BINF2111 - Introduction to Bioinformatics Computing

BASH 101 - Loops and conditionals



Richard Allen White III, PhD RAW Lab

Lecture 9 - Tuesday Sep 19th, 2022

Learning Objectives

- Carnegie Rule
- Review quiz and bonus
- Review assignment 3
- Bash conditionals
- Bash for loops
- Quiz 9

Carnegie rule

Carnegie Rule is a rule of thumb suggesting how much outside-of-classroom study time is required to succeed in an average higher education course in the U.S. system.

Is for every hour spent in the classroom that two or more hours of outside work required.

Carnegie rule

Carnegie Rule is a rule of thumb suggesting how much outside-of-classroom study time is required to succeed in an average higher education course in the U.S. system.

Is for every hour spent in the classroom that two or more hours of outside work required.

OUTDATED!

RAW rule of thumb for computational learning is spend quality time at the terminal, googling, and thinking problems at the terminal..

printf '#!/bin/bash\n \n# This is my first comment\n #Wookies rule' >script.sh

more script.sh

#!/bin/bash

This is my first comment # Wookies rule

Is that my correct script output?

printf '#!/bin/bash\n \n# This is my first comment\n #Wookies rule'
>script.sh

more script.sh

#!/bin/bash

This is my first comment # Wookies rule

Is that my correct script output?

printf '#!/bin/bash\n \n# This is my first comment\n #Wookies rule'
>script.sh

more script.sh

#!/bin/bash

This is my first comment # Wookies rule

Is that my correct script output? FALSE

printf '#!/bin/bash\n \n# This is my first comment\n #Wookies rule' >script.sh

more script.sh

#!/bin/bash

This is my first comment #Wookies rule

Is that my correct script output? FALSE

This file is an example of a BLANK type file?

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTT
GTTCAACTCACAGTTT
+
```

```
!"*((((***+))%%%++)(%%%%).1***
+*"))**55CCF>>>>>CCCCCC65
```

This file is an example of a BLANK type file?

+*"))**55CCF>>>>CCCCCCC65

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTT
GTTCAACTCACAGTTT
+
!"*((((***+))%%%++)(%%%%).1***
```

```
FASTQ = .fq/.fastq
```

Provide two unique commands to convert a tsv to csv? They cannot start with the same command (For example, sed/tr, grep/tr etc)

- Command 1 here:

- Command 2 here:

- Another way:

Provide two unique commands to convert a tsv to csv? They cannot start with the same command (For example, sed/tr, grep/tr etc)

- Command 1 here:
sed 's/\t/,/g' file.tsv >file.csv

- Command 2 here:

- Another way:

Provide two unique commands to convert a tsv to csv? They cannot start with the same command (For example, sed/tr, grep/tr etc)

```
- Command 1 here:
sed 's/\t/,/g' file.tsv >file.csv
```

- Command 2 here:
cat file.tsv | tr -s '\t' ',' >file.csv

- Another way:

Provide two unique commands to convert a tsv to csv? They cannot start with the same command (For example, sed/tr, grep/tr etc)

- Command 1 here:
 sed 's/\t/,/g' file.tsv >file.csv
- Command 2 here:
 cat file.tsv | tr -s '\t' ',' >file.csv
- Another way: awk -F '\t' -vOFS=, '{\$1=\$1}1'

Extract the sequence in example2.fasta that has more then one ATG?

more example2.fasta
>chr1_geneA
ATGCTAAGGCTATCTTGACAACTGACTGCCTAG
>chr1_geneB
ATGCTAAGGCTATGTTGGCAACTGACTCCCTAG
>chr1_geneC
ATGCTAAGGCTACCTTGACAACTGACTGGGTAG
>chr1_geneD
ATGAAAAGGCTATCTTGACAACTGACTCCCTAG

>chr1_geneX
ATGCTAAGGCTATCTTGATTTCTGACTTTTTAG
>chr1 geneY

ATGGGGGGCTATCTTGACAACTGACTGCGTAG

>chr1_geneZ ATGCTAAGGCTATCNNGACAACTGACTAAATAG What is a command to find multiple 'ATGs'?

