



Week 1

Guan He

guanh@uchicago.edu

Overview



Installation & Setup

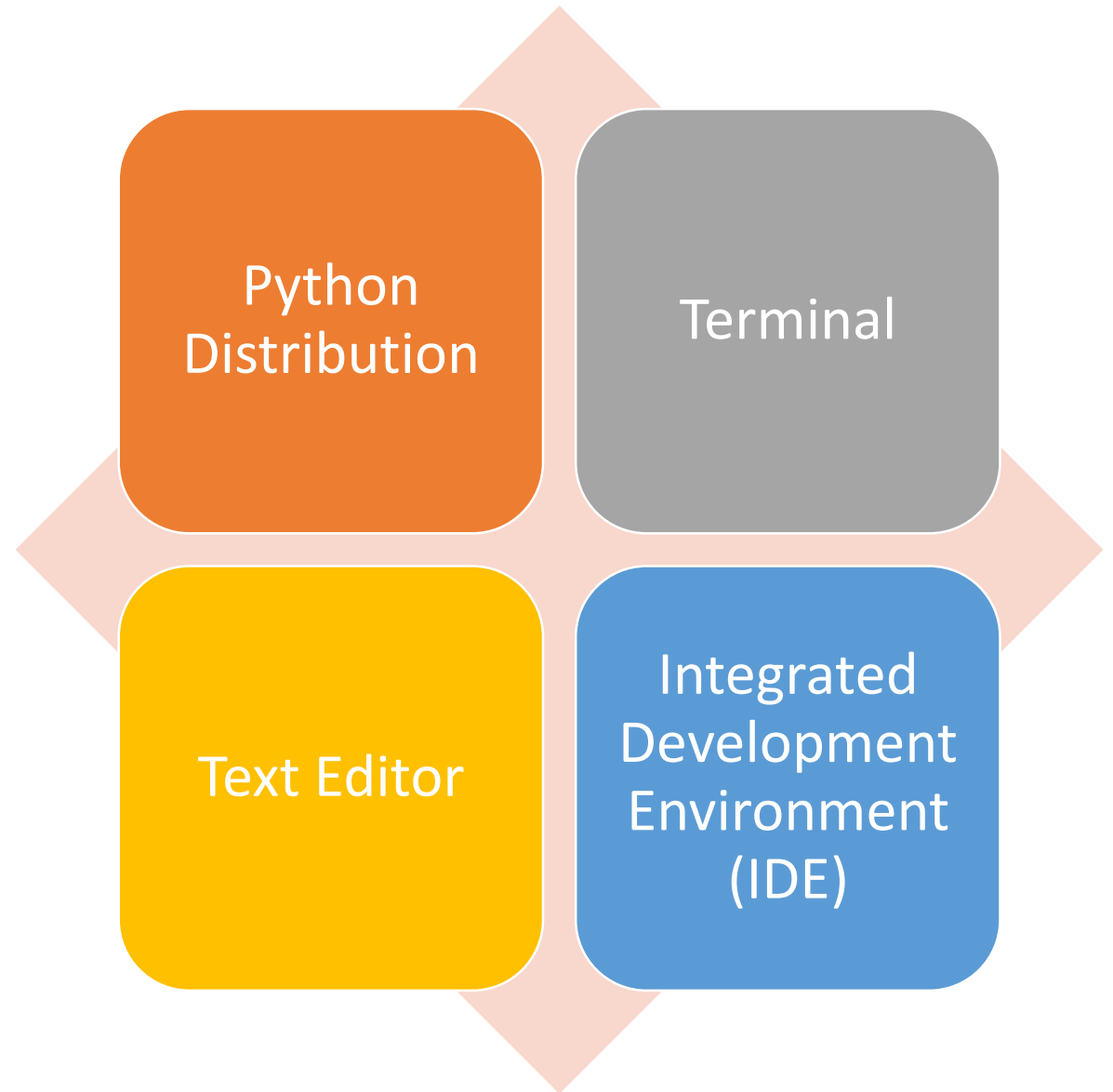


Basic Terminal Commands



Git & Version Control

Installation & Setup



Python Distribution

Anything defined by python.org is counted as “Python” itself. As of March, 2019, python version is 3.7.3

