Fundamentals

Joe

Today

Morning - A tour of Linux

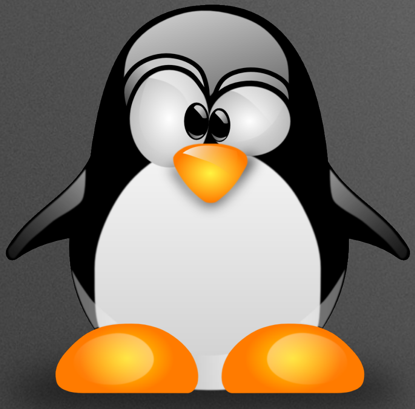
Afternoon - Philosophy of data science and this course, Data Science Workflow

Session Objective

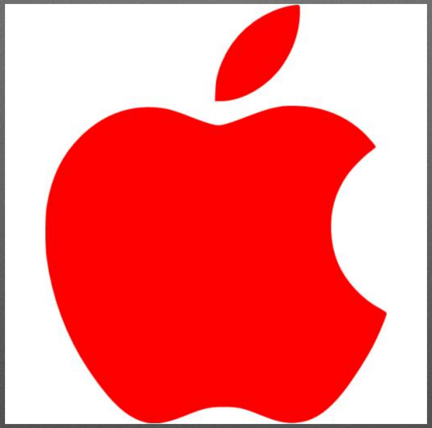
Use the command line to:

- Perform basic file operations
- Get command line instructions using man
- Manage a process with job control
- Set up a .bash_profile
- Write a regular Expression
- Use grep/sed/awk/cut/paste
- Perform a survival edit using nano

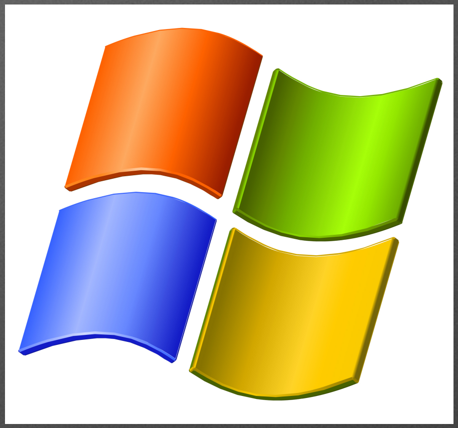
The OS of Data Science



>



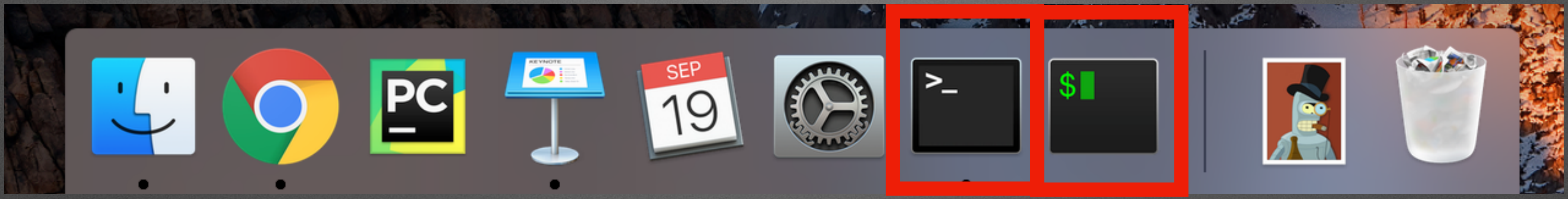
>



Linux is the open sourced version of the unix operating system. There are several ‘Flavors’ of linux (i.e. main development threads), the most commonly encountered are Ubuntu and Red Hat.

Apple’s OS is built on top of linux, so it interfaces more naturally in many scenarios, however, as of Windows 10 this gap has closed.

Linux, in a Bash Shell



There are a few flavors of shells, the most common is bash shell, and it’s the interface type we’ll be using.

