Lesson 0

Getting Started

The Command Line

- Today our computers all have GUIs
  - **G**raphic **U**ser **I**nterfaces
  - Ability to control with mice or touch
- But we did not use to have that (pre mid-80s)
  - Intead, we used text to interact with computers
  - Type commands at prompt; computer reacts
- These tools still exist on your computer
  - Often hidden, and differ on each OS
  - In general, we call them the **command line**

The Command Line

- Today our computers all have GUIs
  - **G**raphic **U**ser **I**nterfaces
  - Ability to control with mice or touch
- But we did
- These tools still exist on your computer
  - Often hidden, and differ on each OS
  - In general, we call them the **command line**

Will be our primary tool in this course

MacOS, Linux: The Terminal

- Unix environments call it the **Terminal**
  - Present in MacOS and most Linux flavors
  - This is the tool we will use in all videos
- On Linux it should be easy to find (we hope!)
  - Typically appears after start-up
- But on MacOS it is a little bit hidden
  - Located in Applications/Utilities
  - Good idea to add to your dock this semester

Windows: The PowerShell

- Windows actually has two options
  - Can find them searching in the search bar
  - Command Shell (old), PowerShell (new)
  - We will use PowerShell for this class
- Mostly works the same as the Terminal
  - Basic commands are only a little different
  - The major differences are not used in this course
- Windows does have a Linux Terminal in beta
  - But this is not recommended for this course

Looking at Files

- When you start, you are inside of a folder
  - Just like you are in a Finder window
  - Start off with in the **home directory**
- To see the files, type the command `ls`
  - Stands for **l**i**s**t files
  - Will show both files and folders
  - **Unix only:** Get more details by adding `-s`
- Some files not visible (name starts with period)
  - **Unix only:** view these files by adding `-a`

The Active Directory

- The current folder is the active directory
  - You can view it with the command `pwd`
  - Will show folders separated by /s (or \ on Windows)

/Users/wmwhite

Parent folder

Current folder

- Parent level folder can be seen graphically
  - Very obvious in Windows Explorer/Finder
  - MacOS needs you to add Path to your toolbar

Changing the Folder

- Done with command: `cd <name>`
  - `<name>` must be folder inside of this one
  - So you are always "moving forward"
- To back out: `cd ..` (two periods)
  - Goes back to containing folder
- Can also quickly return to home directory
  - Just type `cd` by itself
  - Very common to do this accidentally
  - Always use `pwd` if you are confused

Absolute Paths

- Can use cd to jump directly to another folder
  - Important for navigating between Python files
- In that case <name> must be an absolute path
  - Path that lists all parents from the top
  - **MacOS, Linux:** Path must start with /
  - **Windows:** Must start with `C:\` (or `D:\`, `E:\` etc)
- Will almost never need to do this
  - Instead, will learn drag-and-drop trick later