A Python Implementation is an actual program that provides that behavior, like CPython, which you can download on python.org

A Python Distribution is a bundle that contains an implementation of Python along with a bunch of libraries or tools, like what we use for this class – Anaconda Python Distribution

Anaconda

Used for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment

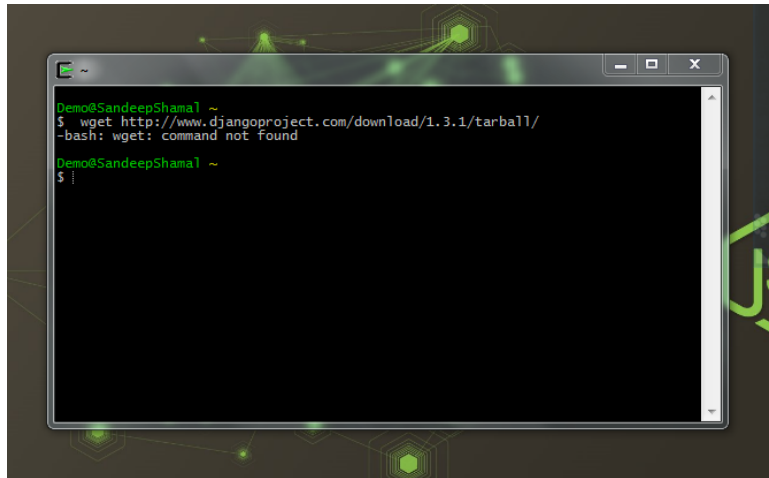
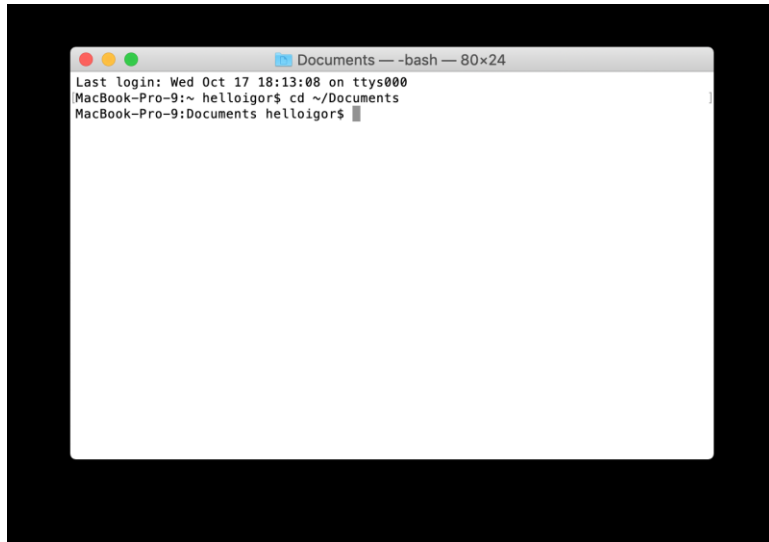
A set of tools: Jupyter Notebook, JupyterLab, Spyder etc. You can add more tools to your Anaconda by using the Python package manager [pip](#) or Anaconda package manager [conda](#)

Anaconda Navigator: a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands

Terminal

What is it? It is a command processor that typically runs in a text window where the user types commands that cause actions. It can also be called Bash, Shell, Prompt, Command Line etc.

Why use it? It will be easier for us to navigate the system, locate files, and run programs. It also has many more advanced functions.



