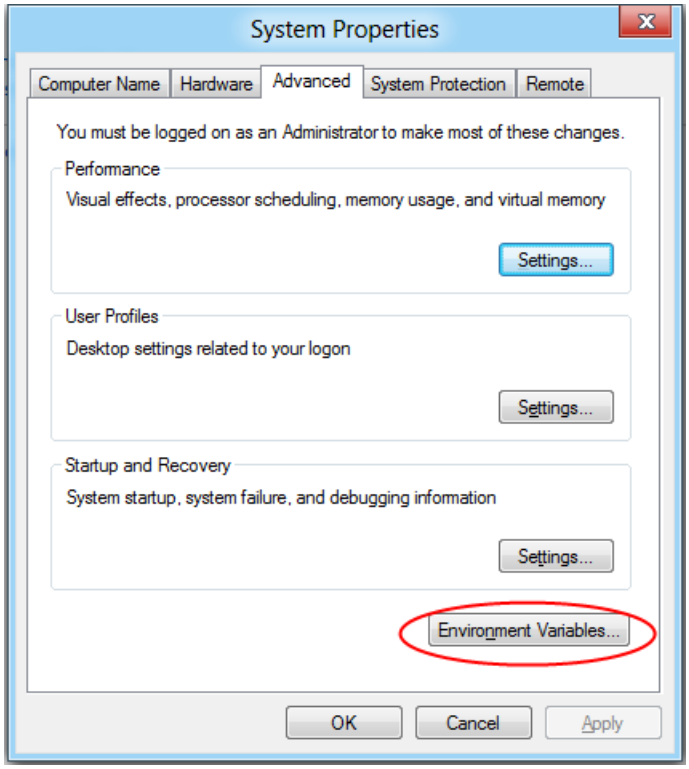
.printenv

```
$ ~ printenv
```



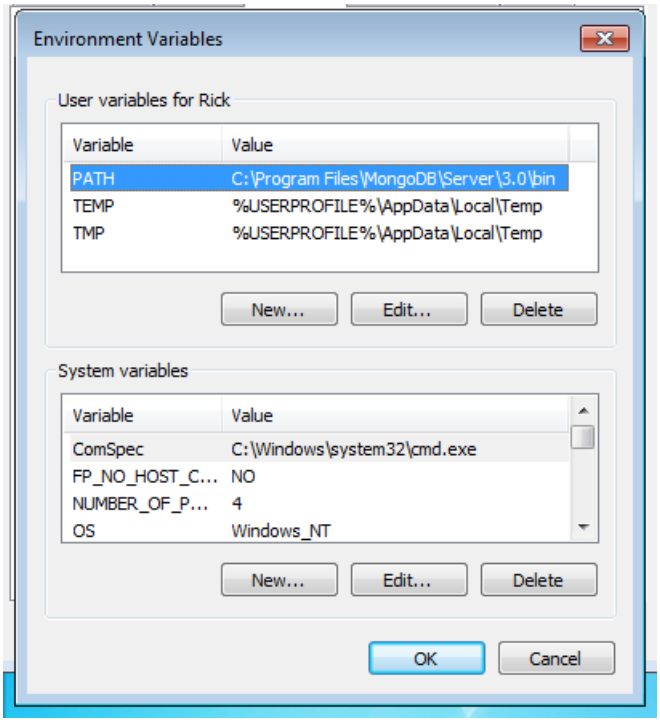
printenv

Windows: In Windows 7 you can find the location path for the variables in the following location.



Windows Location

1. Click the `Start button` and right-click the `computer` option in the `start menu`, and select `Properties`.
2. Click the `Advanced System Settings` link in the left column.
3. In the `System Properties` window, click on the `Advanced` tab, then click the `Environment Variables` button near the bottom of the window (pictured above).
4. You find the `Environment Variables` window (pictured below)



Environment Variables

Variable Assignment

When writing your application you might want to set or read an environment variable. One of the most common custom variables used in *ExpressJS* is the `NODE_ENV` variable. This is a custom variable that is used to distinguish between *production* and *development* environments.

OSX

Setting a variable: To set the variable use the command line.

NODE_ENV

```
1 $ export NODE_ENV=development
```

Reading a variable: To read the variable using the command line.

NODE_ENV

```
1 $ echo NODE_ENV
```

Reading a variable: To read the variable using JavaScript.