Starting Python

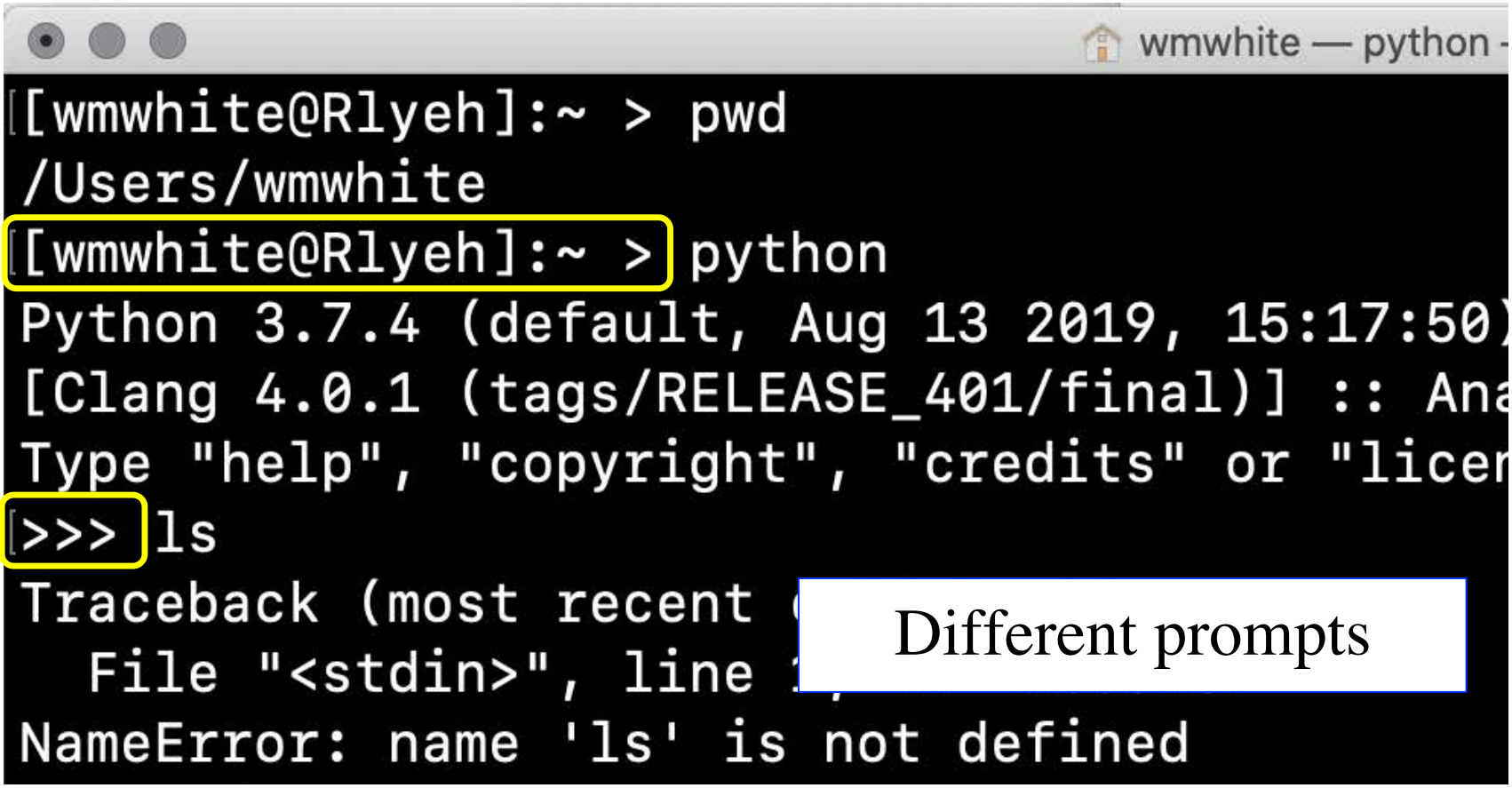
- Python is a *scripting* language
  - Designed to automate tasks on your computer
  - Uses the command line as interface
  - So we have to access Python from command line
- To start Python just type `python`
  - Drops you in the interactive shell
  - Type in Python commands & responds
  - Try typing `1+1` as an example

Python and the Command Line

```
wmwhite — python — 80×24
[wmwhite@Rlyeh]:~ > pwd
/Users/wmwhite
[wmwhite@Rlyeh]:~ > python
Python 3.7.4 (default, Aug 13 2019, 15:17:50)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> ls
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'ls' is not defined
>>> pwd
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pwd' is not defined
>>>
```

