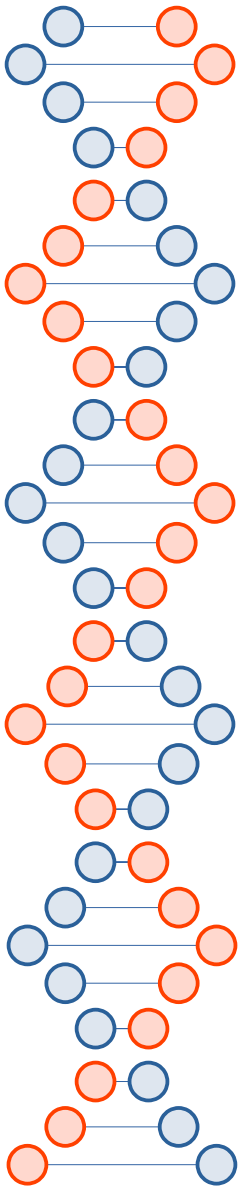


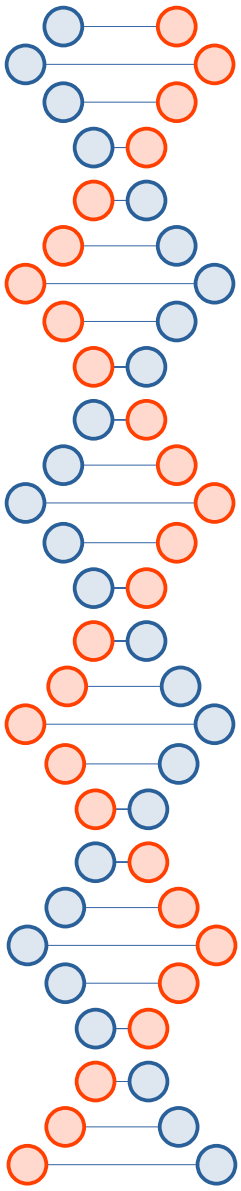
Introduction to Shell Scripting
Here doc/strings, arrays, functions

Skylar Chan - BSCI238G
UMD
April 15th, 2022

Students will be able to

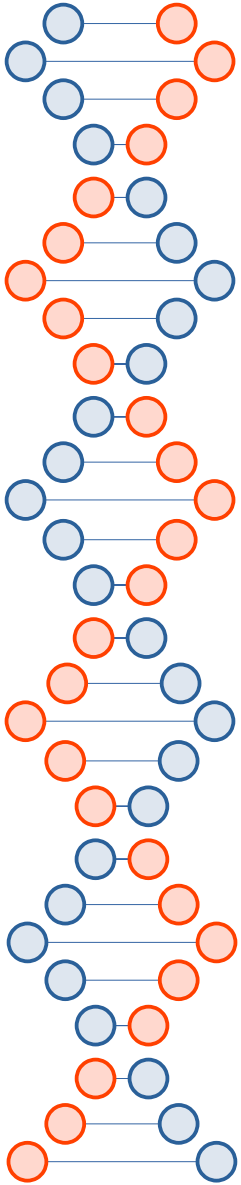
- Here doc
- Here string
- Write functions
- Use arrays

Swoobat



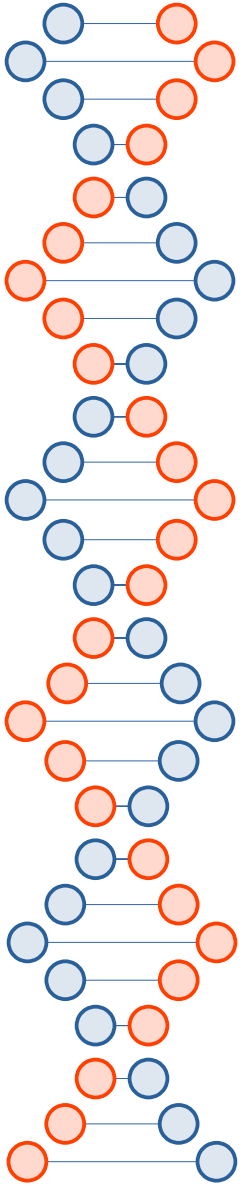
Read/while loop

- Do something per line in file
- **while read -r line ; do echo $line ; done < file**
- Or
- **some_cmd | while read -r line ; do echo $line ; done**



