Software Carpentry

Command cheatsheet

Basic commands for Bash (Unix) shell and Git

Bash: navigating the file system

`pwd` print working directory

`ls` list directory

- `ls -l`: list a lot of file information
- `ls -lh`: list a lot of human-readable file information

`cd` change directory

Bash: interacting with files and directories

`mkdir` make directory

`cat` send file or files to output (in most cases, this shows the content of a file without having to open it)

`head` output first parts of a file or files (default is usually 10 lines)

`tail` output last parts of a file or files (default is usually 10 lines)

`mv` rename or move a file or files. Syntax for renaming a file: `mv FILENAME NEWFILENAME` **USE WITH CAUTION!!!**

`cp` copy a file or files. Syntax: `cp FILENAME NEWFILENAME`

`rm` remove a file or files. **USE WITH CAUTION!!!**

Bash: getting information about file contents

`wc` counts the number of lines, words, and characters in files

- `wc -l`: counts only the number of lines
- `wc -w`: counts only the number of words

- `wc -c`: counts only the number of characters

`sort` sends an alphabetically sorted list of the contents of a file to output (usually to the screen). Does not change the file itself.

- `sort -n`: sorts the output numerically
- `sort -r`: reverses the order of the sort, e.g.: Z-A or 10-01

Bash: pipes and filters

`*` wildcard character that matches 0 or more characters

`?` wildcard character that matches exactly one character

`>` redirect output to a new location. Syntax with `cat`: `cat FILENAME1 FILENAME2 > NEWFILENAME`
USE WITH CAUTION!!!

`>>` append output to an existing location. Syntax with `cat`: `cat FILENAME1 FILENAME2 >> FILENAME3`

`|` called a pipe. Takes the output of one command and sends it to another command. Syntax with `wc`, `sort`, and `head`: `wc -l FILENAMES | sort -n | head -n 1`

Bash: syntax of a for loop

```
for filename in basilisk.dat unicorn.dat
do
head -n 3 $filename
done
```

Context of this for loop:

- "filename" is the variable named in the first line and called ("$filename") in the third line of the loop
- The loop is operating on the two files named in the first line, basilisk.dat and unicorn.dat
- The third line shows what is being done to the two files; in this case, showing the first three lines of each file.

Basic Git commands

`git init`: creates a git repository

`git status` : view the status of your files in the working directory and staging area

`git add`: tells git to start tracking a file, or a series of files.

`git commit`: commits (saves) the staged snapshot to the project history.

`git log`: shows all the commits in the project history

`git diff`: shows changes made to files

`git remote add origin`: add a remote repository where changes will be stored, usually for collaboration

`git push`: sends local changes to a remote repository

`git pull`: brings changes made in a remote repository to the local repository

---

Resources

Software Carpentry Bash (Unix) Shell Lesson: http://swcarpentry.github.io/shell-novice/ (http://swcarpentry.github.io/shell-novice/)

Software Carpentry Git Lesson: http://swcarpentry.github.io/git-novice/ (http://swcarpentry.github.io/git-novice/)

Source

Author: Jamene Brooks-Kieffer

Affiliation: University of Kansas Libraries

Link: https://github.com/kulibraries/swc-workshop-helps/blob/master/command-handout.md (https://github.com/kulibraries/swc-workshop-helps/blob/master/command-handout.md)

Modified from James Baker's original Library Carpentry Git handout: https://github.com/LibraryCarpentry/week-three-library-carpentry-DEPRECATED/blob/master/handout.docx (https://github.com/LibraryCarpentry/week-three-library-carpentry-DEPRECATED/blob/master/handout.docx)