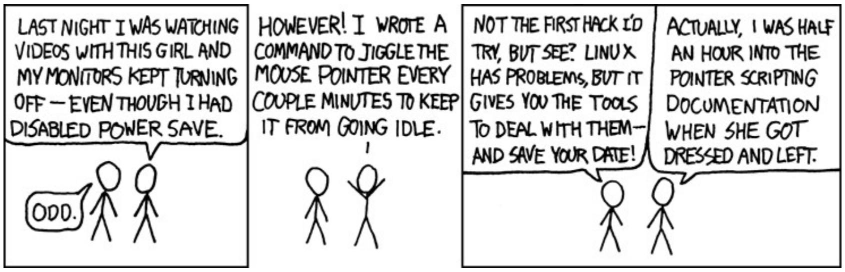
UNIVERSITY OF MICHIGAN

Michigan Data Science Team

TUTORIAL 1

Outline

1. About the presenter
2. Setup (should have this done already!)
3. Terminal
4. Git
5. Python

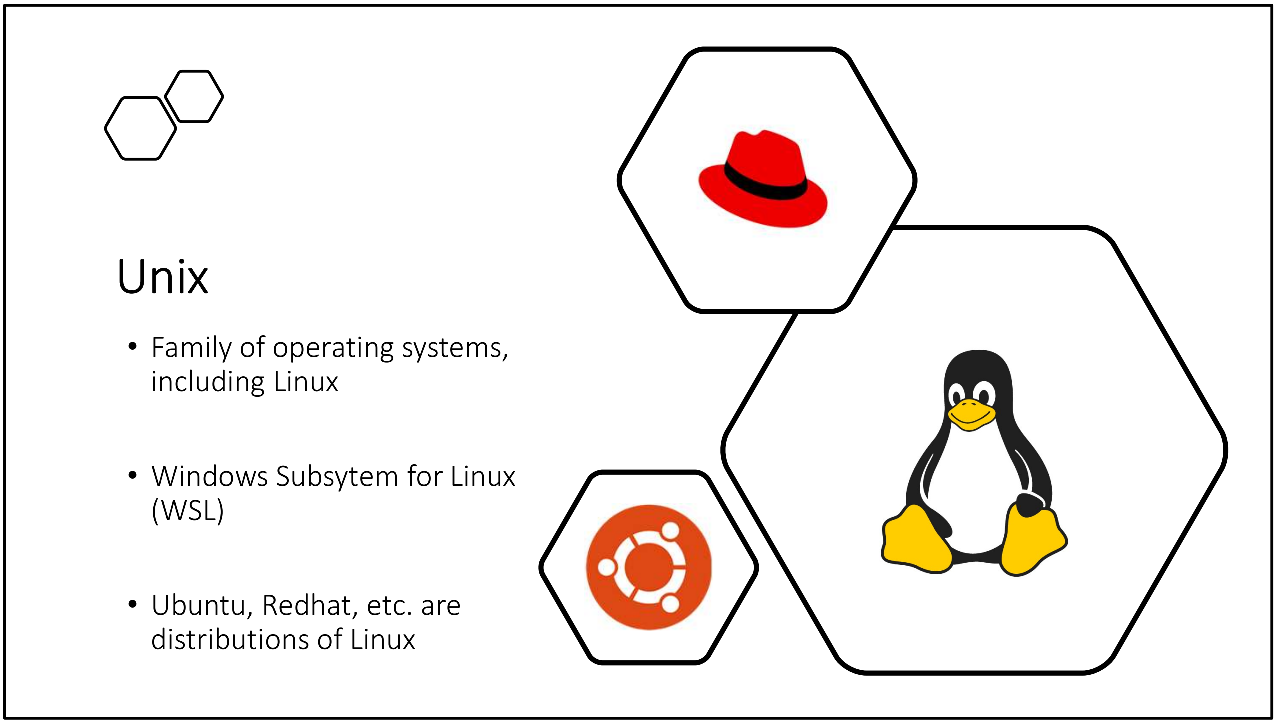

LAST NIGHT I WAS WATCHING VIDEOS WITH THIS GIRL AND MY MONITORS KEPT TURNING OFF — EVEN THOUGH I HAD DISABLED POWER SAVE.
ODD.
HOWEVER! I WROTE A COMMAND TO JIGGLE THE MOUSE POINTER EVERY COUPLE MINUTES TO KEEP IT FROM GOING IDLE.
NOT THE FIRST HACK I'D TRY, BUT SEE? LINUX HAS PROBLEMS, BUT IT GIVES YOU THE TOOLS TO DEAL WITH THEM— AND SAVE YOUR DATE!
ACTUALLY, I WAS HALF AN HOUR INTO THE POINTER SCRIPTING DOCUMENTATION WHEN SHE GOT DRESSED AND LEFT.

