



<code />

2015

Installation &
Pre-work Guide

**Any sufficiently
advanced
technology is
indistinguishable
from magic.**

— Arthur C. Clarke

HOW DOES COMPUTER
PROGRAMMING WORK ?

MAGIC.



全玄鶴

Table of Contents

Click on any of the below links to go directly to that topic's page.

[Welcome!](#)

[Pre-Class Self-Study](#)

[Understanding the Command Line](#)

[Understanding Git](#)

[Basic HTML/CSS](#)

[Installation: You Can Do It!](#)

[Command Line Instructions](#)

[Create A Github Account](#)

[Create a Heroku Account](#)

[Mac Installation](#)

[Important Note](#)

[What You'll Need](#)

[Upgrade Your Mac OS to Mavericks \(10.9\) or Yosemite \(10.10\)](#)

[Installing Developer Tools for the Command Line](#)

[Installing Homebrew](#)

[Installing git](#)

[Installing Ruby and Rails with rvm](#)

[Installing Sublime Text](#)

[Windows Installation](#)

[Verify Your Installation](#)

Welcome!

This guide will introduce you to all of the strange and wonderful tools that you will be using as a developer.

This class will be teaching you how to be an awesome developer in just four short weeks. It will be rigorous, time consuming, and require a dedication to the craft. It will also require you to have fun! Programming can be very exciting.

Perhaps most importantly, you will leave your four weeks of class prepared to join a social improvement business in Cape Town. While there you will contribute real and effective code.

After that, everything is up to you! You can become a full-time developer or start your own organization.



Ruby convention logo

Please note that **everything in this guide is required before class starts!** There are many things a new developer must know and tools they must use. In order to focus on the code and design aspects of web development, you'll have to do some early work on your own. You will be far behind on your first day if you do not complete the pre-work self-study and the installation!

Don't worry though. There are plenty of helpful resources on the Internet to get you started. We won't leave you hanging!

Getting Help

We will be using Piazza for collaboration. It's a great tool for students and teachers to communicate, both before and during class. You will receive an invite along with this document. Join Piazza by following the link in the email. Please contact Seth (seth@ixperience.co.za) if you did not receive an invite.

If you are stuck during prework or installation and our suggestions for help haven't worked for you, post a question on Piazza asking for help. Teaching staff or another student will assist you. Be prepared to explain how you've already tried to solve the problem on your own.

Help out your fellow students if you know an answer! One of the best ways to learn something is to explain it to someone else.

Pre-Class Self-Study

This section of the guide will prepare you for all of the new information you'll be learning in class. You are not expected to be masters of everything in this guide (no one is) but a familiarity and comfort with these topics is very important.

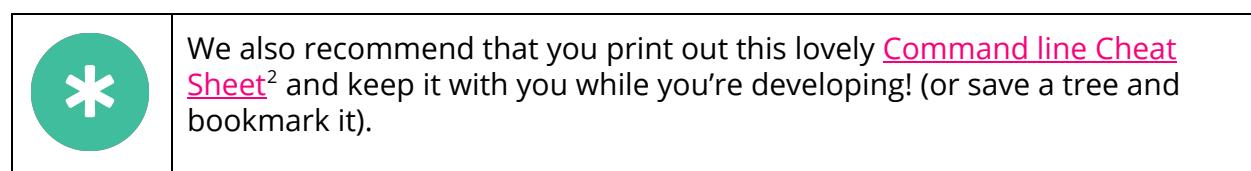
Understanding the Command Line

The command line is where most of the developer's magic happens. It's a text-based interface that allows you to quickly direct the computer to do things for you. Unlike the interface in your browser or your email client, there is nothing to click and no fancy features. It's just you, a black screen, and a big list of powerful commands at your finger tips. You may also hear it referred to as "the shell" or "the terminal", which means the same thing.

