

Lab 01 - Setup a development environment

The first step is to build a development environment. Most software installations are done on remote servers. These servers do not have a GUI at all and all of this work is done at the command line. Some systems, like IoT systems, are too small to use a GUI - again all the work is done at the command line.

The command line has advantages. It can be scripted so that you can re-produce your results. It is also easier to document.

This is going to be broken up based on the operating system you are working with.

Windows

On Windows the command line is the Power Shell. There are 2 different security levels that the power shell can run under. The first is as your login user. The 2nd is as an administrative user.

There is an older command line, command.com, that can also be used for this.

It is important that you configure your system for development.

1. Setup so that PowerShell can run .ps scripts.
2. Configure your desktop explore shows all files and shows file extensions. The file extension is the last few characters after the last dot (.) in a file name. In windows 10 there is a configuration superficially for development that will set this and a few other settings to make your life easier.
3. For a lot of stuff we will use the Linux/Unix 'bash' shell - this will be installed as a part of our 'git' install a little bit later. 'bash' is the same shell that is used on many Mac and most Linux systems.

TODO - navigate to this.

TODO - screen capture for this.

Package Manager

Windows is lacking a package manager. We need to install chocolate as the package manager.

Install chocolatey for Windows 10, this is a package manager. You will need to determine if your system is 32 bit or 64 bit. This will be true for most of the Windows Installs. <https://chocolatey.org/install>

Install Google Chrome

A number of tools depend on the Google Chrome portability library.

Install Chrome (if you have not already done it) https://www.google.com/chrome/?brand=CHBD&gclid=CjwKCAjwyo36BRAXEiwA24CwGSgDDdrI4XOUKv4CPwFQfs7M2HaXiRJ-MMeszA20rC72r-9U13-8jBoCQV4QAvD_BwE&gclsrc=aw.ds

Source code control

We will be using ‘<https://github.com>’ for class handouts and for turning working with files through the semester. The underlying system that github uses is ‘git’. You will need to to ‘<https://github.com>’ and create a free account on the site.

Bring up this page in the “chrome” browser with xyzzzy

Determine what version of windows you have, the 32 bit or 64 bit version. <https://support.microsoft.com/en-us/windows/which-version-of-windows-operating-system-am-i-running-628bec99-476a-2c13-5296-9dd081cdd808> has an explanation.

You will also need the windows ‘git’ tools installed. <https://git-scm.com/download/win> in chrome will start the download as soon as you go to the page. Run the installer. You should end up with a MinGW Bash shell icon on your desktop.

git for windows (Includes MinGW and bash) <https://git-scm.com/download/win>

5. Install VS Code <https://code.visualstudio.com/download>
6. Install Anaconda Python <https://www.anaconda.com/products/individual>
7. Install “vim”, <https://www.vim.org/download.php#pc> A good guide to vim on windows <https://www.freecodecamp.org/news/vim-windows-install-powershell/>

Install Packages in Python

The installation manager for Python is “pip”.

On Mac/Linux:

```
$ conda install -c conda-forge tensorflow
```

On Windows:

```
C:\> conda install -c conda-forge tensorflow
```

Installation on a Mac

1. Install XCode (Apple Store) On your Apple Mac bring up the Apple Store. Search for “XCode” - it is free. Install.
2. Install brew / git. Search for “mac brew”. Cut and paste the line. Bring up a “terminal” - In the finder brows to your /Applications, then in the Utilities you will find a terminal. Paste the “brew” install line into that. Run. Now at the command line do `$ brew install git`. <https://brew.sh/>
3. Install iTerm2 - Search for “iterm2” and follow the instructions to install a better terminal. Use that instead of the Mac default Terminal application. <https://www.iterm2.com/>
4. Install Chrome (if you have not already done it) (Search for “Download Chrome” - follow googles instructions) https://www.google.com/chrome/?brand=CHBD&gclid=CjwKCAjwyo36BRAXEiwA24CwGSgDDdrI4XOUKv4CPwFQfs7M2HaXiRJ-MMeszA20rC72r-9U13-8jBoCQV4QAvD_BwE&gclsrc=aw.ds
5. Install VS Code. Search for “Visual Studio Code” Install. The add the “Python Package to it”. I also installed the “Python Lint” package. <https://code.visualstudio.com/download>
6. Install Anaconda Python. Search for “Mac Install Anaconda Python” - install the anaconda package (Takes a while). <https://www.anaconda.com/products/individual>
7. Install “vim” <https://github.com/macvim-dev/macvim/releases/tag/snapshot-171>
8. Install “iTerm 2.x”

Linux Installs.

This depends on the kind of Linux Ubuntu, RedHat, CentOS, Arch etc, that you have. Let’s get together and figure out hat detail and work on it one-on-one.

Configure and Demo of Using Debugger

1. Configure VS Code (common) (Note on Windows the path (if you have to enter it) is `C:\anaconda3\python.exe` Usually VS Code will give you a drop down menu to pick from.
2. Use VS Code debugger (common)

Simple Hello world Program

Edit a file called `hello-world.py` and put the following in it:

```
import sys
print ( "hello world" )
print ( sys.version )
```

At the command line:

```
$ python hello-world.py
```

Modify the file to be:

```
import sys
import tensorflow as tf
import pandas
import numpy
import bottle
import sqlite3
import tensorflow
import keras
import matplotlib
from bs4 import BeautifulSoup
```

```
print ( "hello world" )
print ( sys.version )
```

Run it again.

Use a '#' pound-sign to create a line with your name on it.

```
# Author: Jagadish Bapanapally
# Author: Philip Schlump
```

```
import sys
import tensorflow as tf
import pandas
import numpy
import bottle
import sqlite3
import tensorflow
import keras
import matplotlib
from bs4 import BeautifulSoup
```

```
print ( "hello world" )
print ( sys.version )
```

Run it again.

If it works then turn this in as a part of your lab.

Lab Questions

Use the editor and write up an answer to:

1. What is your name?
2. What part of software installation causes the most frustration?
3. Did you remember to put your name in your code as a comment?

Save the file and upload this as a part of your lab work.