Presented on:

UNIVERSITY OF
MICHIGAN

Setup

What were all those things we made you install?!

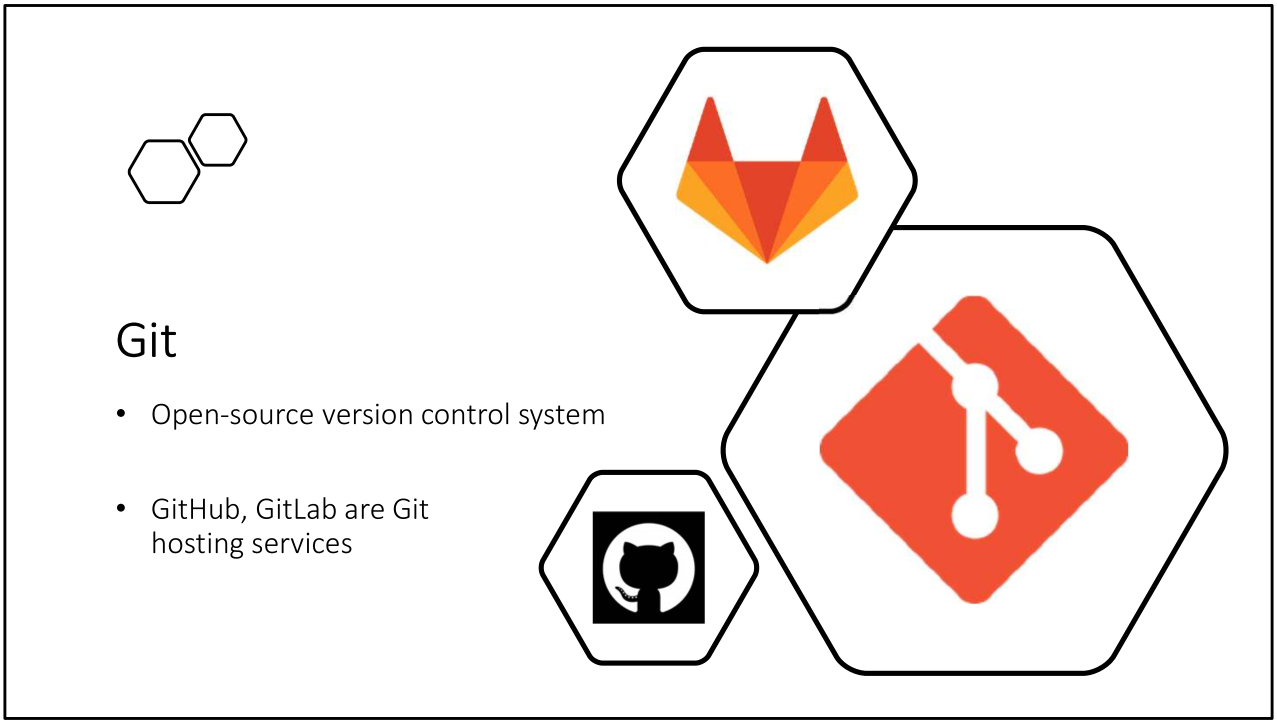


Unix

- Family of operating systems, including Linux
- Windows Subsytem for Linux (WSL)
- Ubuntu, Redhat, etc. are distributions of Linux

Unix and Unix-like operating systems are a family of OS’s, just like Windows or MacOS. Linux is the most popular variant of Unix, and its selling points include its customizability, easy package management, and the fact that it is free and maintained by the open-source community. Some of these features are also shared by macOS, particularly with the help of the package manager Homebrew. Windows, however, tends to be not as friendly; WSL allows users to run Linux within Windows without needing a virtual machine or dual-booting.

Ubuntu and Redhat (used by CAEN) are among the most common distributions of Linux. Distros primarily differ in their appearances, package managers, and customizations.

