



Introduction to Shell Scripting

Globs and Booleans

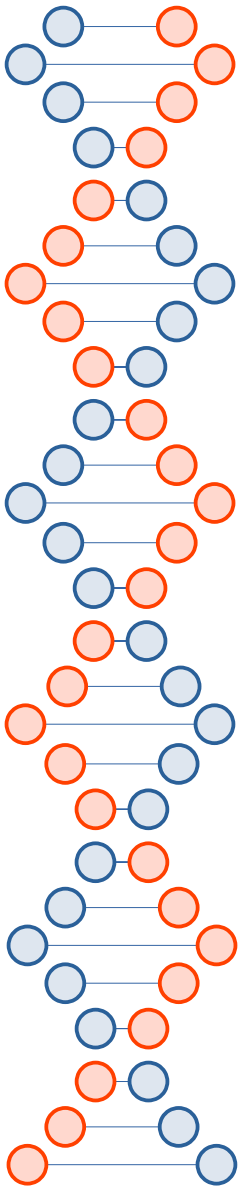
Skylar Chan - BSCI238G
UMD
March 11th, 2022

Students will be able to

- Use globs
- Use boolean statements
- Do well on the midterm!



Swoobat



Midterm Info

- The actual midterm will look like the practice midterm
- Directory manipulation (1 question)
- 2 question block (some bio scenario)
 - Pipelining stuff
 - awk/grep/sed (use regex)
 - Regex itself will not be tested
- Chmod numbers (1 Q?)
- Vocab/short answers/MC (a few)
- Glob and boolean (small)

Globbering (different from regex)

- Globbs match filenames in the current working directory
- *

 - Zero or more characters

- ?

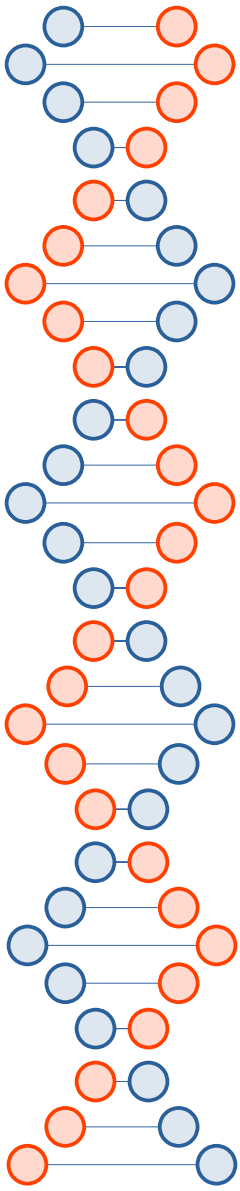
 - Any one character

- [a-z]

 - Character range/class

- Globbs do not expand in quotes
- By default if there are no matches the string is interpreted literally!

Example globs



- `ls *`
- `ls '*'`
- `less *`
- `less *.c`
- `less *.?`
- `grep b*`
- `grep b*.c`
- `ls [a-e]*`



Useful Bash options for globs

- **shopt -s nullglob**
 - Do not interpret globs as literal strings if no files match
- **shopt -s globstar**
 - Enables the double-star **, which matches files/directories recursively

