THE COMMAND LINE

Objectives

- Define what the command line is
- Learn command line navigation and file structure
- Learn to manipulate files and folders via command line

What the Command Line Does

The command line (or terminal) is a faster and more powerful way to maneuver your operating system than by using a GUI (**g**raphical **u**ser **i**nterface), such as Windows Explorer or Mac Finder.

Use special keywords to do everything you can with a GUI *and more.*

```
Get-Service | Where-Object {$_.Status —eq “Running”}
```

Note about Windows vs. OSX

OSX (Mac) and Linux operating systems are both based on Unix and share many of the same terminal commands.

Windows has an entirely different *kernel* (base) and has traditionally had its own command-line syntax for MS-DOS and command prompt.

However, **Powershell** has adopted **aliases** that mimic the most common Unix commands.

Powershell

Powershell ships with all modern Windows systems (since Windows 7) and is a superior shell to command prompt. It compares favorably with Unix-based shells (such as bash).

The commands we will learn in Powershell are mostly Unix-like aliases for "cmdlets" that do things like change directories, manipulate files, and so forth.

You can write really powerful scripts with Powershell, which is built on the .NET platform. We will not cover that in this course.

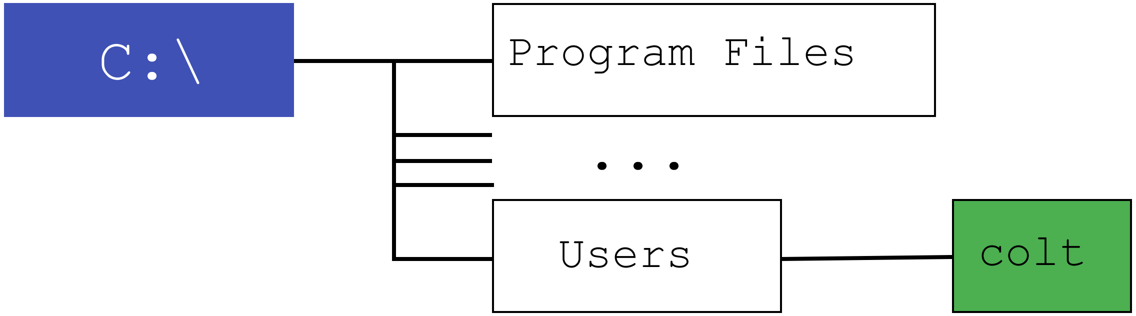
How we'll use it

We will use the terminal to:

- navigate around
- create and remove directories and files
- move, copy, and paste things
- later, execute Python scripts

OS File Structure

Operating Systems organize their folders in a hierarchy (a tree) with parents and children, all relative to a base **root** directory.

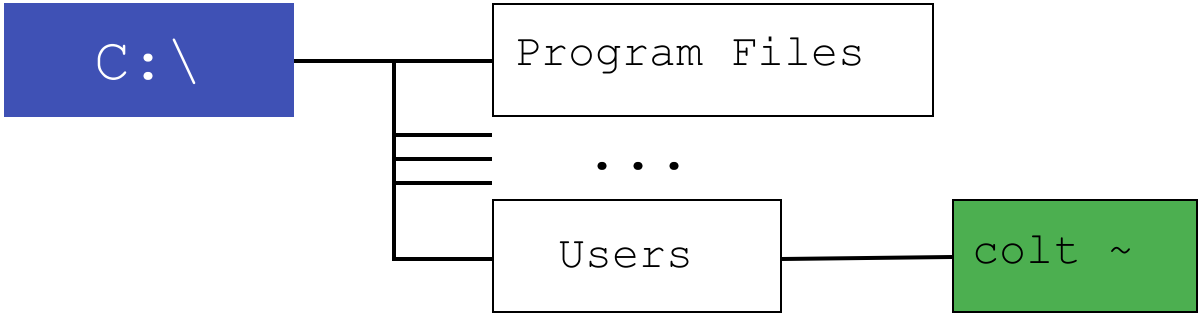


Files and directories have **absolute paths** based on the root, where each additional level down adds a "\".