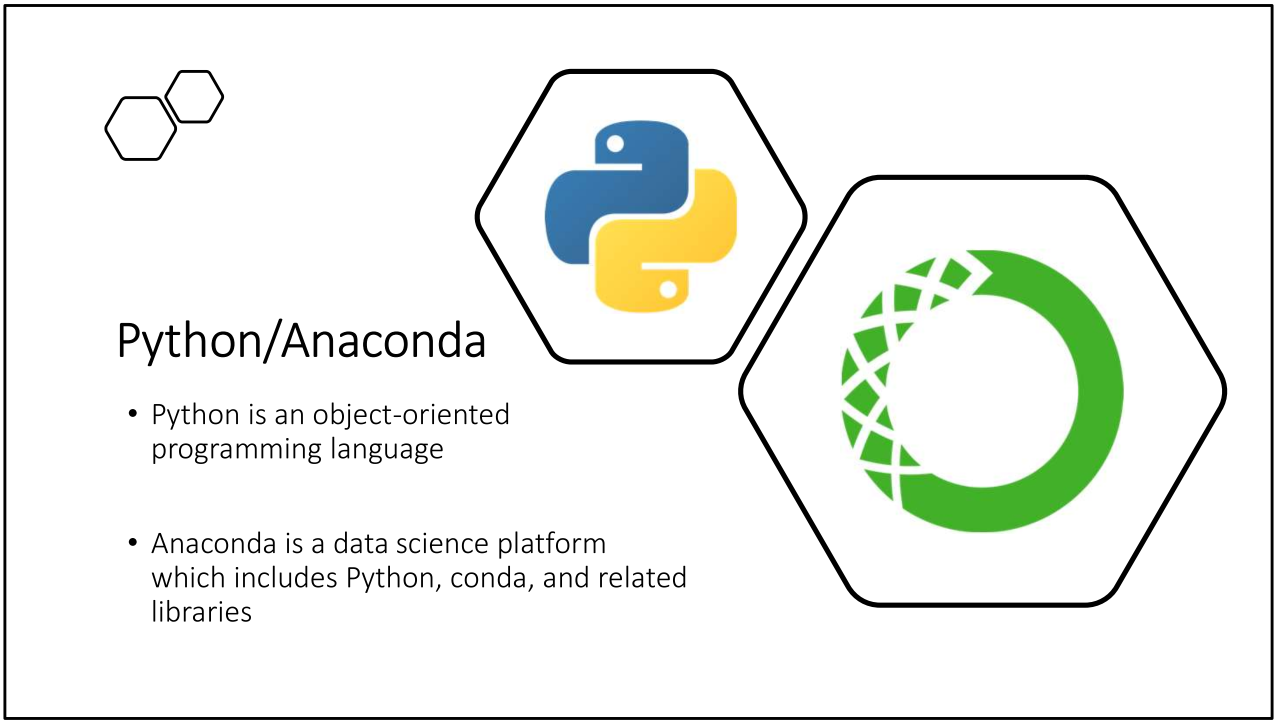


Git

- Open-source version control system
- GitHub, GitLab are Git hosting services

Git is the most widely used version control system. As the name suggests, version control is used to manage different versions of code as they are being developed. Its design is suited for maintaining code versions over time and allows for programmers to collaborate on the same code.

GitHub and GitLab are software development platforms which support git and other features, allowing you to store and manage code, teams, etc. online. Since they are both git services, there are few differences between the two in day-to-day use.



