useful UNIX commands for CIS 180 - Intro to Python

* **remember:** UNIX is CASE-SENSITIVE! (So is Python...)

* once you have logged on to cs-server or redwood, here is a collection of UNIX commands that you might find useful in CIS 180 - Intro to Python:

python-specific commands

python	...will start up the python interpreter on either cs-server or redwood. (use ctrl-d to exit)

if you have a script of Python commands *myscript.py*, then:

python *myscript.py*	...will execute that script at the UNIX command-line.

if the FIRST line of your script on cs-server is: `#! /usr/bin/python`
                       ...or on redwood is: `#! /usr/local/bin/python`
...and if you make that script executable using **chmod 700** *myscript.py* , then

myscript.py	...will execute that script at the UNIX command-line

help-related commands

man *desired_command*	print the UNIX manual page for *desired_command*, if it exists. (try w/**man python**)
apropos *string*	print names of UNIX commands followed by 1-line descriptions whose 1-line descriptions contain *string* within them.

directory-related commands

cd, **cd** *dirname*	change directory command; make the current working directory *dirname*, or make the home directory the current working directory if no directory name is given
.	a nickname for the current directory
..	a nickname for the parent of the current directory
~username	a nickname for *username*'s home directory
~	a nickname for the current user's home directory
pwd	gives the name of the current (present) working directory
mkdir *dirname*	make a new directory named *dirname* in/under the current working directory
rmdir *dirname*	remove the directory *dirname* (note that it must be empty for this to work)
ls	list the contents of the current working directory
ls -l	...in "long" format, which includes file permissions
ls -ld	...including permissions and information for subdirectories, instead of their contents
ls *dirname*	list the contents of directory *dirname*
chmod 700 *dirname* **chmod 700** *scr.py*	protect directory *dirname* or python script *scr.py*--- only YOU can read, write, or execute its contents.

file-related commands

cp *filename newfilename*	creates a copy of the file *filename* with the name *newfilename*
cp *f1 f2 f3 f4 ... dirname*	creates copy of files *f1 f2 f3 f4 ...* (all that you care to list) in the directory *dirname*
mv *filename newfilename*	change the name of the file *filename* to the name *newfilename*
mv *f1 f2 f3 f4 ... dirname*	move files *f1 f2 f3 f4 ...* (all that you care to list) to the directory *dirname*

rm *filename*	remove the file *filename*
chmod 600 *filename*	protect file *filename* ---only YOU can read or write it.
more *filename*	look at the contents of file *filename* on-screen, one screen at a time
cat *filename*	look at the contents of file *filename* on-screen, all at once
pico *filename*	edit file *filename* (**pico** is simplest – commands at the bottom, ctrl-o to save
vi *filename*	*what you've typed, ctrl-x to exit)*
emacs *filename*	

other commands and etc.

*	wildcard character that matches **any** 0 or more characters.
	example: **ha*s** matches **has**, **ha3s**, **haaaaaaaaaas**, etc.
?	wildcard character that matches **any** single character
	example: **ha?s** matches **hams**, **ha3s** but does NOT match **has**, **haaas**
ESC key	in several UNIX shells (including the default you get on cs-server and redwood), typing the tab key after you have started typing a file name will cause the shell to try to **complete** (fill in) the file name you have started typing, if it can. This is called **filename completion**.
grep *pattern* *	look for files in the current working directory that contain **inside** of them the pattern or letters *pattern*
diff *file1 file2*	compare the contents of *file1* and *file2*, and show any differences. If the two files are identical, nothing is returned.
history	show a list of the most recently-done commands in this UNIX session
!!	redo the last UNIX command done
!com	redo the most recent UNIX command done starting with the letters *com*
!-x	redo the UNIX command done *x* UNIX commands ago
!x	redo the UNIX command numbered *x* in the history list
up-arrow key	(the key, you don't type this verbatim!) lets you scroll through the commands in the history list (in several UNIX shells, including the cs-server and redwood default).
quota	(note: this command is useful for axe, redwood, and sorrel --- it is not available on cs-server.) lets you know how much of your disk space quota you are using --- the second column, **blocks**, shows how much you are currently using, and the third column, **quota**, shows how much you are permitted to use.