

Prologue: Prerequisites

version 1.0

Ben Welsh

May 13, 2017

Contents

Prologue: Prerequisites	1
Command-line interface	1
Windows	1
Mac OSX	1
Python	1
Windows	2
Mac OSX	2
pip	2
Windows	2
Mac OSX	2
virtualenv	2
Windows	3
Mac OSX	3
Code compiler	3
Windows	3
Mac OSX	3
Git	3
Windows	4
Mac OSX	4

Prologue: Prerequisites

Your computer needs the following tools installed and working to participate. Visit each page to verify you are prepared. Installation instructions are provided.

Command-line interface

A [command-line interface](#) to interact with your computer is required.

Whether you know about it or not, there should be a way to open a window and directly issue commands to your operating system. Different operating systems give this tool slightly different names, but they all have some form of it.

Windows

On Windows the command-line interface is called the "command prompt." It is a perfectly fine piece of software, but to ease your experience with Python you should install [cygwin](#), a free alternative. Here's how:

1. Download the installer from [cygwin.com](#).
2. Run the installer.
3. Verify it's working.

Warning

When the class begins, all the examples will assume you are using cygwin instead of the standard command prompt. Please take the time to install it now. You'll thank me later.

Mac OSX

On Apple's Mac OSX operating system, a suitable command-line program is already installed. It's called "Terminal." You should try opening it up and typing in some basic commands to verify it's working. Here's how:

1. Click on the magnifying glass in the upper right hand corner.
2. Type in "Terminal".
3. Press enter or select the first search result.
4. A command line prompt should appear.
5. Verify it works by typing a simple command like `whoami` and pressing enter.

Python

[Python](#) is a free and open-source computer programming language. It's one of the most popular in the world and praised by its supporters as clear and easy to read.

That makes it ideal for beginners and is partly why it's been adopted by professionals in many fields, ranging from engineering and web development to journalism and music.

You can check if Python is already installed on your computer by visiting your command line and entering the following:

```
python --version
```

You should see something like this after you hit enter:

```
Python 2.7.13
```

If not, you'll need to install Python on your system.

Windows

Here's how to do so on Windows:

1. Verify Python is not already installed with the command above.
2. Download the Python 2.7 installer from python.org <<http://python.org>>.
3. Run the installer, and make sure to check the "Add python.exe to Path" setting is checked.
4. Verify that Python has been installed.

Mac OSX

If you're using Mac OSX, it's more likely that Python is already installed. Here's how to find out what version you have.

You'll start going to your terminal.

1. Go to your Terminal.
2. Type the `python --version` command described at the top of the page.

pip

The [pip package manager](#) makes it easy to install open-source libraries that expand what you're able to do with Python. Later, we will use it to install everything needed to conduct a data analysis.

Verify pip is installed by typing the following into your command-line interface:

```
pip --version
```

Windows

If you've followed the previous installation instructions for Python, pip should be already installed and ready to go. You can test it out by opening up the command prompt or cygwin and typing in the command above.

If you don't have pip, carefully review the instructional video for Python and make sure you've followed all of the steps. If that doesn't work, try following the official instructions [from pip's creators](#).

Mac OSX

In some cases, pip will be already be installed on Mac OSX computers. But to make sure you should take the following steps:

1. Check if pip is installed.
2. If not, install it using `easy_install`.
3. Verify pip is now available.

virtualenv

The [virtualenv environment manager](#) makes it possible to create an isolated corner of your computer where all the Python tools you use to build an application are sealed off.

It might not be obvious why you need this, but it quickly becomes important when you need to juggle different tools for different projects on one computer.

By developing your applications inside separate virtualenv environments, you can use different versions of the same third-party Python libraries without a conflict.

You can also more easily recreate your project on another machine, handy when you want to copy your code to a server that publishes pages on the Internet.

Windows

Here's a walkthrough of how to do it on Windows:

1. Verify pip and python are already installed as we did in the previous installers.
2. Verify if virtualenv is not already installed.

```
virtualenv --version
```

3. Install virtualenv with pip.

```
pip install virtualenv
```

4. Verify that virtualenv has been installed.

```
virtualenv --version
```

Mac OSX

Here's how to do it on Mac OSX:

1. Verify pip and python are already installed as we did in the previous installers.
2. Verify if virtualenv is not already installed.

```
virtualenv --version
```

3. Install virtualenv with pip. We'll need to use `sudo` to install it in the system folders.

```
sudo pip install virtualenv
```

4. Verify that virtualenv has been installed.

```
virtualenv --version
```

Code compiler

A [code compiler](#) is a tool that lets your computer installed more advanced software. It is required to take advantage of the heavy-duty data analysis tools we will be using like the [pandas](#) library for Python.

Windows

Windows users will need to download and install [the Microsoft Visual C++ Compiler for Python 2.7](#). Here's a walkthrough:

Mac OSX

If you are using Mac OSX, you will need to install [Xcode](#). It's a developer toolkit that includes a compiler. Here's how: You can make sure you've got it by running this on your command prompt.

```
xcode-select --install
```

Git

[Git](#) is a version control program for saving the changes you make to files over time. It is useful when you're working on your own, but quickly becomes essential with large software projects, especially when you work with others.

For this class you will need to have git installed and working from your command-line interface.

You can verify it's working from your terminal by typing in the follow code and hitting the enter key:

```
git --version
```

If git is installed and working, you should see something like this:

```
git version 2.11.0
```

If you don't have it installed, you'll need to follow the instructions below.

Windows

Here's how to install git on Windows.

1. Verify the git is not already installed on the command prompt.
2. Download the installer from git-for-windows.github.io.
3. Run the installer, accepting all of the default options.
4. Return to the command prompt and verify git is now installed.
5. Configure git with your identity with these two commands:

```
git config --global user.email "your@email.com"
git config --global user.name "your name"
```

Mac OSX

If you've followed the previous installation instructions for a code compiler, git should be already installed and ready to go. You can test it out by opening up the command prompt or cygwin and typing in the command above. Here it is again:

```
git --version
```

If it's not, visit git-scm.com and download the installer. Run it.

Either way, configure git with your identity using these two commands:

```
git config --global user.email "your@email.com"
git config --global user.name "your name"
```

Warning

You will likely need administrator privileges on your computer. Be sure you have them before you begin.