Python/Anaconda

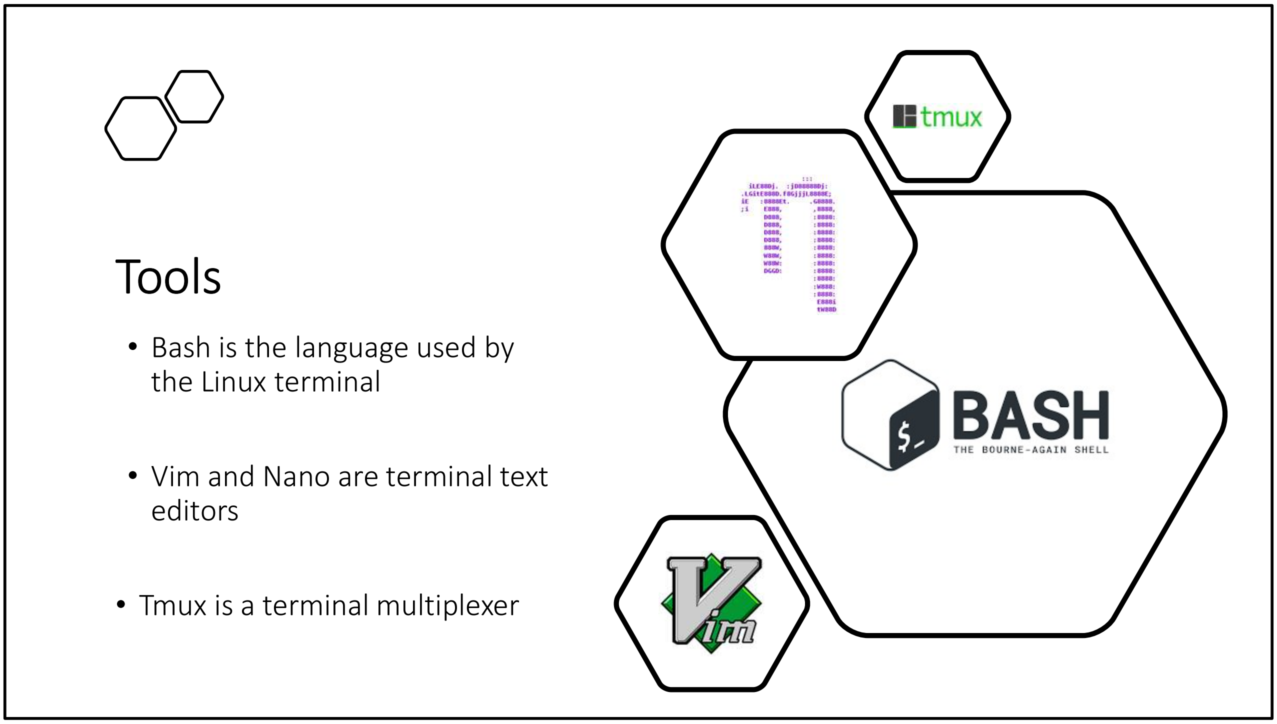
- Python is an object-oriented programming language
- Anaconda is a data science platform which includes Python, conda, and related libraries

Python is one of the most popular languages used in data science applications (closely matched with R). It’s easy to learn, scalable, and provides many useful libraries. Of course, Python can be installed and used on its own; in our setup, however, we asked you to instead install anaconda.

Anaconda is a data science platform. Its install includes Python and conda, a package manager. Like mentioned before, the Python community has created a ton of useful libraries to implement code without having to write it yourself. Conda lets you easily install these packages, and Anaconda will pre-install many of them for you. Conda also helps manage virtual environments, which allows you to install multiple package versions in separate spaces for separate projects.

Terminal

There are two major products of Berkeley, CA -- LSD and UNIX. We don't believe this to be strictly by coincidence.



