

terminalExercise

January 11, 2023

0.1 File System Exploration and Easter Egg Hunt

Open a terminal (Mac, Linux) or a terminal or powershell (Windows). Note for Windows users, make sure you are using Terminal or Powershell, not the older Console so that the commands are all the same as for Linux.

Verify that you're in your home directory with `pwd` (Print Working Directory).

If you're not, magically transport yourself there with `cd ~` (Change Directory to home).

Look at the files and folders in your home directory using `ls`. Also look using `ls -p`, which will put a `/` after anything that is a directory.

The purpose of most of the folders in your home directory will be clear (e.g. "Pictures" or "Downloads"), but some may not be (e.g. "Library"). Explore any and all of these mystery directories and see if you can figure out what they are for.

Once you've explored your home directory, pop up to the root level (`cd /`). Explore these directories a little bit. Are there any directories that have the same names as directories in your home folder? If so, why might that be?

Show Larry or Franco any interesting things you find along the way.

Type `date` into the terminal.

This is a "command" and, as such, it has to be instantiated in code somewhere. See if you can find it in your filesystem!

Here are a couple hints: 1) `date` is available to all potential users of your computer, not just you. 2) it is compiled code, and therefore a "binary".

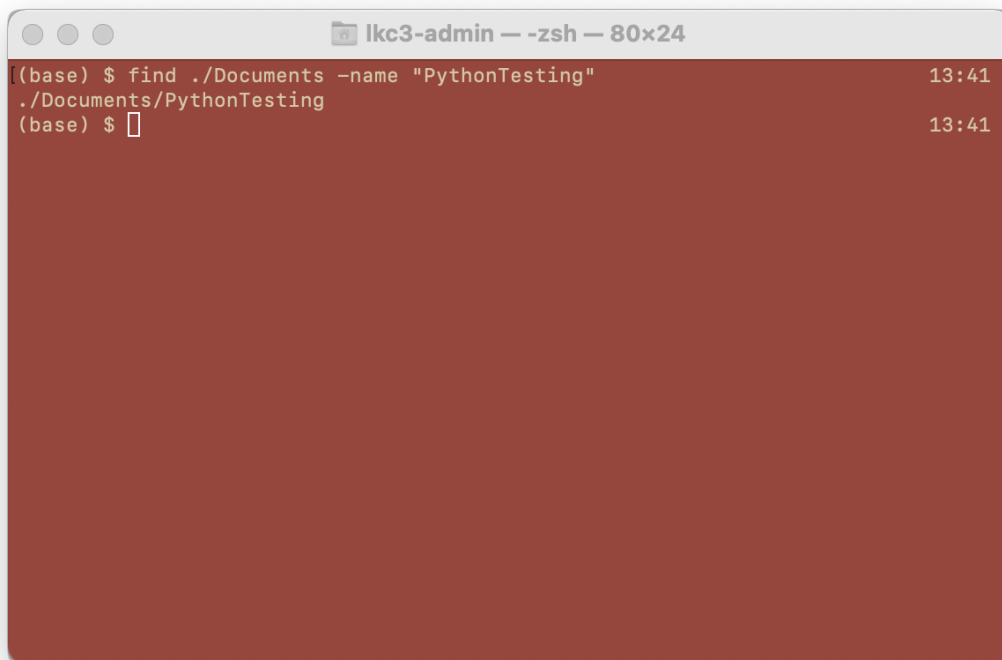
Feel free to explore "by hand" using `cd` and `ls`.

You can also experiment with the `find` command.

The `find` command takes the form `find <path> -name "wantedfile"`.

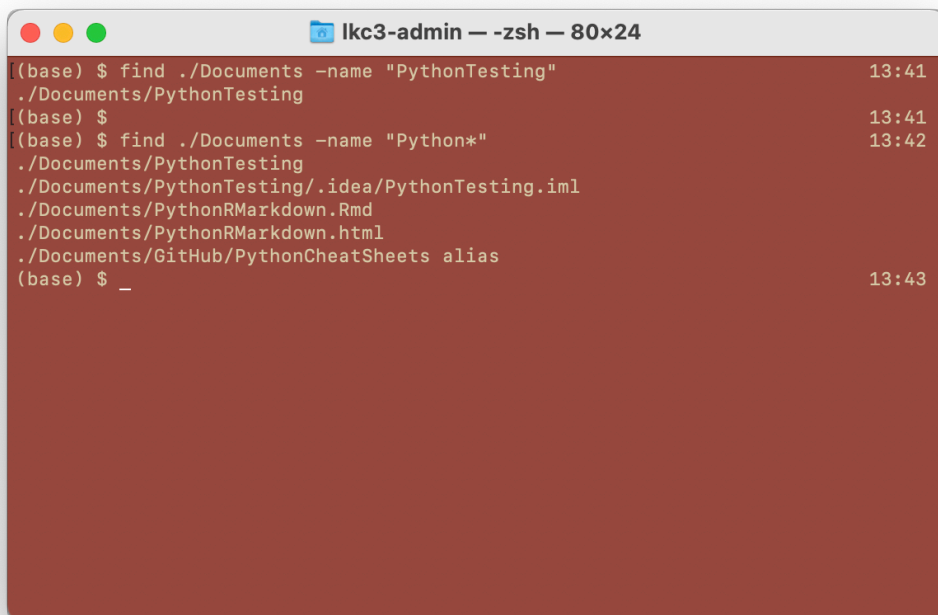
Here are two examples of the `find` command used from my home folder:

This finds anything named (exactly) "PythonTesting" in my Documents folder:

A terminal window titled "lkc3-admin — -zsh — 80x24" with a dark red background. It shows a command prompt where the user runs `find ./Documents -name "PythonTesting"`. The terminal outputs `./Documents/PythonTesting` and then returns to the prompt. The time 13:41 is shown on the right side of the first two lines.

```
(base) $ find ./Documents -name "PythonTesting" 13:41 ]
./Documents/PythonTesting
(base) $ 13:41
```

And this finds anything whose name starts with “Python” in my Documents folder:

A terminal window titled "lkc3-admin — -zsh — 80x24" with a dark red background. It shows a sequence of commands. First, the user runs `find ./Documents -name "PythonTesting"` and gets `./Documents/PythonTesting`. Then, the user runs `find ./Documents -name "Python*"`, which lists several files and directories starting with "Python". The time 13:41, 13:42, and 13:43 are shown on the right side of the first, third, and fifth lines respectively.

```
(base) $ find ./Documents -name "PythonTesting" 13:41 ]
./Documents/PythonTesting
(base) $ 13:41 ]
(base) $ find ./Documents -name "Python*" 13:42 ]
./Documents/PythonTesting
./Documents/PythonTesting/.idea/PythonTesting.iml
./Documents/PythonRMarkdown.Rmd
./Documents/PythonRMarkdown.html
./Documents/GitHub/PythonCheatSheets alias
(base) $ _ 13:43
```

The splat (“*”) is a wildcard, meaning than anything will match it. It can go at the beginning or in the middle also, and you can use more than one (so, e.g., `*.*` would match any filename with

an extension).

Once you've found date, try typing `cal` into your terminal. Assuming you have it, try finding it! (If you can't find it and are getting frustrated, ask for help.)

As our last easter egg hunt, consider that Python itself is also just launchable code. So let's try and find Python! (There actually might be multiple installations of Python on your computer.) Specifically, let's try and find iPython, which is a cool version of Python to run at the command line and is also what Jupyter Notebooks run under the hood.

Wherever it lurks, the actual file will be called "ipython".

Once you've found it, run it at the command line and do some Pythony things!!!

[]: