Lauren Sherburne

Vim Commands:
- i = insert mode
- esc = to command mode
  - /word = search for "word" within file
  - :w = save file
  - :q = quit

File Navigation:
- vim *filename* OR touch *filename(s)* = create and open new file
- pwd = print working directory
- mkdir *directory name* = make directory
- rmdir *directory name* = remove directory
- mv *filename* *directory name* = move file(s) or directories to another directory
- cp *filename* *directory name* = copy file(s) or directories to another directory
- cd = change working directory
  - cd ~ = shortcut to home directory
  - cd .. = navigate to parent directory (. = current directory)
  - cd directoryname = navigate to "directoryname" directory
- ls = list everything within the current file
  - ls -1 = (-one) lists each filename on its own line
  - ls -l = (-letter L) long list of everything in the current directory
- man *command* = display the user manual of a given command
- apropos *keyword* = help search for a command when you cannot remember
  - apropos -e OR apropos -exact = search exact keywords
- less *filename* = read a text file one page (one screen) at a time
- find = used to find files or directories
- file = used to determine type of file
  - * = matches one or more occurances of any character
  - ? = matches a single occurance of any character
  - [ ] = matches occurence of character(s) enclosed
  - { } = terms are separated by commas and each term must be the name of something or a wildcard
  - ! = means not-something
- stat = provides info about file: size, inode number, access permissions, time of last modification, etc.
- df = displays the amount of disk space available on current file system
- wc *filename* = used to find the number of lines, word count, bytes, and character count in specified files
- tar *file or directory name name* = used to create archive and extract archive files and/or directories
  - tar cvf *file(s)* = used to create a compressed file
  - tar xvf *filename*.tar = used to uncompress file
- chmod = customize file permissions
  - chmod u+x -*filename* = give permission
  - chmod u-x -*filename* = take permission
    u = user
    a = all

Input/Output Manipulation and Variables:

- echo = display line of text to terminal
  - echo $*variable name* = prints variable
  - echo "*text*" = prints text
  - echo *filename = print file contents
  - echo -e = allows terminal to understand:
    \b = removes any space between words before and after command
    \n = new line
    \r = x characters after this replace the first x character
    \v = new line + indent
- cat = reads data from the file(s) and gives their content as output
  - cat -n = prints line numbers
- > OR 1> = standard output
  - *command* > *filename* = output of command is written to file instead of to terminal
  - *command* >> *filename* = output of command is appended to file
- < OR 0< = standard input
  - *command* < *filename* = input can be redirected from a file
- 2> OR 2>> = standard error
  - *command* 2> *filename* = can filter out the error messages from a command result and save them to a file
  - *command* 2>> *filename* = can filter out the error messages from a command result and append them to a file
- /dev/null = black hole file :) basically a trashcan that you cannot access
  - find /name "*" -print 2> /dev/null = this example discards any error messages that are generated by find command
- export = marks an environment variable to be exported with any new program/script and thus allows it to inherit all marked variables
- alias = replace one string with another
- unalias *alias name* = remove an existing alias

Important Extras:
Double Quotes "..." - use when you want to enclose variables or use shell expansion inside a string, all characters within are interpreted as regular characters except $ or ` ` which will be expanded on the shell
Single Quotes '...' - all characters within are interpreted as string characters
run script ./-scriptname

Command Line Arguments:
$0 = filename of currents script
$n = here n is a positive decimal number which corresponds to the arguments with which a script was invoked
$# = the number of arguments supplied to a script
$* = access all arguments (as one)
$@ = access all arguments
$$ = the process number of the current shell
$! = the process number of the last background command
$? = exit status of a linux command (max exit status = 255)

Control Signals

- ctrl + C = interrupt/kill whatever you are running
- ctrl + L = clear the screen
- ctrl + S = stop the output to the screen
- ctrl + Q = allow output to the screen (after ctrl s)
- ctrl + D = EXIT; send an eof marker, will close the current shell
- ctrl + Z = send the signal SIGTSTP to the current task, which suspends it; to return to it later enter fg 'program name'