Tools

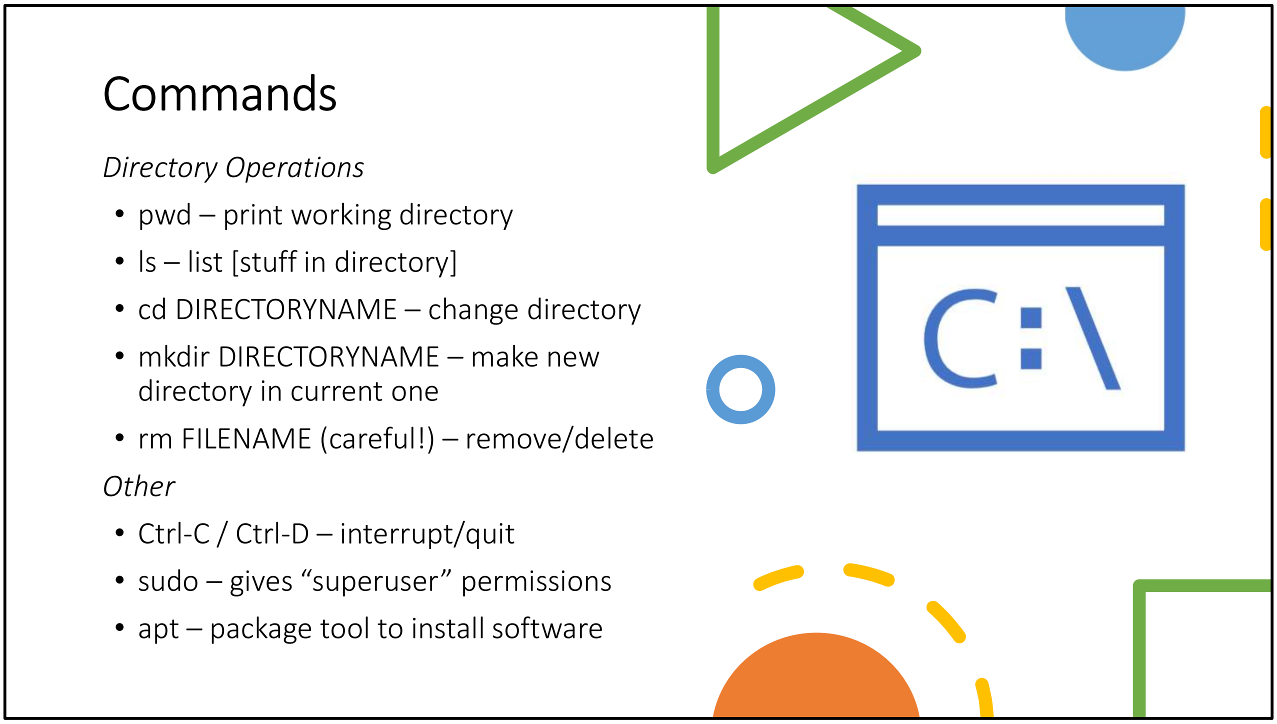
- Bash is the language used by the Linux terminal
- Vim and Nano are terminal text editors
- Tmux is a terminal multiplexer

tmux
BASH
THE BOURNE-AGAIN SHELL
$_
Vim

Bash is the shell, or command language interpreter, used by the Linux terminal. It can handle complex scripting tasks, but its basics are most useful for managing directories, managing packages, and running scripts (and these are the primary uses for MDST).

Vim and Nano are terminal text editors, useful for viewing and editing files quickly while working in the terminal. Though they both have a learning curve, learning to use one of them will make terminal work far more convenient.

It may also be useful to try tmux, particularly for WSL users. Tmux is a terminal multiplexer, which allows multiple terminal sessions to be accessed in a single window (similar to browser windows and tabs). It can also be used to detach from terminal sessions and let code and scripts run in the background.



Commands

Directory Operations

- pwd – print working directory
- ls – list [stuff in directory]
- cd DIRECTORYNAME – change directory
- mkdir DIRECTORYNAME – make new directory in current one
- rm FILENAME (careful!) – remove/delete

Other

- Ctrl-C / Ctrl-D – interrupt/quit
- sudo – gives “superuser” permissions
- apt – package tool to install software

** terminal demo the commands

These are some of the most commonly used commands for terminal use. Of course there are many more and an easy way to learn and remember them is to look up a cheat sheet.

Notes:

- ls has options to show hidden files, show files of certain extension types, etc. (can find these on a cheatsheet!)
- cd can take the names of subdirectories, ‘..’ for going up a directory, ‘~’ for home directory
- Be careful using rm! Unlike regular file explorer delete which sends files to the recycling bin, rm is *very* hard to undo. May want to use some other command to send to trash. Solutions for this can also be found online (gio trash FILENAME in Ubuntu).