Mac Terminal vs Cygwin

Try a few of the commands:

1. `pwd`: print working directory (path)
2. `cd`: change directory
3. `ls`: list all items under current directory
4. `mkdir`: make a directory (a new folder)
5. `touch`: create a new file
6. `rm`: remove a file
7. `rm -r`: remove a folder (recursively)

Text Editor



A type of computer program that edits plain text



It does not run a python program



There are many text editors like Microsoft Notepad, Atom(what we use for this class), Visual Studio Code(what I personally recommend), and Sublime etc.

How to Use a Text Editor

1. Open atom -> Drag a folder or file into atom -> Work on relevant files

2. Set the directory in terminal -> use the “atom .” or “atom filename” command (note there is a space in the commands) -> Work on relevant files

Text Editor

After you finishing working on a “.py” file and save it. You could simply run it in your terminal to test the script. Simply set the directory to the current directory you are working in, and use the “python yourfilename.py” command to execute the script.

Of course, there are many extensions that you can add to your text editor to handle the execution process automatically. You do not need to execute the commands in terminal by hand to see all the results.

Integrated Development Environment (IDE)

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools, and a debugger.

Examples: RStudio, Spyder, Jupyter, and PyCharm etc.

Different from the text editor, you can usually see the output of your code immediately in the ide. You may like these ide better than text editor.

Relevant Terminal Commands for This Course

`cd ~/Desktop (you can use Tab to autocomplete path)/path to directory`

`cd ..(go up one level)`

`cd ./ (go down one level)`

`mkdir foldername (make a new directory)`

`touch filename.xxx (make a new file of any type)`

`python filename.py (to execute the script)`

`jupyter notebook`

`jupyter lab`

`spyder`

`atom (filename.py/.)`

`code (filename.py/.)`

Git & Version Control

What is version control? A component of software configuration management, version control, also known as revision control or source control, is the management of changes to documents, computer programs, large web sites, and other collections of information. Git is version control system (VCS). GitHub is a web-based service for version control using Git.

Simply put, GitHub is like a Google Drive in the Cloud (where coders can see the changes each other makes and collaborate), though it has more advanced features. You can see and use all previous version of your code (GitHub tracks and stores all of them). In short, **Git is Version Control System and GitHub is a hosting service for Git Repositories**. There are also other Git services like GitLab, but we will use GitHub for this class.

How to Use GitHub to Submit Homework in Terminal?



Repository: A Git Repository, or a repo, is a folder that you've told Git to help you track file changes.



`git clone` (download the repo)



`git add filename.py` (add the files you want to upload or all the files
“.”)



`git commit -m ""` (A commit is a set of one or more changes to a file(or a set of files). Every time you save, it creates a unique ID(“hash”) which helps it keep track of the history. Track the changes and add a message)



`git push` (upload the files to GitHub)

More about Git

You can also use GitHub Desktop to do the same thing

.gitignore (a file that specifies which file you do not want to push to GitHub)

`git pull`

Push as frequently as you can, even if you just make a small change.

Make commit messages meaningful.

Any Questions? This is very important!

Link Git to Your GitHub Account

Setting your email address for *every* repository on your computer

- 1 Open Git Bash.
- 2 Set an email address in Git. You can use your [GitHub-provided no-reply email address](#) or any email address.

```
$ git config --global user.email "email@example.com"
```
- 3 Confirm that you have set the email address correctly in Git:

```
$ git config --global user.email  
email@example.com
```
- 4 Add the email address to your GitHub account by [setting your commit email address on GitHub](#), so that your commits are attributed to you and appear in your contributions graph.

How to Use GitHub Desktop?

There is a very detailed documentation on using GitHub Desktop:

<https://help.github.com/en/desktop/getting-started-with-github-desktop>

If you want to use the terminal commands to manage Git, I will send out detailed instructions on how to set up your Git and use it.

Procedures

1

Go to Your Class GitHub
Repo and Clone the
Repo

2

Do Homework in the
Repo and Add or Make
Changes to Relevant
Files

3

Push to the Remote
Repository in the Cloud

Some suggestions



Google the questions before you ask



Come to the class prepared



Do homework independently