There is a terminal alternative called iTerm which has some nice features, feel free to use either

The Very Basics

Navigation

- cd - change directory
  "." == current directory
  ".." == directory above current
  "~" == home area, where terminal opens to by default
  "-" == previous directory
- pwd - print working directory (think you are here)
- ls - list contents of working directory

Create/Destroy things

- cp - copy a file
- mkdir/rmdir - make/remove and empty directory
- rm - remove a file (note, this does not move to the trash, this is good bye forever)
- touch - mark last modified to right now. Creates file if it does not exist

Look at files

- cat - output file to STDOUT
- less/more - page through files
- head/tail - look at the begining/end of a file

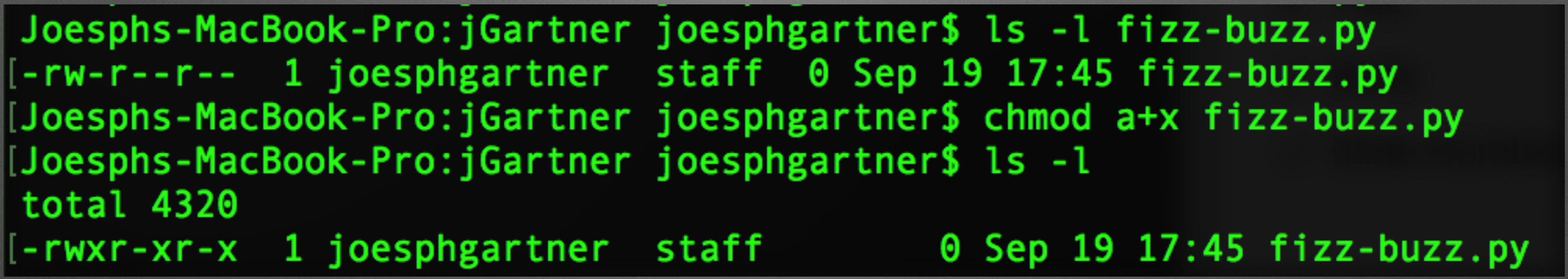
Options

- Most commands can be modified by options that are preceded by single or double dashes

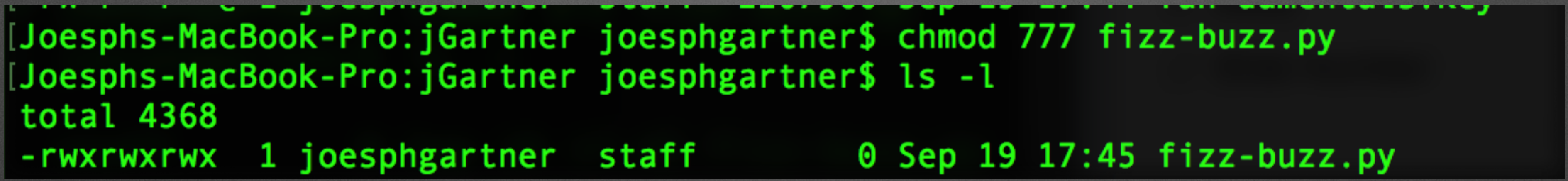
  E.G.

  - python —version (two dashes)
  - ls -l (list)
  - ls -alrth (all - list - reverse order - time order - human readable)