- Command:

Extract the sequence in example2.fasta that has more then one ATG?

more example2.fasta >chr1 geneA ATGCTAAGGCTATCTTGACAACTGACTGCCTAG >chr1 geneB ATGCTAAGGCTATGTTGGCAACTGACTCCCTAG >chr1 geneC ATGCTAAGGCTACCTTGACAACTGACTGGGTAG >chr1 geneD ATGAAAAGGCTATCTTGACAACTGACTCCCTAG >chr1 geneX ATGCTAAGGCTATCTTGATTTCTGACTTTTTAG >chr1 geneY ATGGGGGGCTATCTTGACAACTGACTGCGTAG

ATGCTAAGGCTATCNNGACAACTGACTAAATAG

>chr1 geneZ

What is a command to find multiple 'ATGs'?

- Command:

grep "ATG" example2.fasta --color

Extract the sequence in example2.fasta that has more then one ATG?

more example2.fasta >chr1 geneA

ATGCTAAGGCTATCTTGACAACTGACTGCCTAG

>chr1_geneB

ATGCTAAGGCTATGTTGGCAACTGACTCCCTAG

>chr1 geneC

ATGCTAAGGCTACCTTGACAACTGACTGGGTAG

>chr1 geneD

ATGAAAAGGCTATCTTGACAACTGACTCCCTAG

>chr1 geneX

ATGCTAAGGCTATCTTGATTTCTGACTTTTTAG

>chr1 geneY

ATGGGGGGCTATCTTGACAACTGACTGCGTAG

>chr1 geneZ

ATGCTAAGGCTATCNNGACAACTGACTAAATAG

How do I grab the one with two ATGs?

- Command:

Another way?

Extract the sequence in example2.fasta that has more then one ATG?

more example2.fasta >chr1 geneA ATGCTAAGGCTATCTTGACAACTGACTGCCTAG >chr1 geneB **ATGCTAAGGCTATGTTGGCAACTGACTCCCTAG** >chr1 geneC ATGCTAAGGCTACCTTGACAACTGACTGGGTAG >chr1 geneD ATGAAAAGGCTATCTTGACAACTGACTCCCTAG >chr1 geneX ATGCTAAGGCTATCTTGATTTCTGACTTTTTAG >chr1 geneY ATGGGGGGCTATCTTGACAACTGACTGCGTAG >chr1 geneZ **ATGCTAAGGCTATCNNGACAACTGACTAAATAG**

How do I grab the one with two ATGs?

Command:grep 'ATG.*ATG' example2.fasta>extracted.fasta

Another way?

Extract the sequence in example2.fasta that has more then one ATG?

more example2.fasta >chr1 geneA ATGCTAAGGCTATCTTGACAACTGACTGCCTAG >chr1 geneB **ATGCTAAGGCTATGTTGGCAACTGACTCCCTAG** >chr1 geneC ATGCTAAGGCTACCTTGACAACTGACTGGGTAG >chr1 geneD ATGAAAAGGCTATCTTGACAACTGACTCCCTAG >chr1 geneX **ATGCTAAGGCTATCTTGATTTCTGACTTTTTAG** >chr1 geneY ATGGGGGGCTATCTTGACAACTGACTGCGTAG >chr1 geneZ ATGCTAAGGCTATCNNGACAACTGACTAAATAG

How do I grab the one with two ATGs?

- Command: grep 'ATG.*ATG' example2.fasta >extracted.fasta

Another way? grep '\(.*ATG\)\{2\}' example2.fasta >extracted.fasta

- Write a bash script that states 150 kg at 178 cm is overweight?

```
Var1 = ?
Var2 = ?
```

- Write a bash script that states 150 kg at 178 cm is overweight?

```
Var1 = weight
Var2 = height
```

What are these variables?

- Write a bash script that states 150 kg at 178 cm is overweight?

```
Var1 = weight
Var2 = height
```

What are these variables?