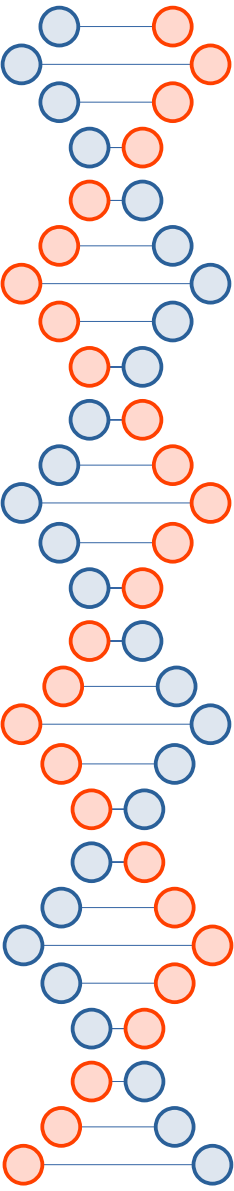
Exit codes

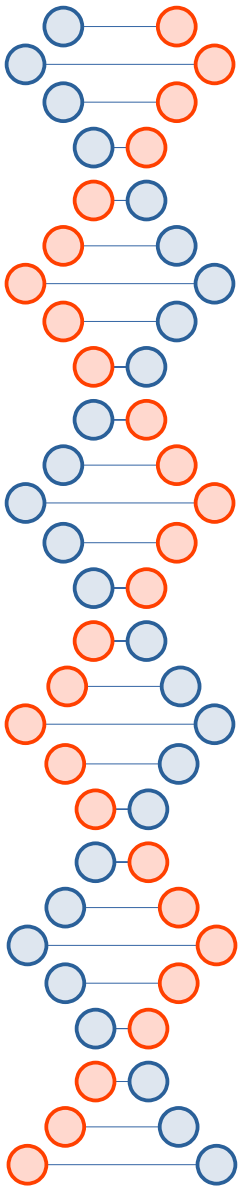
- Commands report an exit code when they are finished.
- The `$?` *variable* contains the exit code
- Can be handy to put this into your prompt
- 0 = successful exit
- >0 = something wrong happened
- Can use the exit status of command as truth value

Boolean logic

- **true** and **false** return 0 and 1
- `&&` = logical and, `||` = logical or, `!` = logical not
- Shortcircuiting – do the least work possible to determine the exit code
- `A && B`
 - If A exits with 0, run B and exit with B's exit code
 - Else exit 1
 - **B runs if A exits with 0**
- `A || B`
 - If A exits with 1, run B and exit with B's exit code
 - Else exit 0
 - **B runs if A exits with 1**
- `!A`
 - If A exits with 1, exit with 0. Else exit with 0.
- `&&` has higher precedence than `||`

Boolean logic examples

- 
- true && true
 - true && false
 - false && true
 - false && false
 - true || true
 - true || false
 - false || true
 - false || false



If/else

```
if condition1 ; then  
    A  
elif condition2 ; then  
    B  
else  
    C  
fi
```

```
if condition1  
then  
    A  
elif condition2  
then  
    B  
else C  
fi
```

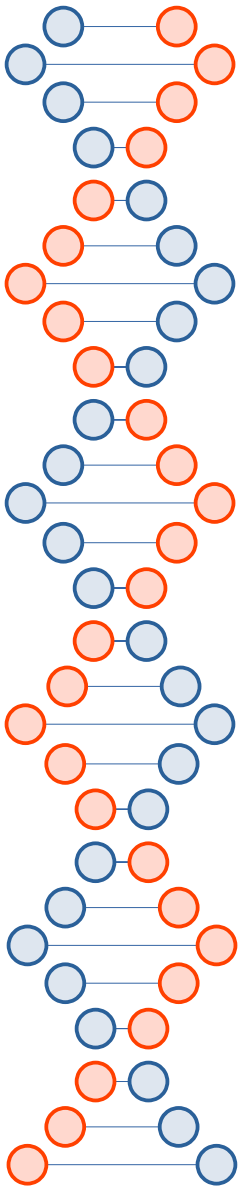
Boolean logic

- **if A ; then B ; fi equals A && B**
- **if ! A ; then B ; fi equals A || B**
- **if A && B ; then C ;fi equals A && B && C**
- **if A ; then B ; else C ; fi does NOT equals A && B || C**
- **if A || B ; then C ; fi does NOT equals A || B || C**



Printing informative messages

- `if grep -q U * ; then echo 'found U' ; fi`
- `if diff one.txt two.txt ; then echo 'files are the same' ; fi`
- `If ! diff one.txt two.txt > /dev/null ; then echo 'files are different' ; fi`



The End

Today's Terminal Toy:
[nyancat](#)