- Not the Terminal!
  - No ls, cd, pwd
  - Type quit() to exit

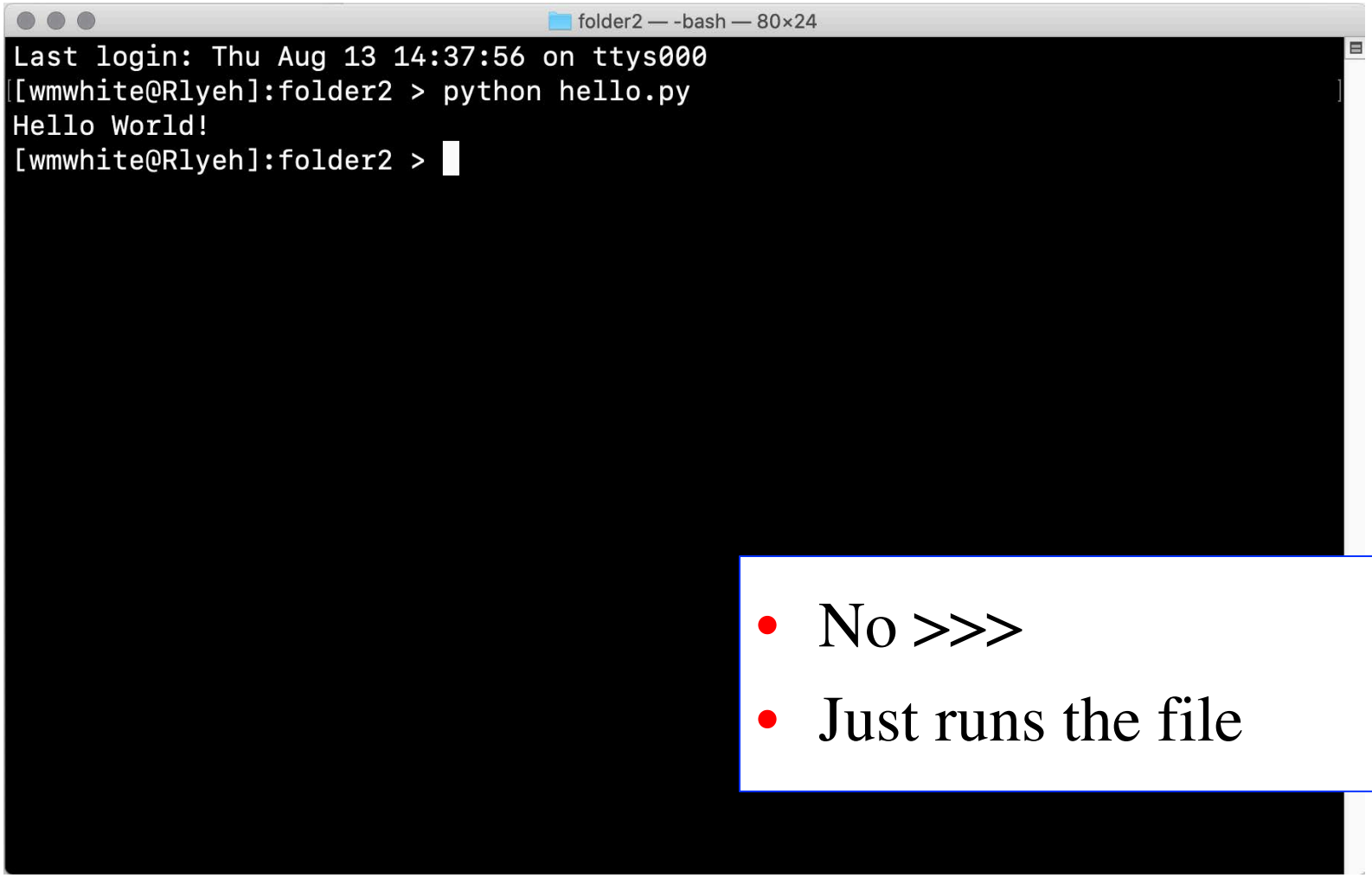
Python and the Command Line



Python Scripts

- Interactive shell just runs one command at time
  - Not very effective for complex projects
- A **script** is a file containing Python code
  - Ends with the suffix `.py`
- Examples available for download
  - Will not understand the contents at all
  - But you can run them anyway!
  - Common workflow – someone hands you a script

Basic Example



Visual Scripts

- Some Python scripts can create windows
- **Example:** `hello_tk.py`
  - Note that we do not return to prompt
  - Still producing the window on the desktop
  - Have to close the window
- **Example:** `hello_kv.py`
  - Looks the same as `hello_tk.py`
  - But uses Kivy (an add-on), not TCL/Tk (built-in)
  - We like Kivy because it is more powerful

Command Line Shortcuts

- Tab completion: start typing and hit tab
  - **Linux, MacOS:** will complete to nearest match
    - If two files have same prefix, stops at difference
  - **Windows:** will complete to **FIRST** match
    - Less useful if two files have same prefix
- Up arrow/down arrow navigate history
  - Useful when repeating tasks
  - Use left/right arrow to edit what you typed
- Works **inside** of Python as well!

