

Hack Astro Academy



Owen Chase and Óscar Chávez Ortiz

Introducing the Instructors: Owen

- 2nd year grad student in astronomy
 - Cosmology and large-scale structure
- Been coding for a long time
 - (still constantly look things up as I work)
- Grew up in PA

Introducing the Instructors: Oscar

- 5th year graduate student
- Works on studying very distant galaxies
- Has coded with Python for 7+ years
- Learning R for a class atm
- Taught Python for Astronomers at UC Berkeley
- Taught the Python bootcamps for the REUs at UT Austin
- Go to person for Mac user

Faculty Survey: What should students know?

How to **install a python package**. How to read in various file formats, especially **fits files**, and how to use **pandas**. How to do basic **plotting using matplotlib**. How to fit trends to data, e.g., **scipy**. How to use basic astronomical packages, i.e., **astropy**. Should be able to use coordinates, constants, astronomical units. How to **calculate things in python**.

Basic proficiency with **Python**

Python, command line, bash, awk (just kidding on awk...)

Python, pip, GitHub, numpy, scipy

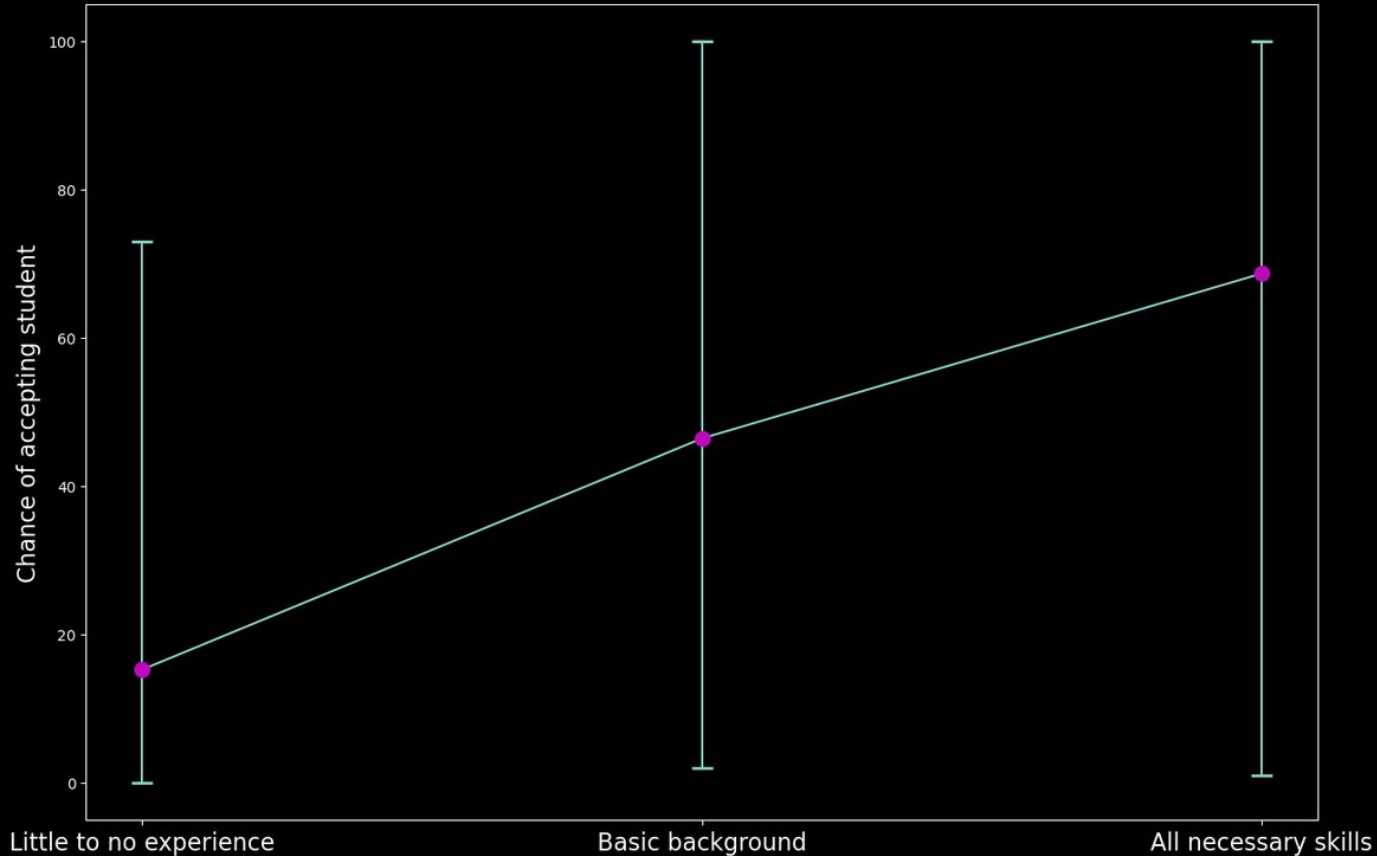
Faculty Survey: What should students know?

...the ability to open an existing **Jupyter notebook** and run its contents...
they need to know **what a function is**, what **numpy/astropy**/etc are (even if they don't know very many of the specific functions), and how **matplotlib** works at least in a very cartoon view...

Python, preferably C or C++, FORTRAN

unix **command line** (navigation, running code, etc.), **reading in files, writing files, using a for loop, plotting data**, running calculations on data, interpolating data, **looking up how to do stuff** (e.g. reading/parsing documentation for a new thing in Astropy, numpy, etc.). Some **conda** environment stuff too -- creating a new environment for a new project especially if using specific code with dependencies.