To get a good start on the command line, read the [Command Line Crash Course](#)¹. This short book will teach you everything you need to know for class in an hour or two. All of the commands are necessary to be a good developer, though you don't have to remember *all* of them every day.

It's very important that you have a beginner's understanding of the command line when you come to class. The commands you should *definitely* understand well on the first day of class are:

ls	rm
pwd	cp
cd	mv
mkdir	man



¹ <http://www.vikingcodeschool.com/web-development-basics/a-command-line-crash-course>

² <http://www.git-tower.com/blog/command-line-cheat-sheet/>

Understanding Git

Simply, **Git is a tool that developers use to back up the source code to their applications.** Just like you use something like Dropbox to back up your files and photos, we use Git to back up our code. Git uses a database to store all of its information, which is called a **repository**.

However, Git does much more than that! It also keeps track of the history of changes to your files. This means that as you are developing your application, you can create “snapshots” that will be stored in history. This way, if you change something you didn’t mean to, you can go back into history and retrieve it. These snapshots are called **revisions** by Git.

When developing applications, we spend a lot of time experimenting with the code. Git allows developers to feel safe to experiment with new ideas and make mistakes, without negatively affecting their current application. You can try all of the crazy ideas you have as much as you want, as long as you have backed up your code with Git!

Another important feature of Git is making collaboration between multiple developers much easier. Instead of sending files through email or some other method, developers can simply use Git to share code from the **repository**. If two developers make changes to a single file, Git will help those developers **merge** them together easily.

Unfortunately, Git can be very confusing and unfriendly to newbies (and even experienced developers struggle with it). It’s totally fine if you’re still confused about git when you get to class—we will talk about it a lot!

To learn more about git, follow through the tutorial at [Git Immersion](#)³.

³ <http://gitimmersion.com/index.html>

Basic HTML/CSS

One of the parts of coding web sites is understanding how to use HTML and CSS. We will not be teaching HTML and CSS in class, so learning them on your own is one of the most important parts of the pre-work! Luckily, they are relatively simple to learn.

For our HTML/CSS pre-work, we will be using General Assembly's Dash project. It is a great introduction to HTML and CSS with a guided approach. You won't have to install anything for this exercise, everything is done through the web site.

At the end of each lesson, you will be given a link to the project page that you just created. You can share this with friends and family, and show off that you're already doing web development! Most importantly—you can share it with us and we can check your progress before class.

Keep track of the links as you complete the projects, as you will be emailing them to us when you are complete!

To get started, go to [General Assembly Dash](#)⁴ and create an account, then do the following projects:

Project 1 - Build a Personal Website

In this exercise, you will learn basic html and css, and how to create your own personal site. You can use this for your pre-class presentation (described below).

Project 2 - Build a Responsive Blog Theme

Continuing on your lessons from the last project, you will learn to build a blog theme using basic HTML and CSS, and learn how to create web sites that look good on both a computer and a mobile phone browser.

Project 3 - Build a Small Business Web Site

One more step into development, you'll create a web site for a small business. The final project for this will look great!

If you want to, you can do the other two projects available on the site. They cover more advanced subjects like JavaScript, which you do not need for class (but it can't hurt!).

Pre-class Demonstration: Personal site!

Once you have completed these lessons, your homework will be to create a personal page using the last part of Project 1.

⁴ <https://dash.generalassemb.ly/projects>

Using the Project 1 template they provide, create a nice looking personal site that includes additional information for your South African adventure:

- Your name in big, bold letters!
- A photo of you
- Demonstration of custom CSS that makes your site beautiful
- Your school or home state
- An answer to these three questions:
 - Why do you want to learn to code?
 - What are you most excited about in South Africa?
 - What's one surprising or cool thing about you that others might not know?

Pre-class Demonstration: Quiz!

Now that you've made your own personal site, blog, and business site and have shown it off to the world (at least, to the teaching staff), it's time to take a quiz!

Fill out [this multiple choice questionnaire](#) by the two week start date of your class. Use your full name, and the password is **iXperience2015**. The teaching team will be reviewing it and will contact you about it before class starts.