The absolute path for "Colt" is: `C:\Users\colt`

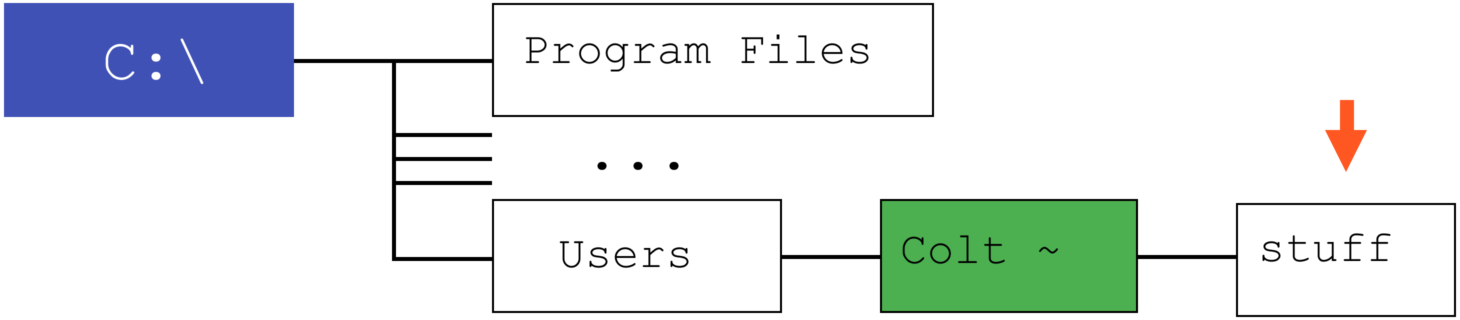
Where am I?

The **green** directory below is a special directory called "home", which is also known as "~". This is the default directory upon opening your terminal.



How do I find out where I am?

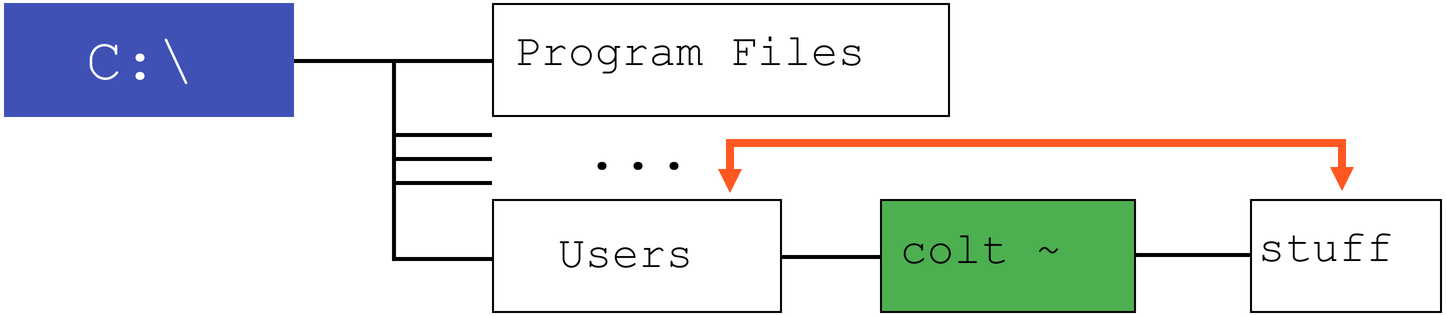
The cmdlet "**pwd**" (**p**rint **w**orking **d**irectory) will tell you the full *absolute path* of where you're at!



```
pwd
Path
----
C:\Users\colt\stuff
```

Navigating *Absolutely*

The command "**cd**" ("**c**hange **d**irectory") followed by the absolute path of the folder will navigate you directly there.



C:\

Program Files

. . .