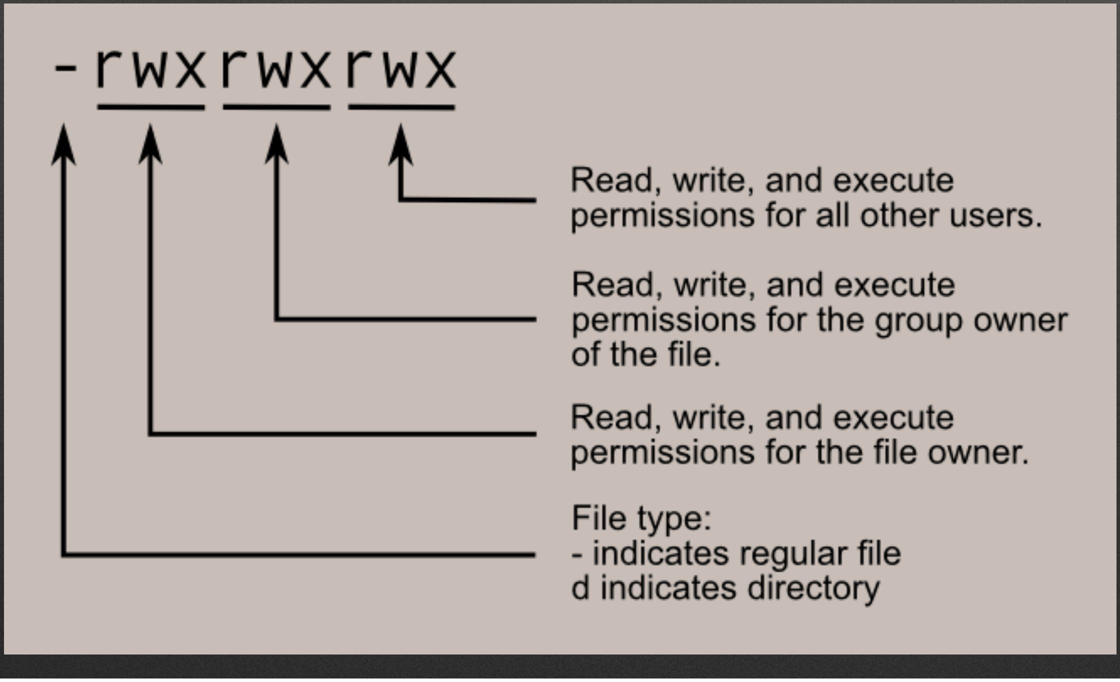
Permission



```
 Joesphs-MacBook-Pro:jGartner joesphgartner$ ls -l fizz-buzz.py
[-rw-r--r--  1 joesphgartner  staff  0 Sep 19 17:45 fizz-buzz.py
[Joesphs-MacBook-Pro:jGartner joesphgartner$ chmod a+x fizz-buzz.py
[Joesphs-MacBook-Pro:jGartner joesphgartner$ ls -l
 total 4320
[-rwxr-xr-x  1 joesphgartner  staff        0 Sep 19 17:45 fizz-buzz.py
```



```
[Joesphs-MacBook-Pro:jGartner joesphgartner$ chmod 777 fizz-buzz.py
[Joesphs-MacBook-Pro:jGartner joesphgartner$ ls -l
 total 4368
 -rwxrwxrwx  1 joesphgartner  staff        0 Sep 19 17:45 fizz-buzz.py
```



Read, write, and execute permissions for all other users.

Read, write, and execute permissions for the group owner of the file.

Read, write, and execute permissions for the file owner.

File type:
- indicates regular file
d indicates directory

Piping Output

Most commands read from STDIN and write to either STDOUT and STDERR

Python program “print” statements go to STDOUT as well!

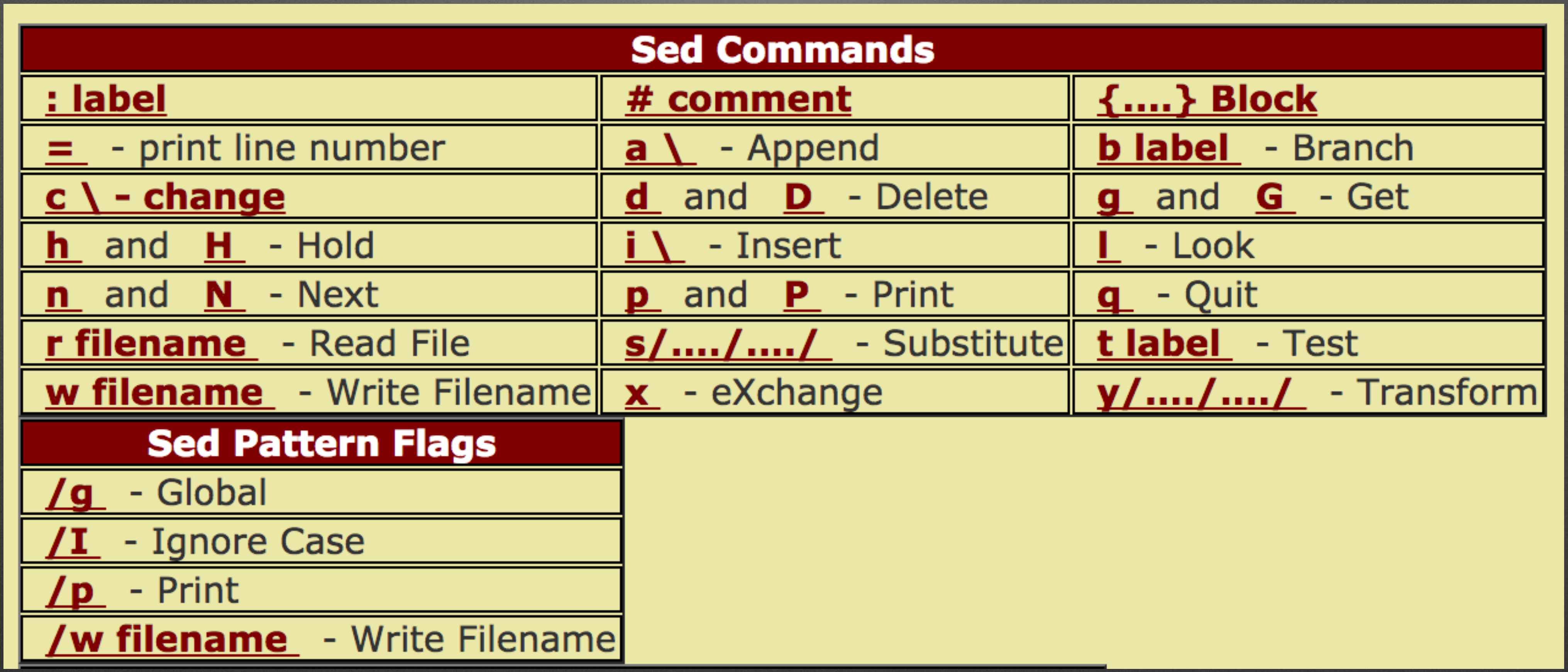
- Redirect a file to STDIN with <
- Redirect STDOUT to a file with >
- Redirect STDERR to a file with 2>
- Append STDOUT to a filewith >>
- Connect STDOUT of one command to STDIN of another with |