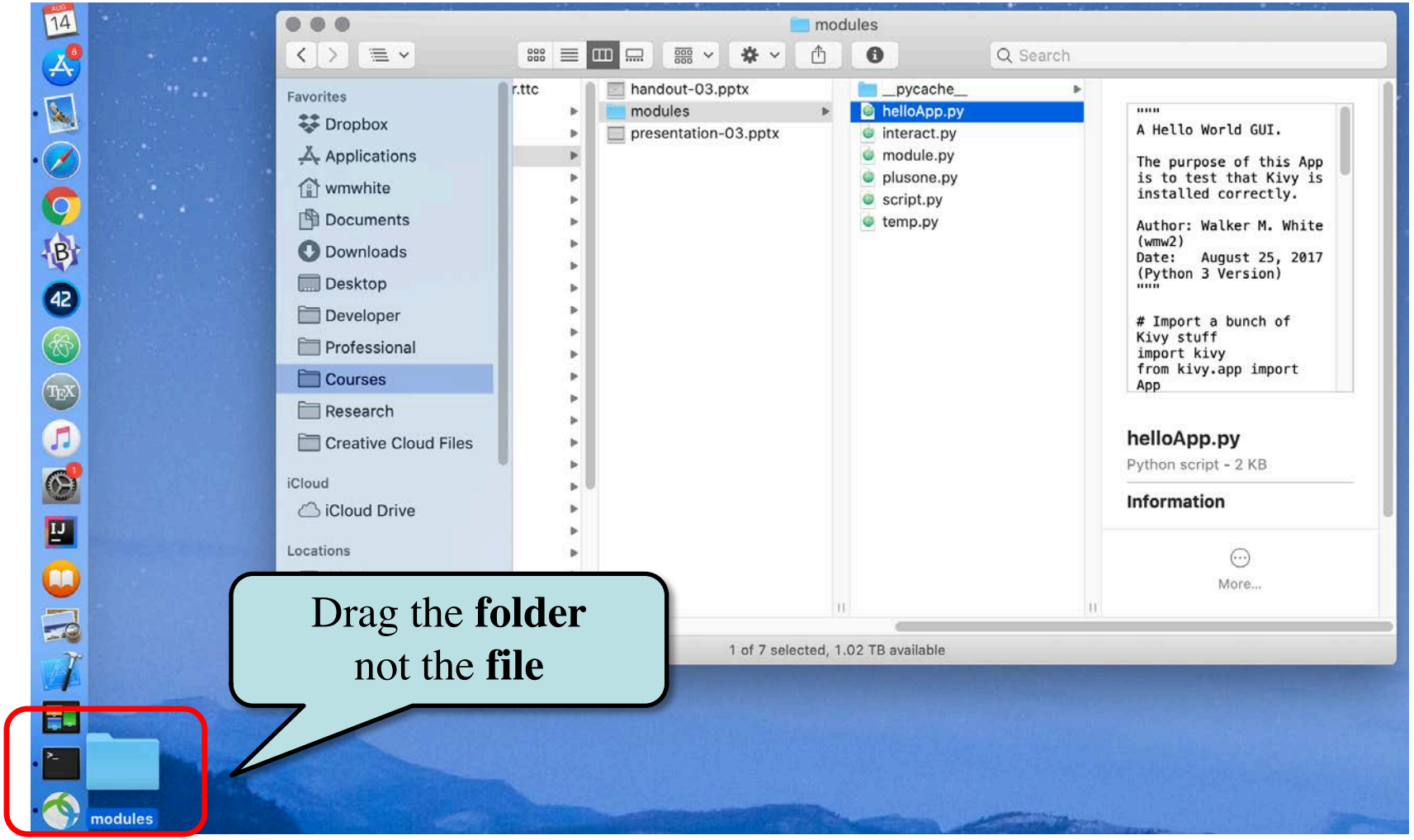
Clearing the Screen

- The command line can get really busy
  - May need to scroll back if you missed something
  - Look at `hello_ky.py` output as example
  - Clearing screen periodically makes this easier
- On all platforms, can type `clear`
  - But only outside of Python
  - So limited usefulness with Python
- **MacOS:** Command-K works inside of Python
  - Will use this a lot in the videos

Drag-and-Drop (MacOS)

- Navigation can get tricky really fast
  - Simple if you just dump all files on Desktop
  - But that is a bad idea; should organize files
- The drag-and-drop trick
  - Make sure you have the Terminal in the dock
  - Take the folder containing the file (not file)
  - Drag the folder on to the doc
  - This opens a new Terminal window in place

Drag-and-Drop (MacOS)



Drag-and-Drop (Windows)

- Would like to use the MacOS drag-and-drop
  - Windows has something similar, but not as nice
  - Cannot get a new window, but can change existing
- The drag-and-drop trick
  - Open up the PowerShell. Type '`cd` ' (SPACE)
  - Take the folder containing the file (not file)
  - Drag the folder on to the Powershell
  - Hit return to change the directory

Drag-and-Drop (Windows)