Integer variables!! Whole numbers 1,2,3 etc

Inside a bash script

```
1 #!/bin/bash ← hashbang
 3 # This script prints a message about your weight if you give it
  VOUL
 5 # weight in kilos and height in centimeters.
7 weight="$1" ← Var1
8 height="$2" ← Var2
9 idealweight=$[$height - 110]
10 if [ $weight -le $idealweight ] ; then
11 echo "You should eat a bit more food."
12 else
13 echo "You should eat a bit less food."
14 fi
```

Inside a bash script

Inside a bash script

bash -x weight.sh 130 178 >output.txt

Lab Q4

```
Executable Script Var1 output
bash -x count_fasta.sh example2.fasta >out.txt
```

1 #!/bin/bash

2

3 input=\$1

What is \$1 in this script?

Lab Q4

Executable Script Var1 output

bash -x count_fasta.sh example2.fasta >out.txt

1 #!/bin/bash

2

3 input=\$1

What is \$1 in this script? **EXAMPLE2.fasta**

BASH Variables (By content)

Temporary stores of information

In this respect, variables come in 4 types:

- String variables:
- Integer variables:
- Constant variables:
- Array variables:

BASH Variables (By content)

Temporary stores of information

In this respect, variables come in 4 types:

- String variables: Dog, Cat, Bill, Dave
- Integer variables: 1, 2, 101, 2021, whole numbers
- Constant variables: Pi = 3.14 etc
- Array variables: variable containing multiple values. Any variable may be used as an array. There is no maximum limit to the size

BASH Reserved words

! - Pipelines]] - Conditional Constructs } - Command Grouping **break -** Looping Constructs case - Conditional Constructs **continue** - Looping Constructs do - Looping Constructs done - Looping Constructs elif - Conditional Constructs else - Conditional Constructs esac - Conditional Constructs Conditional Constructs for - Looping Constructs **function - Shell Functions**

if - Conditional Constructs
 in - Conditional Constructs
 select - Conditional Constructs
 then - Conditional Constructs
 time - Pipelines
 until - Looping Constructs
 while - Looping Constructs

Similar from UNIX

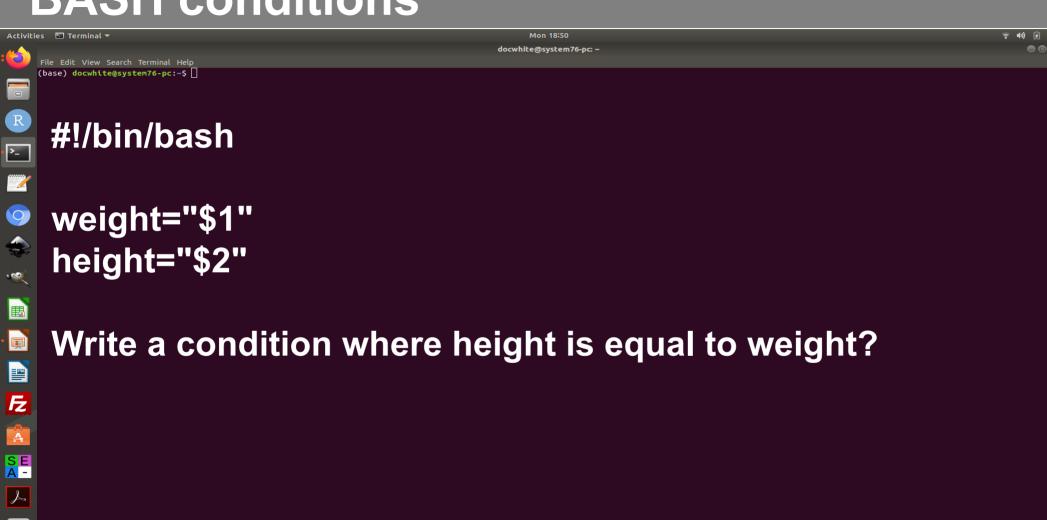
& , | , > , < , ! , = # , \$, (,) , ; , {, } , [,] , \

