Objectives

Questions

* How can I perform the same actions on many different files?

Objectives

* Write a loop that applies one or more commands separately to each file in a set of files.
* Trace the values taken on by a loop variable during execution of the loop.
* Explain the difference between a variable’s name and its value.
* Explain why spaces and some punctuation characters shouldn’t be used in file names.
* Demonstrate how to see what commands have recently been executed.
* Re-run recently executed commands without retyping them.

Video notes

1. change directory to **data-shell** folder, type:

pwd

cd Desktop/data-shell

ls

2. change directory to **creatures** folder, type:

cd creatures

ls

3. See the top 5 lines of a list of files, type:

head -n 5 basilisk.dat minotaur.dat unicorn.dat

(it shows top 5 lines of the 3 files)

4. a simple loop: extract only second line, type:

one way:

for filename in basilisk.dat minotaur.dat unicorn.dat

do

head -n 2 $filename | tail -n 1

done

alternative way:

for x in basilisk.dat minotaur.dat unicorn.dat

do

head -n 2 $x | tail -n 1

done

(Here, use “x” instead of “filename”, still works. But filename may make better sense)

5. a more complex loop, type

for filename in \*.dat

do

echo $filename

head -n 100 $filename | tail -n 20

done

6. loop, where file name “red dragon.dat” and “purple unicorn.dat” don’t exist, type:

for filename in “red dragon.dat” “purple unicorn.dat”

do

head -n 100 $filename | tail -n 20

done

(for file names with space, use “ ”)

7. Make a copy of original files, type:

(if use cp \*.dat original-\*.dat, it doesn’t work, since cp cannot copy more than 2 files at a time. In this case, use for loop)

for filename in \*.dat

do

cp $filename original-$filename

done

ls

8. change directory to **2012-07-03** folder, type:

pwd

cd ..

ls

cd north press tab, 2012 press tab (get north-pacific-gyre/2012-07-03/)

9. find all files start with “NENE” and end with “A” or “B”, type:

for datafile in NENE\*[AB].txt

do

echo $datafile

done

(\*[AB] means the file names contain either A or B)

10. (1) find all files start with “NENE” and end with “A” or “B”, and rename them to stats-, type:

for datafile in NENE\*[AB].txt

do

echo $datafile stats-$datafile

done

(2) An easier way (not write code every time), now use upper arrow and change the previous code

for datafile in NENE\*[AB].txt; do echo $datafile; **bash** **goostats** $datafile stats-$datafile; done

(add bash goostats $datafile)

11. Look at history commands, can re-run those code, or export it in a file and re-use it, type:

history | tail -n 5

!!

(retrieves most recent command)

!$

(retrieves the last word of last command)

Quizzes

**Question 1**

Graphical user interface, text, application, email

Description automatically generated

**Question 2**

**Graphical user interface, text, application, email

Description automatically generated**

My answers:

Version 1: It creates a new .pdb file named all.pdb

Version 2: It just prints out cat \*pdb >> all.pdb