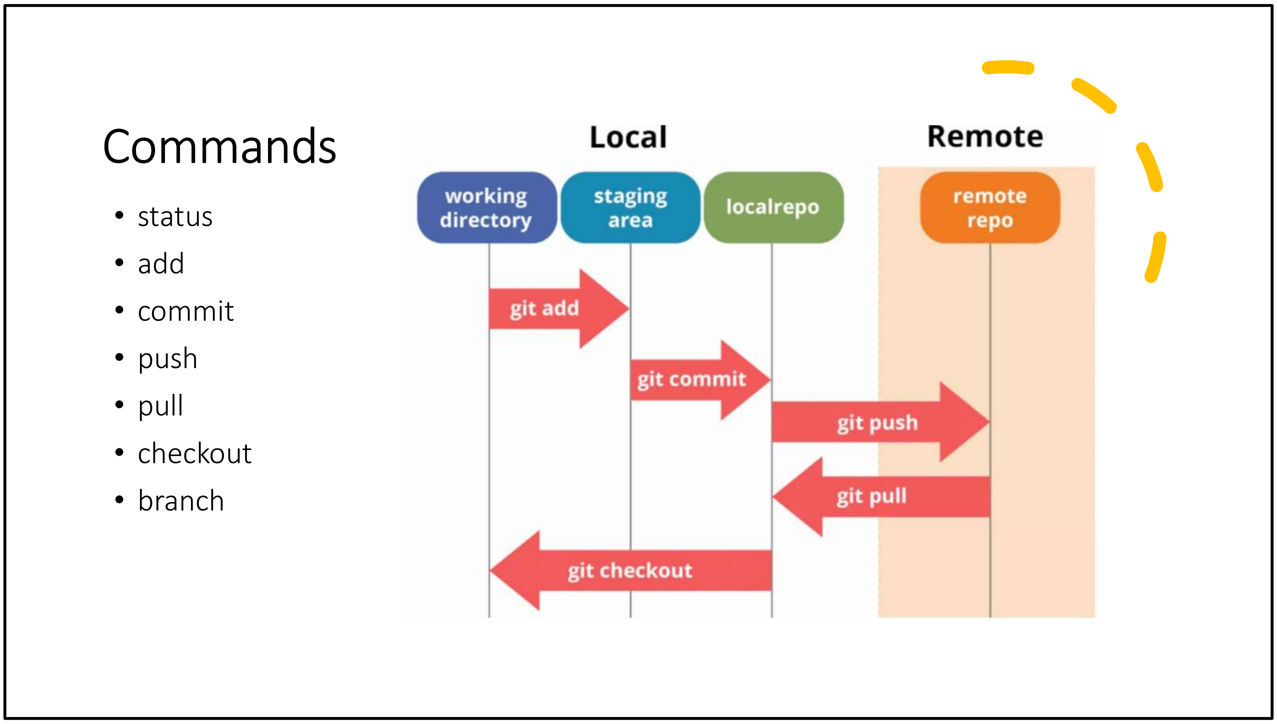
Git

In case of fire: git commit, git push, leave the building

Workflow

- Content (files, folders, etc.) is stored in repositories
  - Local is on your computer
  - Remote is online (github.com)
- Branches are versions of a repository which diverge from the main version (master)