Filters

- grep - determines if a line of a document contains
- wc - count (lines/words/characters)
- sort - sort text doc
- uniq - combines repeated liness
- cut - select specific columns of a csv
- sed - stream editor
- awk/perl - scripting languages for stream editing

grep, sed, and awk are particularly powerful tools for search and edit. If you're hoping

SED Quick Tips



Sed Commands		
: label	**# comment**	**{....} Block**
= - print line number	**a ** - Append	**b label** - Branch
c \\ - change	**d** and **D** - Delete	**g** and **G** - Get
h and **H** - Hold	**i ** - Insert	**l** - Look
n and **N** - Next	**p** and **P** - Print	**q** - Quit
r filename - Read File	**s/..../..../** - Substitute	**t label** - Test
w filename - Write Filename	**x** - eXchange	**y/..../..../** - Transform

Sed Pattern Flags
/g - Global
/I - Ignore Case
/p - Print
/w filename - Write Filename

http://www.grymoire.com/Unix/Sed.html

Regular Expression

sed, grep (awk, perl, ect) use what’s called regular expression. Some helpful tips:

- . - anything other than new line
- * - match zero or more of previous atom
- + - match one or more of previous atom
- | - match either previous or next item
- [ ] - any characters within
- ^[ ] - any characters within
- \ - escape character
- \(\) - for grouping

Long Running Processes

- Some processes run long, and executing a script interactively means either leaving the terminal (and an ssh connection) alive, or running it in the background
  - Option 1 -> have ‘&’ at the end
  - Option 2 -> use ‘screen’

- ps = list running processes (with pids)
- jobs = list running processes (with shell id)
- kill = stop running a process (by id)
- tail -f = follow output of a file
- ^Z = stop foreground job
- ^C = kill foreground job
- bg = resume job in background
- fg = bring job to foreground

Environment Variables

- env -> list all environment variables
- THE MOST IMPORTANT VARIABLE IS “PATH”
  - This is the place your system looks when you try to execute binaries, if you break this variable, your terminal will be effectively worthless
  - You can add new things to it via:
    export PATH = /my/new/path:$PATH

- Your terminal automatically sets environment variables when you start a new shell or tab by executing a bash script called .bash_profile
- Let’s play with this a bit

Session Objective

Use the command line to:

- Perform basic file operations
- Get command line instructions using man
- Manage a process with job control
- Set up a .bash_profile
- Write a regular Expression
- Use grep/sed/awk/cut/paste
- Perform a survival edit using nano

End of Morning