JavaScript

```
console.log(process.env.NODE_ENV);
```

Linux

Setting a variable: To set the variable use the command line.

NODE_ENV

```
$ export NODE_ENV=development
```

Reading a variable: To read the variable using the command line.

NODE_ENV

```
$ echo NODE_ENV
```

Reading a variable: To read the variable using JavaScript.

JavaScript

```
console.log(process.env.NODE_ENV);
```

Windows

Setting a variable: To set the variable use the command line.

NODE_ENV

```
set NODE_ENV=production
```

Reading a variable: To read the variable using the command line.

NODE_ENV

```
ECHO %NODE_ENV%
```

Reading a variable: To read the variable using JavaScript.

JavaScript

```
1 console.log(process.env.NODE_ENV);
```

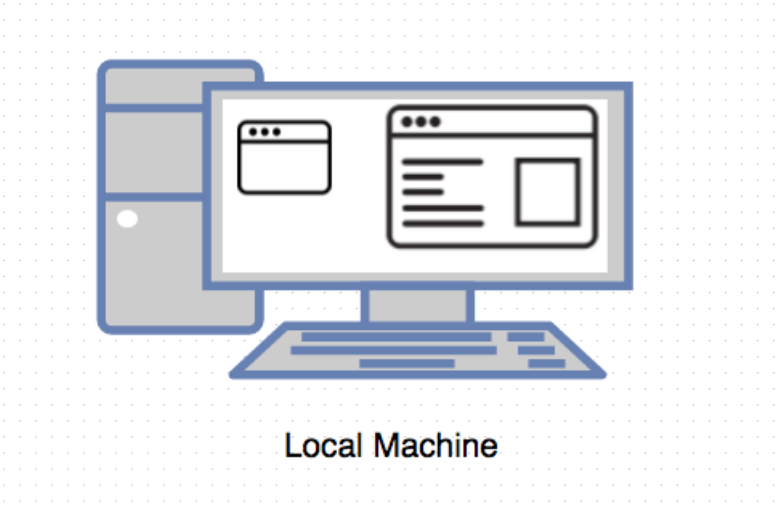
Warning

Environment variables optionally can have the value of `undefined`, always check the variable before using it in your program.

Summary

Remember it's your personal development environment, it needs to be as unique as you. You will be using it hours on end, if you decide to use windows great, if you go with linux great, or OSX then great.

Local Environment



Local Machine

One of the great things about working with JavaScript, is that you don't need to have a complex development environment. Most modern computers have a web browser and a pre installed text editor which is all you need to start to program using JavaScript. In this section we will cover how to create a simple program with this tools.

Modern Web browsers

Most modern operating systems support the ability for you to install and setup a browser. If you are running windows then IE is pre installed, if you running Linux you might get Chrome or Midori for OSX users you will have Safari pre installed.

The rest of this book will focus on using Chrome. If your computer does not have chrome installed you can download the latest version below.

Chrome Download[9]

Hold on! What about the rest of the browsers? Great question, this is one of the many decisions that you must take in order for you to create the *"Perfect development environment"*. Here is a list of the most popular web browsers for 2016.

1. Firefox
2. Chrome
3. Opera
4. Safari
5. Internet Explorer
6. Torch
7. Midori

Pro Tip

If you decide to be a web developer in the future you will have to support this list of browsers, and many more.

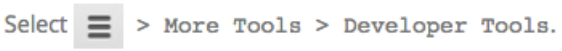
The JavaScript console

The JavaScript console provides you the ability to log messages, enter executable code or evaluate regular JavaScript expressions.

How to open the console? The JavaScript Console can be accessed in three different places.

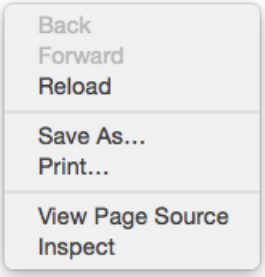
1. Use the keyboard
    1. Mac Command + Option + J (Mac)
    2. Windows and Linux Control + Shift + J (Windows/Linux).
2. Use the Menu

[9] https://www.google.com/chrome/

1. 