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



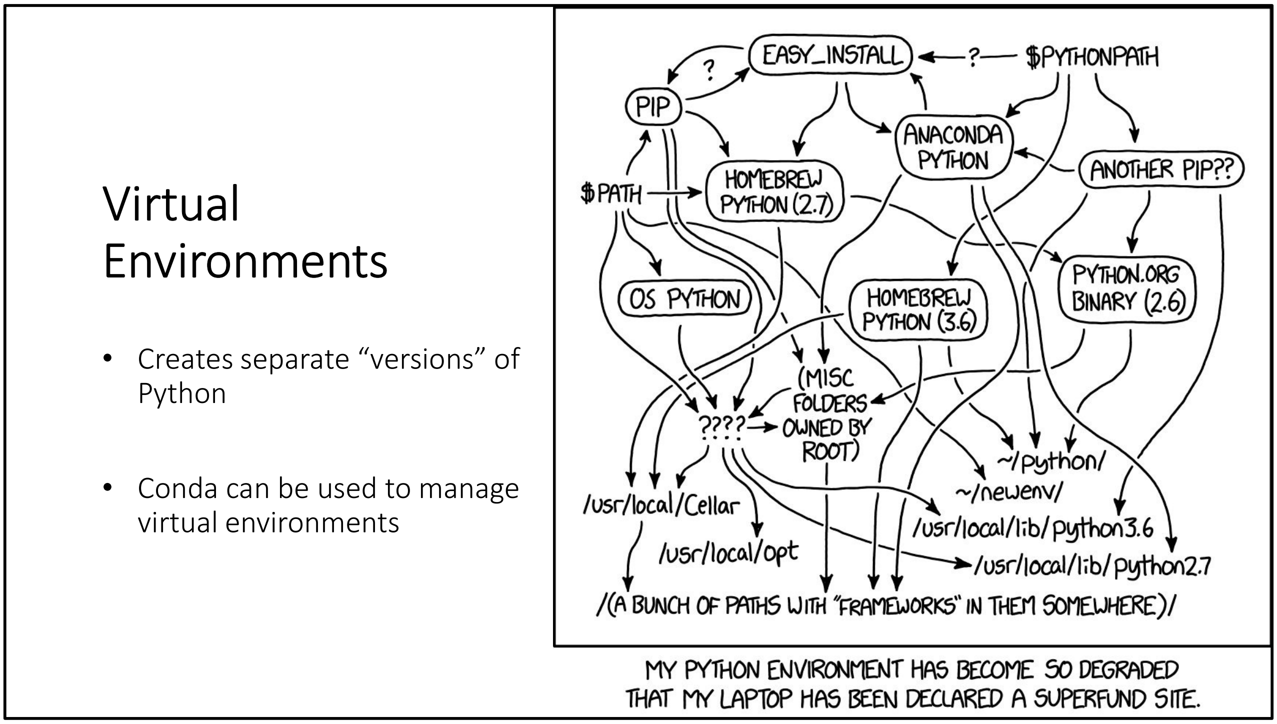
Commands

- status
- add
- commit
- push
- pull
- checkout
- branch

** terminal demo the commands

Python

Why should you take a data scientist with you into the jungle?
They can take care of Python problems



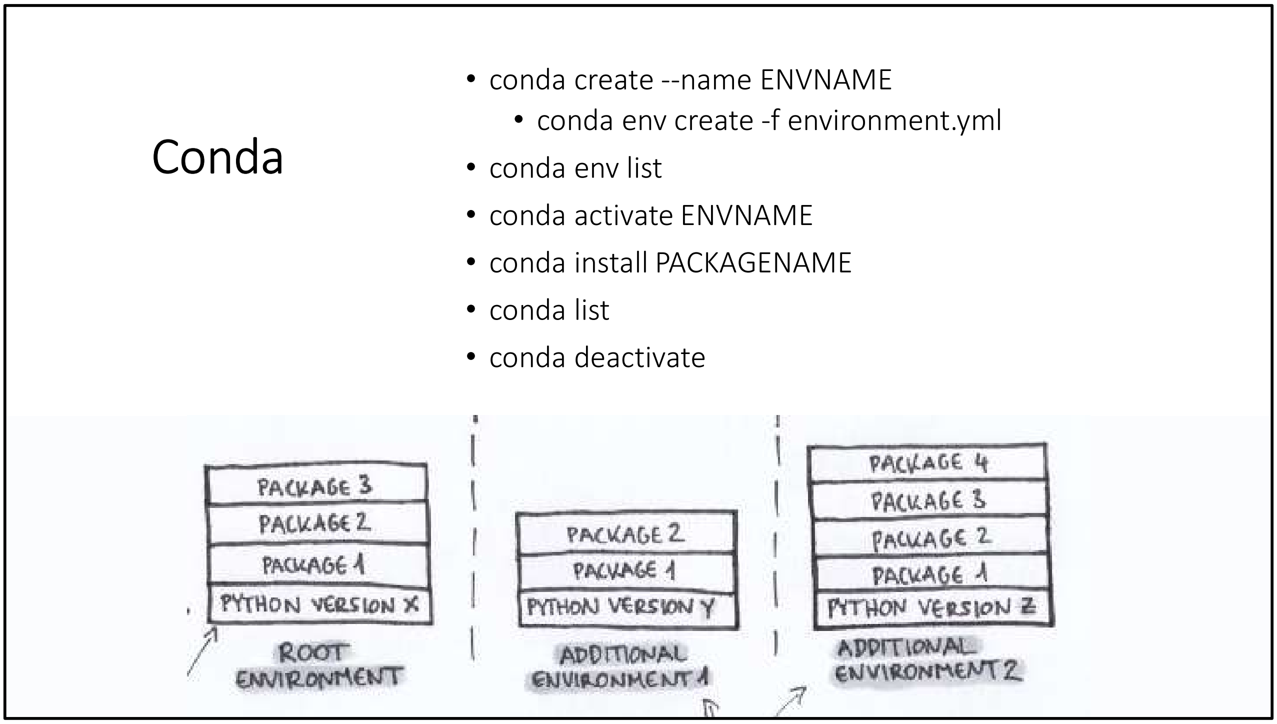
Virtual Environments

- Creates separate “versions” of Python
- Conda can be used to manage virtual environments

MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

A note about virtual environments – use them! Sometimes projects will require different libraries, especially different versions of libraries, and having isolated environments makes it easy to accommodate these differences. Plus, it makes it easy to fix problems caused by broken libraries or dependencies.