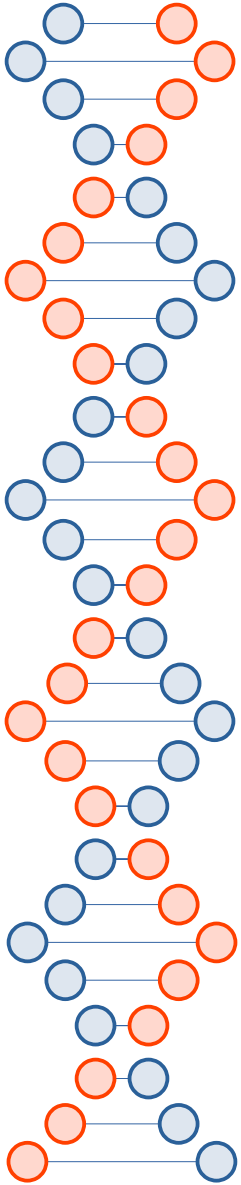
Here doc

- **cat << EOF … EOF**
- Read input from start word until end word
- This goes to standard input
- Saves time writing so many echos
- Variables are expanded inside and can get confusing



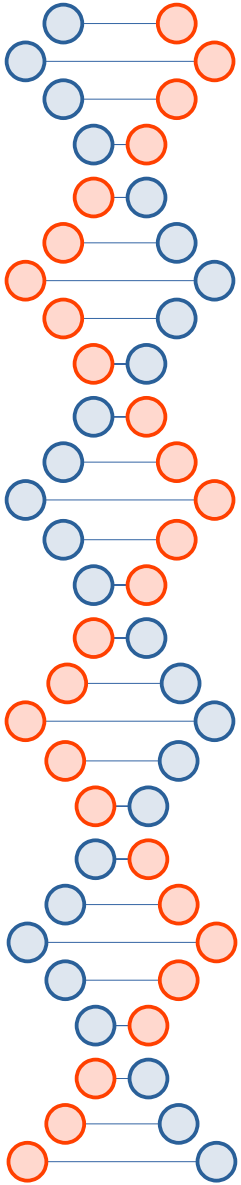
Here string

- **grep stuff <<< $1**
- Read the contents of a variable and use it as standard input
- Bashism! But a useful one



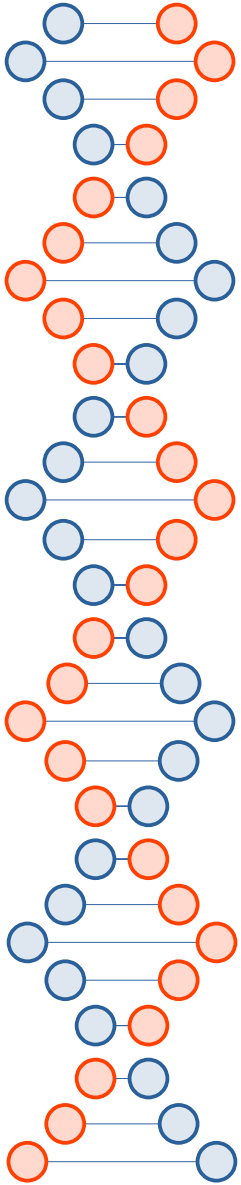
Arrays

- A list of strings and numbers
- Bash arrays are unwieldy so we'll keep it simple
- **arr=(0 1 2)**
- **echo "${arr[@]}" # right**
- **echo "$arr" # wrong**



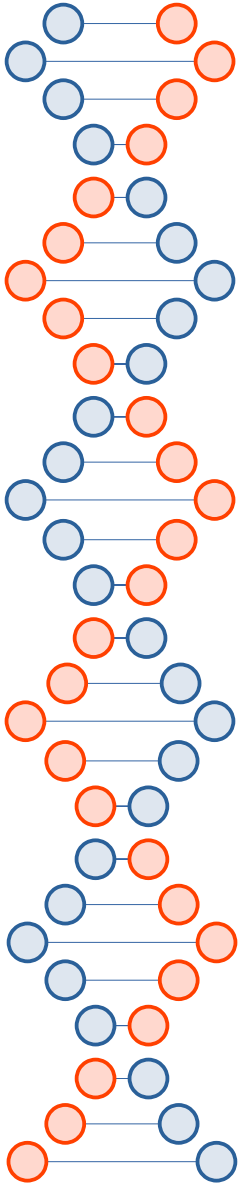
Array indexing

- Arrays are 0-indexed (the 0th element is the first one)
- **arr=(a b c)**
- **echo “${arr[0]}”**
- **echo “${arr[1]}”**
- **echo “${arr[2]}”**
- Printing out of bounds does not give an error
- **echo “${arr[3]}”**
- **echo “${arr[-1]}”**