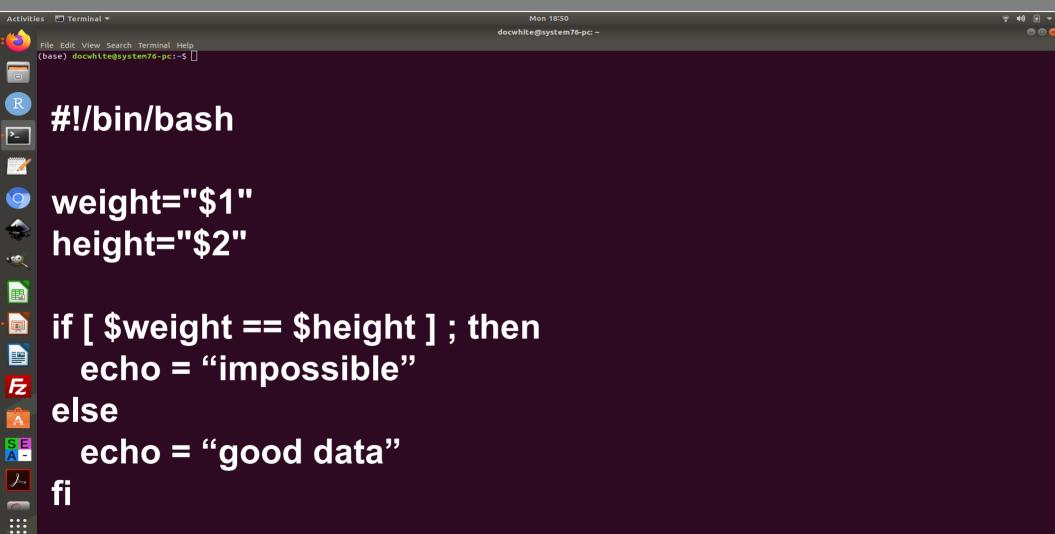
https://www.gnu.org/software/bash/manual/html_node/Reserved-Word-Index.html

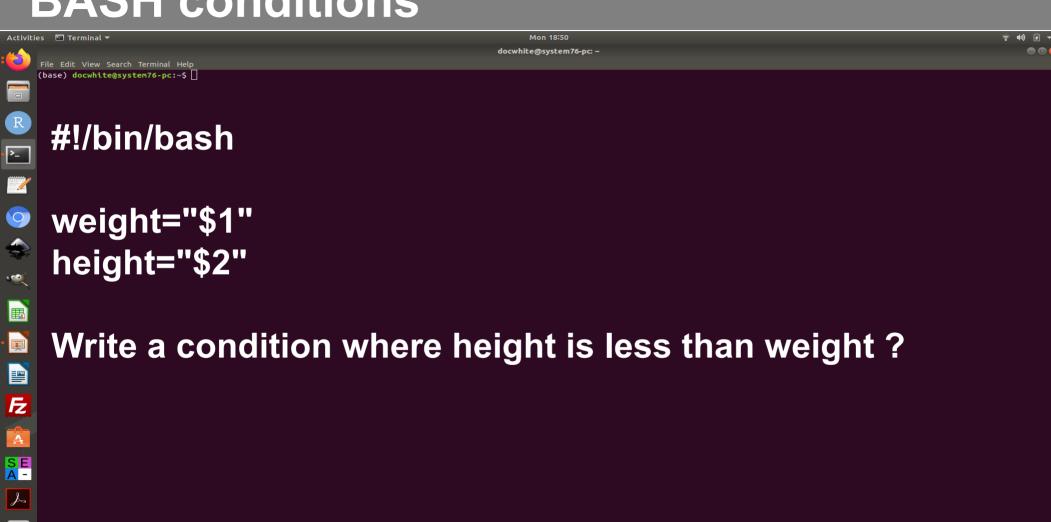
BASH Conditionals (conditions)