Menu

3. Use the Mouse
    1. Right click anywhere inside a webpage and select inspect



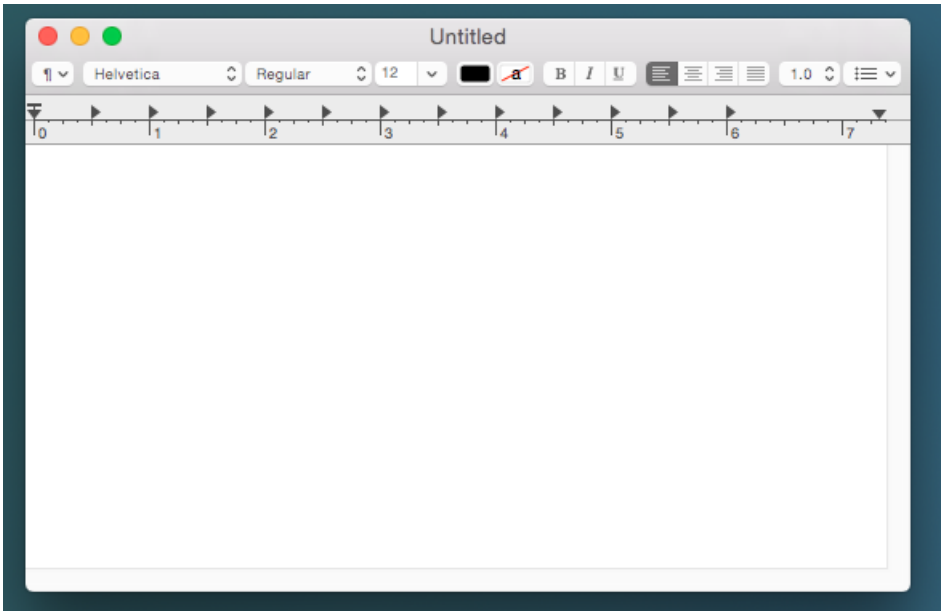
Menu Page

Now that you know how to open the console, this will be used later when creating your first JavaScript program.

The Text Editors

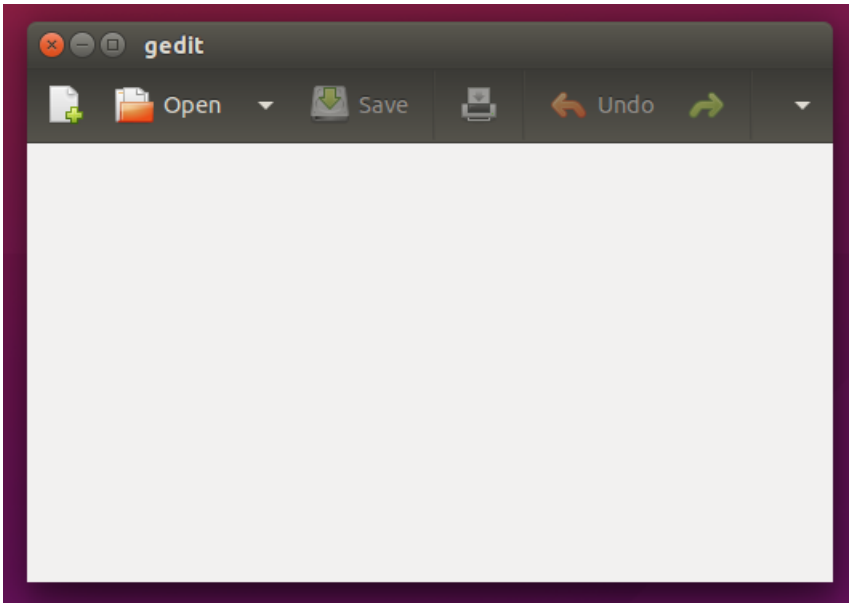
In your local environment, the operating system will provide you with a text editor that is pre installed. Below you will find the list of the most common text editors that are included in your operating system.