Users

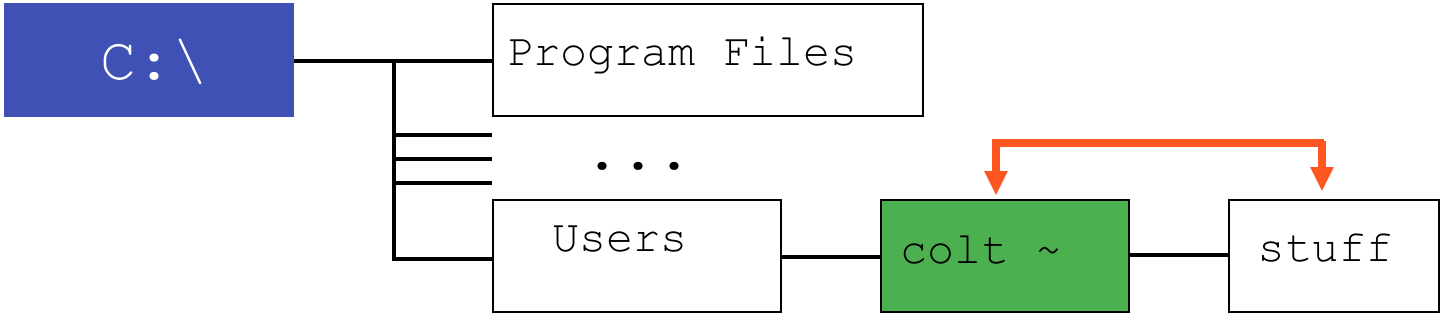
colt ~

stuff

```
pwd
Path
----
C:\Users\colt\stuff
cd C:\Users
pwd
C:\Users
```

Navigating *Relatively*

The dot "." stands for current directory, and dot-dot ".." stands for parent directory. This allows for relative navigation:



```
pwd
Path
----
C:\Users\colt\stuff
cd ..
pwd
C:\Users\colt
```

What's Inside?

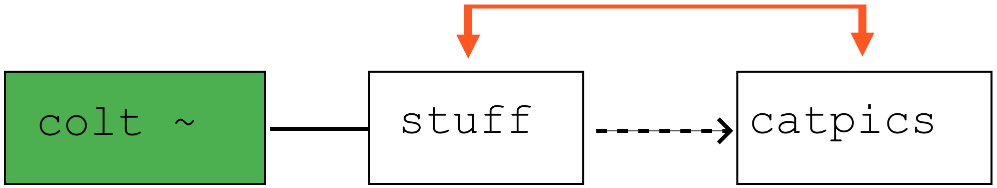
The keyword "**ls**" will "**l**i**s**t" the contents of a directory.

```
ls

    Directory: C:\Users\colt

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----       12/28/2017   3:31 PM                .config
d-r---         1/5/2018   9:26 PM                Contacts
d-r---        1/10/2018   9:55 PM                Desktop
d-r---         1/5/2018   9:26 PM                Documents
d-r---         1/5/2018   9:26 PM                Downloads
d-r---         1/5/2018   9:26 PM                Favorites
d-r---        1/21/2018   3:03 PM                Google Drive
d-----         2/7/2017   3:49 PM                Intel
d-r---        1/18/2018  11:01 PM                Links
d-r---         1/5/2018   9:26 PM                Music
dar---        1/21/2018   3:03 PM                OneDrive
d-r---         1/5/2018   9:26 PM                Pictures
d-----       12/28/2017   3:21 PM                projects
d-r---         1/5/2018   9:26 PM                Saved Games
d-r---         1/5/2018   9:26 PM                Searches
d-----        1/21/2018   3:31 PM                stuff
d-r---        1/21/2018   3:03 PM                Videos
```

QUIZ TIME!

Creating Directories

The command "**mkdir"** ("**m**ake **dir**ectory") followed by the name of the new directory will create a new child directory inside the current directory.