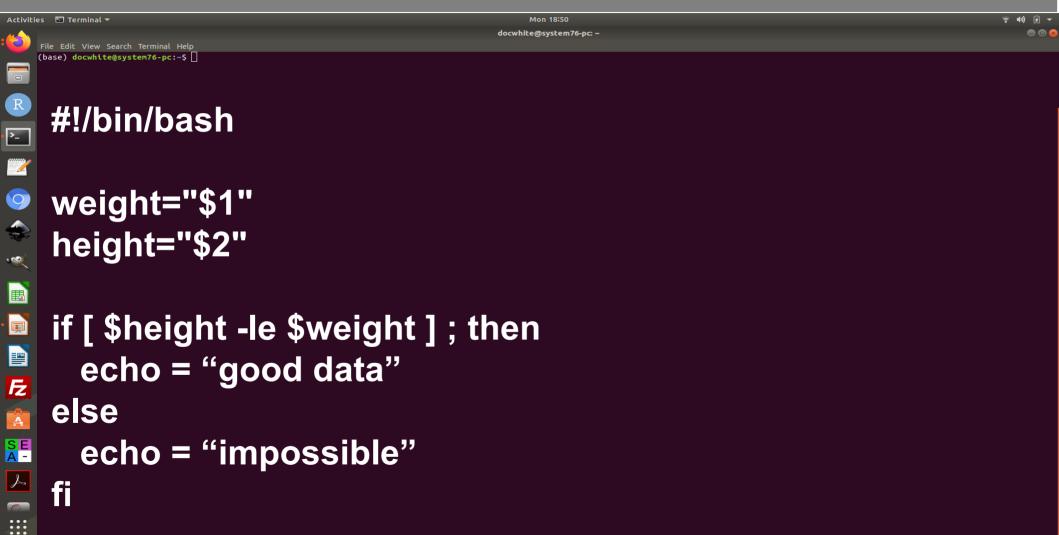
[[-z STRING]] Empty string

```
[[ -n STRING ]] Not empty string
[[ STRING == STRING ]]
                            Equal
[[ STRING != STRING ]]
                            Not Equal
[[ NUM -eq NUM ]]
                 Equal
[[ NUM -ne NUM ]]
                 Not equal
[[ NUM -It NUM ]]
                Less than
[[ NUM -le NUM ]]
                     Less than or equal
[[ NUM -gt NUM ]]
                     Greater than
[[ NUM -ge NUM ]]
                     Greater than or equal
[[ STRING =~ STRING ]]
                            Regexp
((NUM < NUM))
                     Numeric conditions
[[ -o noclobber ]]
                     If OPTIONNAME is enabled
[[ ! EXPR ]]
              Not
[[ X && Y ]]
           And
[[X||Y]]
              Or
```