Installation: You Can Do It!

It's challenging for beginners to install and configure all the tools we need for applications, but don't worry—we'll walk you through everything.

It's very important that you do your best to follow the instructions in this guide for your operating system before class begins! Do your best to come to class with everything installed correctly.

What happens if I get stuck?

Try to fix it! One of the most important developer skills is persistence in overcoming challenges. Programming is an endless series of problems to solve, and the best way to learn is to do it. When you try to solve a problem on your own, you will gain important knowledge. *And knowing is half the battle!*

Here's some ways you might try to understand and fix a problem:

1. Search the Internet using the error message if you have one.
2. Search with a more generic term like "homebrew installation problems mac".
3. Ask on [Stack Overflow](#)⁵, a popular web site for programmers to ask questions when they get stuck. See their [How To Ask](#)⁶ help section for tips!

I really can't figure this out!

If you run into issues that you just can't solve even with your best effort, post a question on Piazza and someone will help you! We will ask what you've already tried, so be prepared to explain the initiative you've taken to try to fix it on your own. See the section [Getting Help](#) above for more info.



⁵ <https://stackoverflow.com>

⁶ <http://stackoverflow.com/help/how-to-ask>

Command Line Instructions

For most of this installation, you'll be using the *command line*. The command line is also known as the Shell or the Terminal. It's a tool for developers to quickly give *commands* to the computer to tell it what to do, instead of having to open programs and click around.

Anywhere that you see text that looks like this:

```
$ brew install coreutils
```

It means that you should type it into your command line *exactly how you see it, except for the beginning \$*. The \$ is not part of the actual command, and is simply a way for us to designate that it's a one-line command. Punctuation, spelling, and quotation marks matter!

If you see a really long line like this:

```
$ command --argument --graph --online  
https://google.com/some/really/long/url/that/wraps/the/page
```

That all goes on **one** line!

	<p>How do I access the command line on Mac/Windows/Linux?</p> <p>If you don't know how to access the command line on your computer, visit Finding the Command Line in the Command Line Crash Course⁷.</p> <p>The command line is also covered in more detail in this document in the section "Understanding The Command Line"!</p>
---	---

⁷ <http://www.vikingcodeschool.com/web-development-basics/a-command-line-crash-course>

Create A Github Account

Github allows you to safely store your code online, much like Dropbox or iCloud allows you to store your files online. Github also has many other project management and collaboration features that make it a dream tool for developers!

If you do not already have a Github account, please create one at <http://github.com>.

Your Github repository is publicly available and most employers will ask you for your Github name so they can see what you've worked on. Pick a good name that you'd be happy to share with a boss, and choose a good photo!

Note: You will be using your Github username and password a lot during development, so it's very important that you remember it.

We will learn all about Github in class. To get started, check out [What Is Github and What Do Geeks Use It For?](#)⁸

Create a Heroku Account

If you do not already have a Heroku account, please create one at <http://heroku.com>.

Note: You will be using your Heroku username and password a lot during development, so it's very important that you remember it!

Heroku is a tool for beginning developers to easily put their web applications online for free. We will use it in class to make your applications available to everyone on the Internet who has the link.

We will learn more about Heroku during the last weeks of class.

⁸ <http://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>

Mac Installation

Do you have a Mac? Great, you're in luck! They're the easiest operating system to use for web development. The installation process will take a while, but once it's set up it's good for a long time, and it's unlikely that you will run into problems while developing.

Important Note

WARNING: It is **very important** that you do not use the "sudo" command at any time during installation, even if you find instructions on the Internet tell you otherwise. It can cause problems. Do not type "sudo" before any command.

What You'll Need

To install all of the software on your Mac, you will need a few things first.

1. **An Apple ID.** If you use the iTunes App Store or iCloud, it is the same Apple ID. If you don't have one, you will be able to create one.
2. **Your computer's Admin User login name** (you will use this a lot). If you log in to your Mac using a password, it is probably the same password.
3. About one hour of time.
4. Enthusiasm!