Faculty Survey: Would you bring on a student?



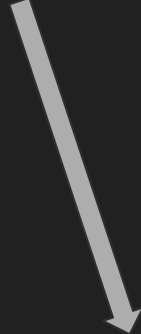
Learning Objective:

Describe and navigate the structure of a
UNIX file system

UNIX



Linux



macOS

DOS



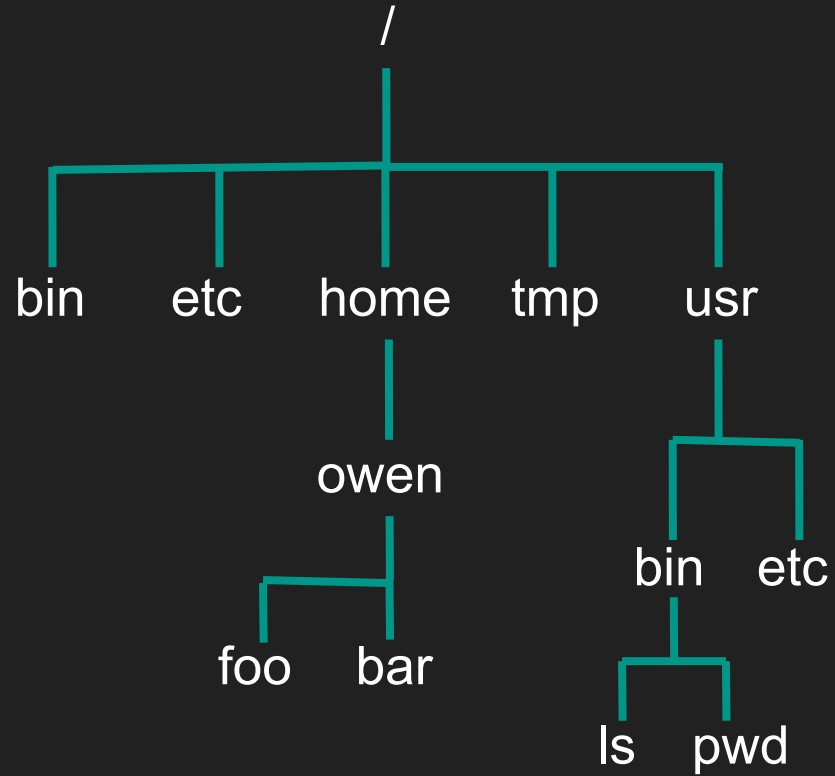
MS-DOS



Windows NT



File system



Files

- Everything is a file
 - Directories are a special type of file
- Files can have any name using any characters
 - All files in a directory must have unique names
 - File names are case sensitive
 - Do not require an extension
- All files are treated the same by Linux
 - It is up to you and your programs to handle them properly

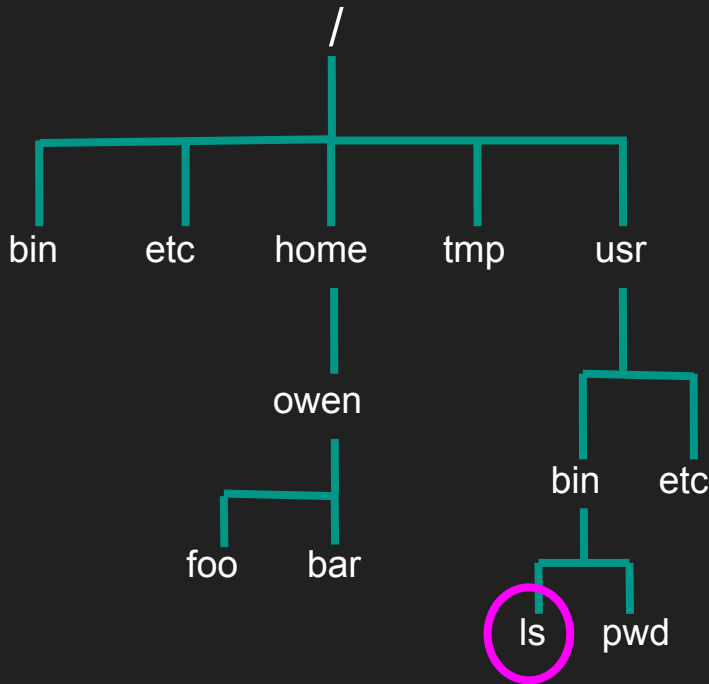
Paths

Absolute paths:

- Start with / and list the whole tree to the file of interest
- `/home/owen/foo`

Relative paths:

- Reference a file relative to the *current working directory*
- Special files `.` and `..` and `~`
 - `.` is the current directory
 - `..` is the parent of the current directory
 - `~` is the home directory of a user
- `../usr/etc`



Useful commands

- pwd
- ls
- cd
- mkdir
- rmdir
- rm
- find
- head
- tail
- more/less
- chmod
- grep
- cat
- awk

pwd (print working directory):

- print the name of the current working directory (cwd) to the screen

ls (list):

- List the files in the cwd

cd (change directory):

- Move to a different working directory

rm (remove):

- Remove a file (or directory)

mkdir (make directory):

- Create a new directory

cat (concatenate):

- Print the contents of a file to the screen

touch:

- Create a new file (or update most recent active time)

head/tail:

- Print the first/last 10 (default) lines of a file to the screen

grep (global regular expression print):

- Search for a pattern within files

find:

- Searches for files or directories

How to invoke a command:

- find [path] [options] [expression]
- find /usr/bin/ -name "ls"