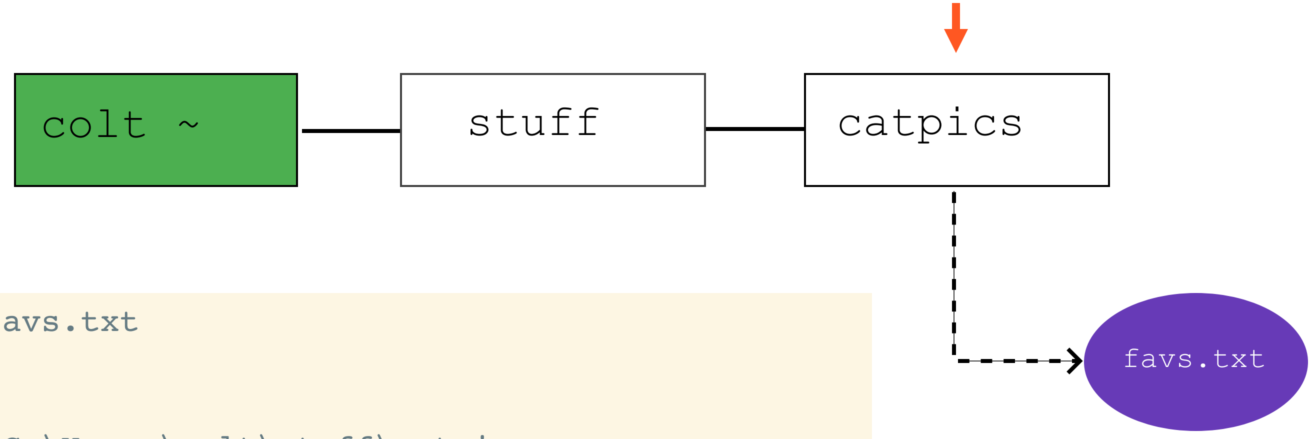


```
mkdir catpics

    Directory: C:\Users\colt\stuff

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        1/21/2018   3:39 PM                catpics


cd catpics
```

Creating Files

In Powershell, we can create files by putting empty content in them using the **echo** cmdlet.

echo $null >> followed by the filename and file-type extension will create a new file of that type.



```
echo $null >> favs.txt

ls

    Directory: C:\Users\colt\stuff\catpics

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        1/21/2018   5:28 PM              2 favs.txt
```

SUPER QUICK ACTIVITY!

- Make a new "animals" directory
- Inside of "animals" create "salamanders" and "frogs" directories
- Inside of "salamanders" add a new file "axolotl.txt"
- Inside of "frogs" add a new file: PyxicephalusAdspersus.txt (pixieFrog.txt is fine)

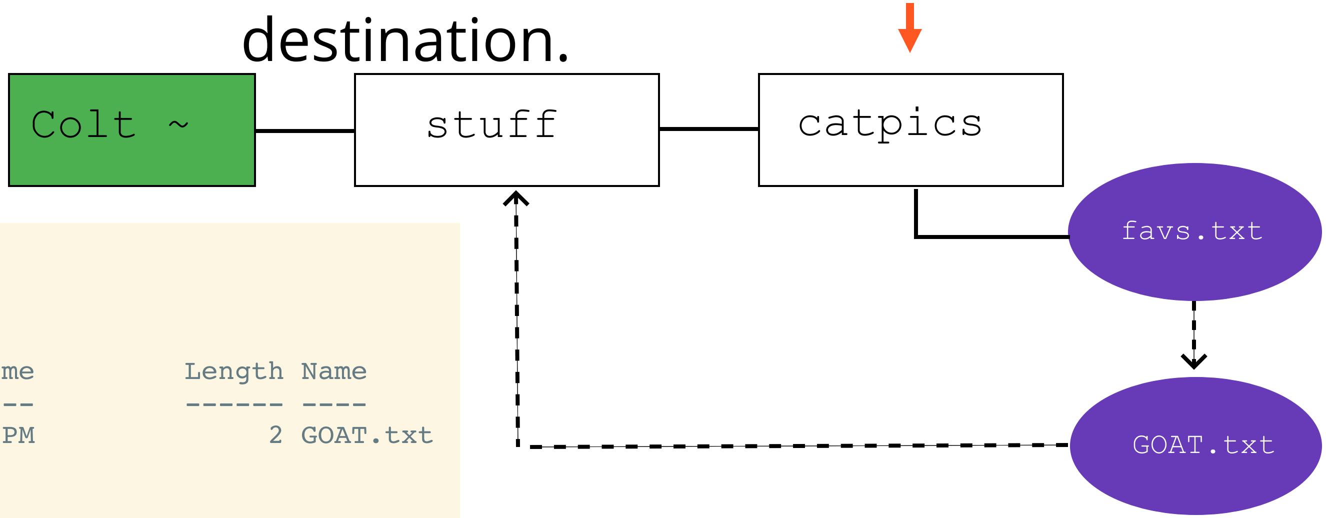
AXOLOTLS ARE AWESOME

- Really adorable smile
- Once worshipped by Aztecs
- Can regenerate limbs, skin, and spinal cord!
- 1000x more resistant to cancer than any other animal on earth!
- They glow in the dark