OSX: In OSX you will have an application named `TextEdit`



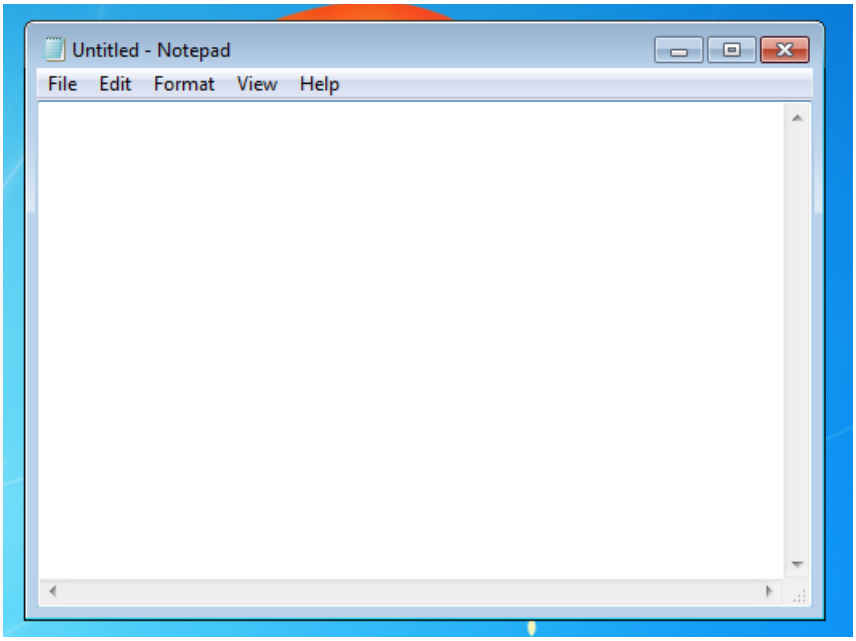
Text Edit

Linux: In Ubuntu you will have an application named `gEdit`



G Edit

Windows: In Windows 7 you will have an application named `Notepad`



Note Pad

Can I use Pages, LibreOffice or Microsoft Word to write code?: Yes and No, although each of this applications allow you to use them as text editors, there main purpose is to be used as a word processor not a text editor. The difference is that word processor will add extra information to the files that are saved, typically meta data such as formatting, font colors, font types etc. Your JavaScript programs wont require any of this. This another decision that you must make to find your *"Perfect development environment"*.

Running a simple program

Your browser requires an `HTML` page in order for it to be rendered inside the browser. JavaScript requires to be inside an `HTML` page in order for the browser to interpret the JavaScript Code.

hello.html

```
1 <script type="text/javascript" src="hello.js" async></script>
```

hello.js

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

Now just double click the `hello.html` file and open the console and you will have a message saying `Hello World`. That's all there is to running a program with your local environment.

Virtual Environment

Virtual Environment with VirtualBox

The main benefits of having and virtual environment to develop your application outside of having it installed in your main machine is that it gives you the power to experiment and try different experiments with out necessarily having an impact on your actual machine. Not only can you create multiple instances of the same environment you can quickly shut off machines and bring more them online, as your application requires them. This is great if one of your interest is scaling your application to multiple machines in a Virtual Environment, making it one of the first steps to moving on to a platform like AWS (Amazon Web Services).

Since the release of VirtualBox in 2007 developing applications has never been the same, if your anything like me you probably don't always want to be programming in the same language nor programming in the same machine, you want to see what happens when your run your application in multiple platforms. Virtual and Cloud Environments allow you just to do that for close to no cost. Cloud environments can be created for as low as $5 USD a month or Virtual Environments can be hosted on your machine for as much as your hardware will allow. In this section we will explore Virtual Environments and find out why and when you should use them.

The current market place has a wide variety of virtualization options that allow you to use your current hardware to virtualize a machine. In this section I will only be covering two of the most common tools used by JavaScript developers.

1. VirtualBox

2. Vagrant

Virtual Environment with VirtualBox

Why a virtual environment? I'm not going to go into the reasons on why VE is important or why you should be using it, instead I will show you how it relates to software development and why this is a valid option when choosing to go with a host or local setup.

Here is a quick list of some of the advantages.

- Multiple instances of the same operating system in different states.
- Quickly backup and restore instances, great for working with teams.
- Only run the virtual machine when it's needed instead of having daemons running all the time.
- Keeps your local machine clean.
- Adds another level of security incase your machine is stolen.
- Allows you to create test cases to actual domain names with host files.
- Allows for a sandbox where you can have machine to machine interactions.
- Cross platform development with VDI's.

VirtualBox is defined as follows: >VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2.

The main take away from VE is to understand that Virtual Environment's are just a tool to keep dependencies by different projects contained into a virtual machine.

Setting up VirtualBox