Upgrade Your Mac OS to Mavericks (10.9) or Yosemite (10.10)

Development on versions of Mac OS before Mavericks (< 10.9) may cause issues. If you are on an older version, please upgrade! You want to be on Mavericks (10.9) or Yosemite (10.10).

If you are unsure which version of Mac OS you have, click the Apple icon in the menu bar, then select About This Mac, and the version number will be on the small panel that appears (example below). You want to see 10.9 or 10.10 or above.



Where to find your Mac OS version number.

Installing XCode Command Line Tools

When developing on a Mac, there are certain developer tools that are needed to ensure your web applications can run on your machine. Mac OSX includes a program called XCode, which is primarily used as a code editor for building iOS and OSX applications. However, it also includes necessary development tools for building web applications, including Ruby on Rails applications.

In order to ensure your machine has all the necessary tools for development, you need to install the XCode Command-Line tools for OSX. There are a few different ways to install these tools (including visiting the Apple developer site at developer.apple.com), but the simplest way to do it is via the command-line, following this tutorial:

<http://railsapps.github.io/xcode-command-line-tools.html>

As noted above, **make sure you have upgraded to Mac OSX 10.9 (Mavericks) or 10.10 (Yosemite) before taking these steps**, otherwise you may run into problems. If you have any issues following this tutorial, please let us know by posting your issue in [Piazza](#).

Installing Homebrew

Homebrew is a tool for very easily installing and updating developer software. It makes a lot of complex processes very simple.

1. To install Homebrew, run this command:

```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

1. Verify that you have installed it by checking the command:

```
$ brew -v
```

You should see “Homebrew 0.9.5”.

Installing git

1. Make sure that you've already [signed up for a Github account](#)⁹.
2. Install git with brew:

```
$ brew install git
```

3. Set up git by typing in the following commands. Be sure to replace YOUR_GITHUB_USERNAME and YOUR_EMAIL_ADDRESS with the correct information for your account!

```
$ git config --global user.name "YOUR_GITHUB_USERNAME"  
$ git config --global user.email "YOUR_EMAIL_ADDRESS"  
$ git config --global core.editor subl  
$ git config --global color.ui true  
$ git config --global push.default simple
```

⁹ <https://github.com/join>

Installing Ruby and Rails with rvm

RVM is a tool that easily installs and manages Ruby without making changes to the rest of your computer.

1. Run this command in your Terminal. (The \ at the beginning of the command is required to type!)

```
$ \curl -sSL https://get.rvm.io | bash -s stable --ruby=2.2.1
```

2. Close your Terminal with Cmd+Q and then reopen Terminal.
3. Run this command to enable ruby.

```
$ rvm --default use 2.2.1
```

4. Finally, install Ruby on Rails! This will take a while to install.

```
$ gem install rails -v 4.2.1
```

Installing Sublime Text

1. Download Sublime Text 3 and install it <http://sublimetext.com/3>

The screenshot shows the 'Download' section of the Sublime Text 3 website. A red circle highlights the text 'Sublime Text 3 is currently in beta. The latest build is 3083.' An arrow points to the 'OS X (10.7 or later is required)' link, which is underlined. Below it is a bulleted list of available platforms: Windows (portable version), Windows 64 bit (portable version), Ubuntu 64 bit (tarball for other Linux distributions), and Ubuntu 32 bit (tarball for other Linux distributions).

- [Windows](#) - also available as a [portable version](#)
- [Windows 64 bit](#) - also available as a [portable version](#)
- [Ubuntu 64 bit](#) - also available as a [tarball](#) for other Linux distributions.
- [Ubuntu 32 bit](#) - also available as a [tarball](#) for other Linux distributions.

2. After you've installed Sublime, enable the command line tool by running this command in your Terminal: (**Type this all on one line**)