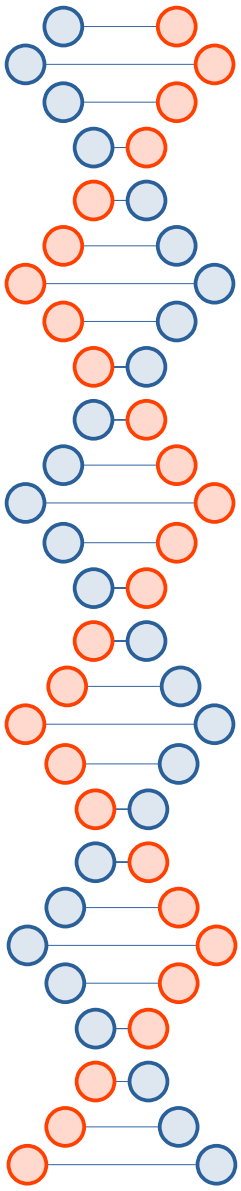
More arrays

- Append to array
- **arr+=(1)**
- Array length
- **echo “${#arr[@]}”**



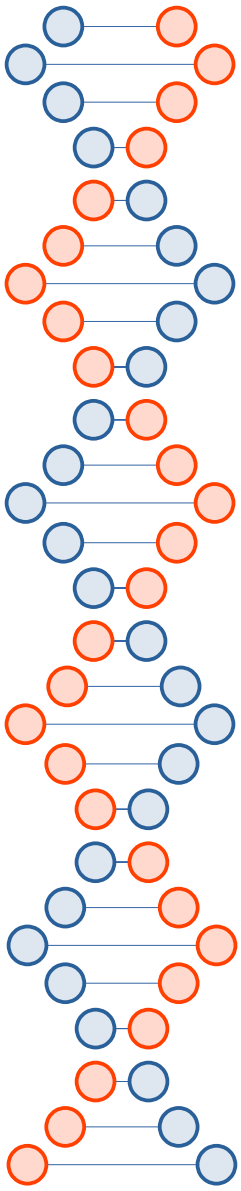
Associative arrays

- Similar to (but not the same) as hash maps
- Associate keys with values
- **declare -A assoc=([one]=1 [3]=three [key]=value)**
  - The declare part is required!
- **echo “$assoc” # wrong**
- **echo “${assoc[1]}” # wrong**
- **echo “$[assoc[one]}” # right**



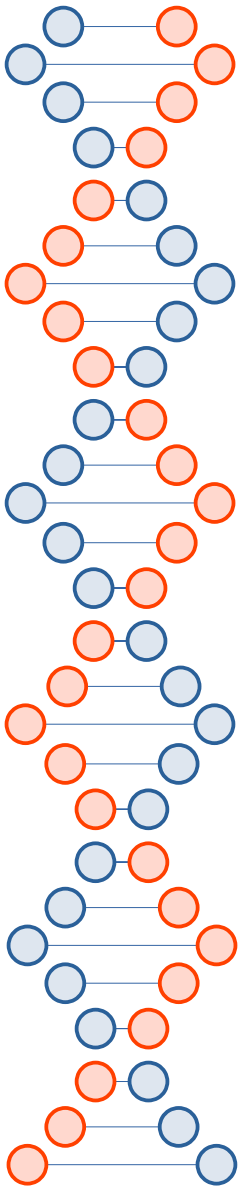
More associative arrays

- Add to associative array
- **arr[v]=k**
- Modify associative array entry
- **arr[v]=j**
- Length of associative array
- **echo “${#arr[@]}”**
- Keys of associative array
- **echo “${!arr[@]}”**
- Values of associative array
- **echo “${arr[@]}”**



Functions

- Reusable blocks of code
- A script in a script!
- Use the **local** keyword for local variables
- https://riptutorial.com/bash/example/2477/functions-with-arguments

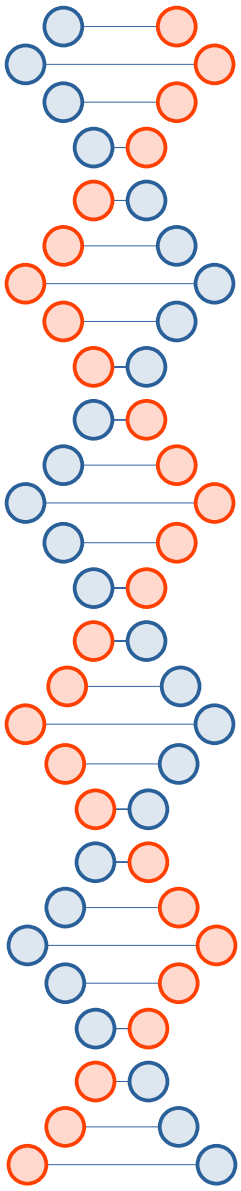


Recursion

- A function that calls itself!
- Very important topic in computer science
- It’s another form of problem solving
- What does the below function do?
- **myyes() {**

  echo “$1”

  myyes “$1”
- **}**



The End

Today’s Terminal Toy:
doge