PIXIE

Moving / Renaming Things

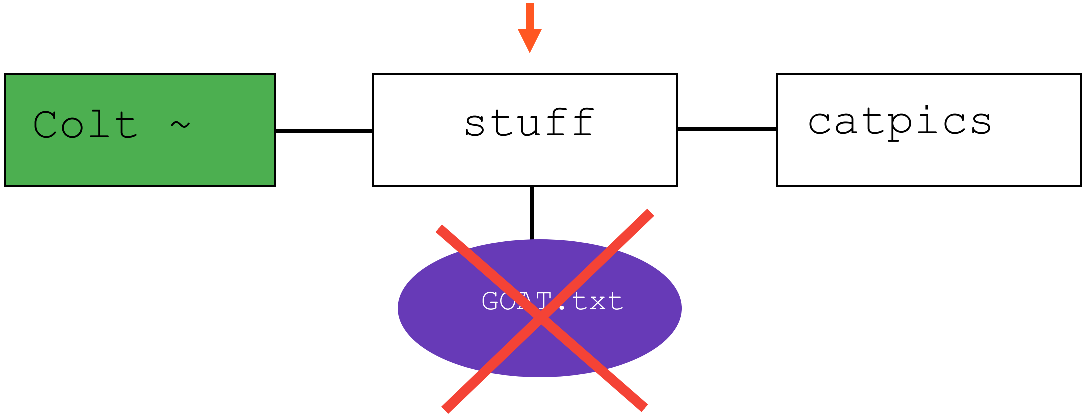
Files can be moved or renamed using the "**mv**" (" **m**o**v**e") keyword, which takes two arguments: the source and the destination.



```
mv favs.txt GOAT.txt

ls

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        1/21/2018   5:28 PM              2 GOAT.txt

mv GOAT.txt ..

cd ..

ls

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        1/21/2018   5:33 PM                catpics
-a----        1/21/2018   5:28 PM              2 GOAT.txt
```

Removing Files

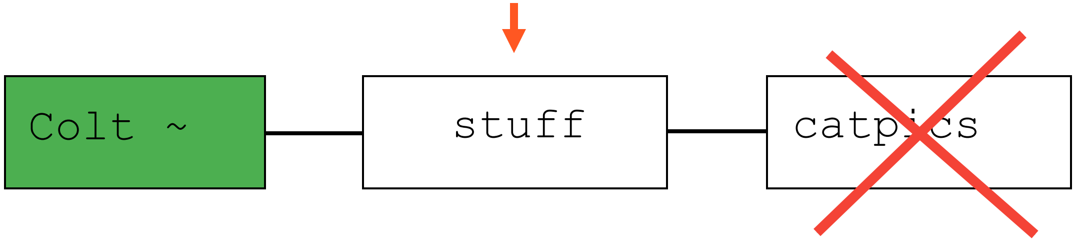
Files can be deleted using the "**rm**" ("**re**m**o**ve") keyword.



```
rm GOAT.txt

ls

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d-----        1/21/2018   5:33 PM                catpics
```

Removing Directories

Directories can be deleted using the "**rm**" keyword, with the added option "**-r**" ("**r**ecursive"). You can also add the "**-fo**" ("**fo**rce") to prevent warnings.



```
ls
catpics
rm -r -fo catpics
ls
```

Warning: "rm -r fo" is a dangerous command! Be extremely careful what folder you pass to it because you will never get it back.

QUIZ TIME!

Recap

- OS file structure is hierarchical, tree-based
- Navigate using these commands:
  - cd "change directory"
  - pwd "print working directory"
  - ls "list contents"
- Remember these aliases:
  - C:\ is root directory
  - ~ is home
  - . is current
  - .. is parent
- Manipulate files with:
  - "mkdir" create directories
  - "echo $null >> filename" create files
  - "mv" move and rename
  - "rm" to remove files, "-r" to remove directories

THE "TEST"

- Make a new folder on your Desktop called my_code
- In my_code add a new file called sheep.py
- Inside sheep.py, write python to print out "baaah"

YOUR
TURN

GIT

AND

GITHUB

Posh Git

For Powershell, we can use the Posh Git package here:

https://github.com/dahlbyk/posh-git

YOUR
TURN