```
$ ln -s /Applications/Sublime\ Text.app/Contents/SharedSupport/bin/subl  
/usr/local/bin/subl
```

Windows Installation

Although it is typically very challenging to install the development tools on a Windows computer, many wonderful people on the Internet have created a few simple tools to make it easier.

Please note: We strongly recommend that you use a Mac (or Linux) for development instead of Windows. Because Windows is not designed for web development, everything you do will be much harder and more likely to have errors. If you do not have a Mac, consider installing and using Ubuntu Linux!

Download and install the latest version of these tools by following along with the instructions at the links:

1. **Gow** <https://github.com/bmatzelle/gow/releases> (download Gow-0.8.0.exe)

The screenshot shows the GitHub release page for Gow 0.8.0. At the top left, there's a green button labeled "Latest release". Below it, the version "v0.8.0" and a commit hash "7e268e3" are listed. The main title is "Gow 0.8.0" in blue. Below the title, it says "bmatzelle released this on Feb 15, 2014". A paragraph of text follows, mentioning the release makes Gow easier to update and release, with time for more work and contributions. A bulleted list of changes is provided, starting with "Added ssh.bat alias for plink". At the bottom, there's a "Downloads" section with a table of files. The first file, "Gow-0.8.0.exe", has a red arrow pointing to its link. The table also includes "gow-utilities-src-0.8.0.tar", "Source code (zip)", and "Source code (tar.gz)".

File	Size
Gow-0.8.0.exe	8.68 MB
gow-utilities-src-0.8.0.tar	46.6 MB
Source code (zip)	
Source code (tar.gz)	

2. **Rails Installer 2.1** - <http://railsinstaller.org/en> (make sure you install 2.1)

Windows Downloads

 **WINDOWS**
RUBY 2.0

 **WINDOWS**
RUBY 2.1



Packages included are

Ruby 2.0.0

Rails 4.1

Bundler

Git

Sqlite

TinyTDS

SQL Server Support

DevKit

Packages included are

Ruby 2.1.5

Rails 4.1

Bundler

Git

Sqlite

TinyTDS

SQL Server Support

DevKit

3. **Sublime Text 3** - <http://www.sublimetext.com/3> (make sure you don't install 2)

Download

Sublime Text 3 is currently in beta. The latest build is 3083.

- OS X (10.7 or later is required)

 Windows - also available as a portable version

 Windows 64 bit - also available as a portable version

- Ubuntu 64 bit - also available as a tarball for other Linux distributions.
- Ubuntu 32 bit - also available as a tarball for other Linux distributions.

Verify Your Installation

These verification instructions will work on any platform.

Run these commands in the command line. You can run the commands from any directory. You should see a version number like the one on the next line after "`=>`". In this example, the `example_command -v` is what you should type, and `=> expected version number` is the result that you should see after you type out the command:

```
$ example_command -v  
=> expected version number
```

Run these commands, and verify that you have the correct version numbers:

```
$ rvm -v  
=> rvm 1.26.10 (anything 1.26 or above is good)  
  
$ ruby -v  
=> ruby 2.2.0p0 (anything 2.2 or above is good)  
  
$ bundle -v  
=> 1.7.9 (anything above 1.7 is good)  
  
$ git --version  
=> git version 2.3.0 (anything above 2.3 is good)  
  
$ rails -v  
=> Rails 4.2.1 (*must* be Rails 4.2 or 4.2.1)
```

	<p>How do version numbers work?</p> <p>Software uses a version numbering system so that developers can understand what has changed in software without having to follow it closely. For example, if we have software versioned 2.10.8:</p> <ul style="list-style-type: none">2. "Major" - changed for very large, often incompatible releases10. "Minor" - large changes, like new features8. "Maintenance" - smaller changes like bug fixes or security updates <p><i>Want to know tons more? Check out http://semver.org/.</i></p>
---	---

**YOU HAVE
MADE IT!**

See you soon!