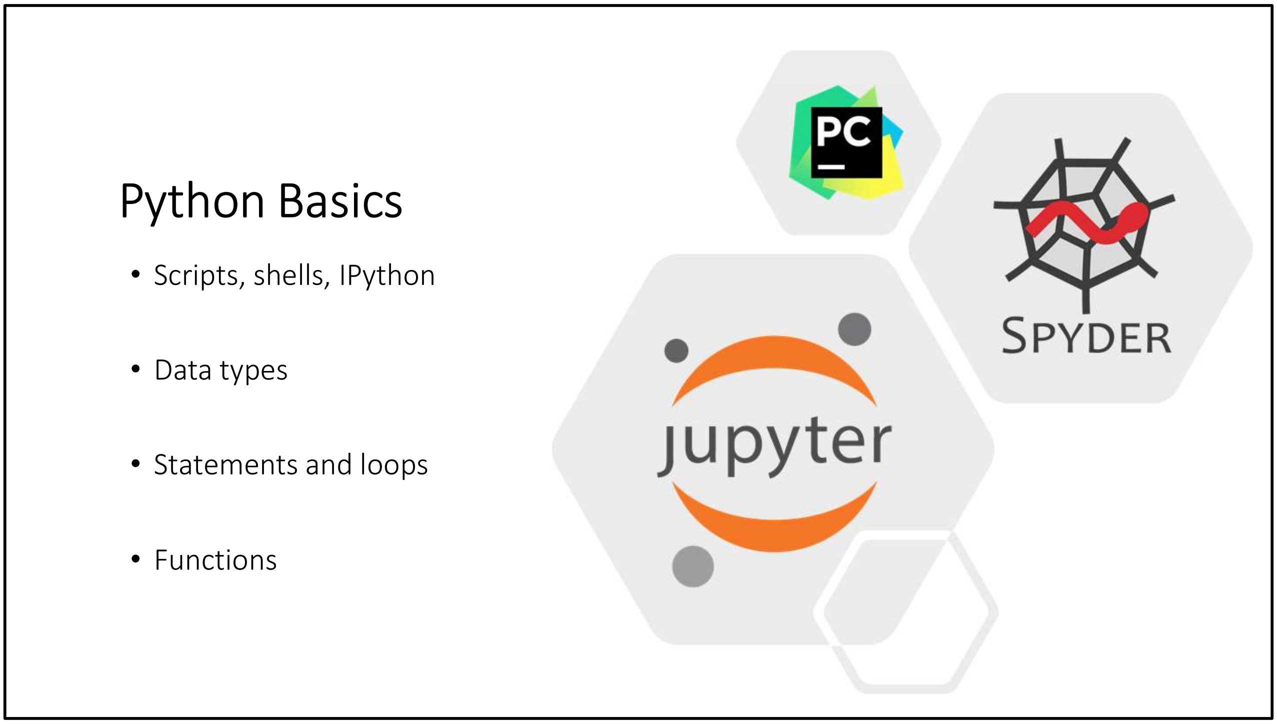
There are many python tools that manage virtual environments, including conda. Conda is both a package manager and can create virtual environments, so after using conda to create environments, it can also be used to install packages into those environments. It does not, however, support all python libraries, so there may be cases where pip is required to install those libraries. In those cases there are workarounds which can be used with caution.



Conda

- conda create --name ENVNAME
  - conda env create -f environment.yml
- conda env list
- conda activate ENVNAME
- conda install PACKAGENAME
- conda list
- conda deactivate

** demo in terminal



Python Basics

- Scripts, shells, IPython
- Data types
- Statements and loops
- Functions

PC
SPYDER
jupyter

Python can be run as a script (from a file) directly from the terminal. **demo create and run simple script (hello world)

Also can run as a shell. **demo commands in shell (maybe some math)

IPython stands for interactive python, which is pretty much a pretty version of the shell. Jupyter was a spin-off from Ipython and is an interactive, notebook-style front end for it. **Jupyter demo