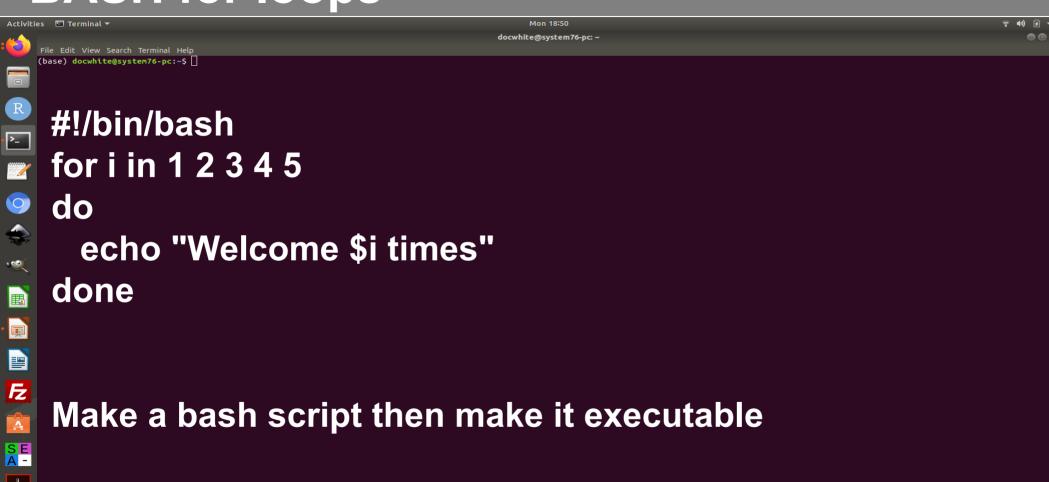


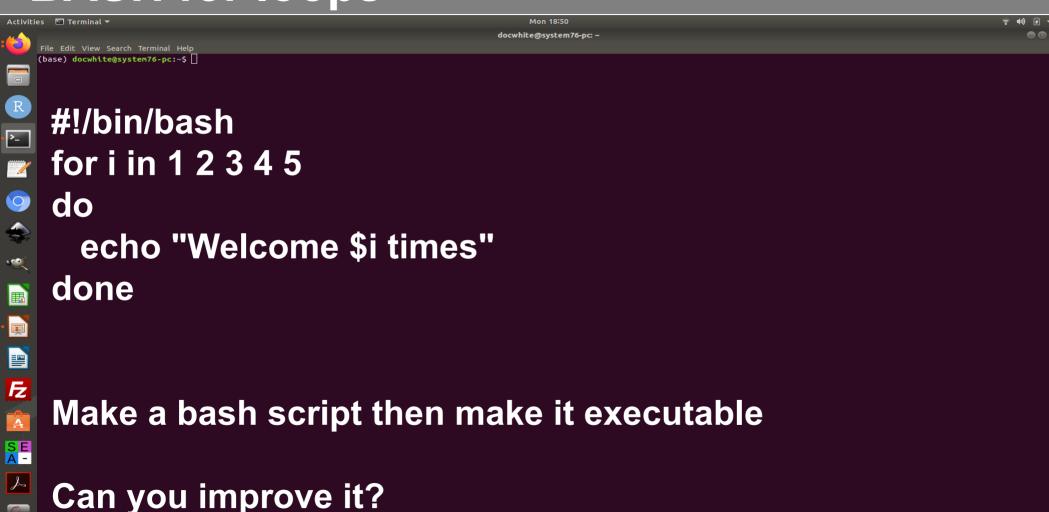
BASH Conditionals (file conditions)

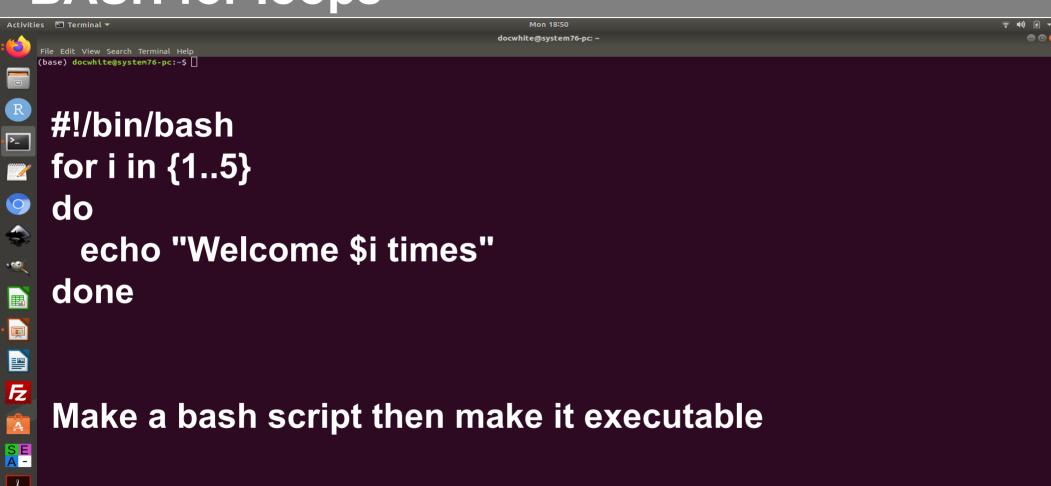
```
[[ -e FILE ]]
                Exists
[[ -r FILE ]]
                     Readable
[[ -h FILE ]]
                Symlink
[[ -d FILE ]]
                Directory
[[ -w FILE ]]
                Writable
[[ -s FILE ]]
                Size is > 0 bytes
[[ -f FILE ]]
                     File
[[ -x FILE ]]
                Executable
[[ FILE1 -nt FILE2 ]]
                      1 is more recent than 2
[[ FILE1 -ot FILE2 ]] 2 is more recent than 1
[[ FILE1 -ef FILE2 ]] Same files
```

bash -x weight.sh 130 178

for i in file.*;do command \$i done

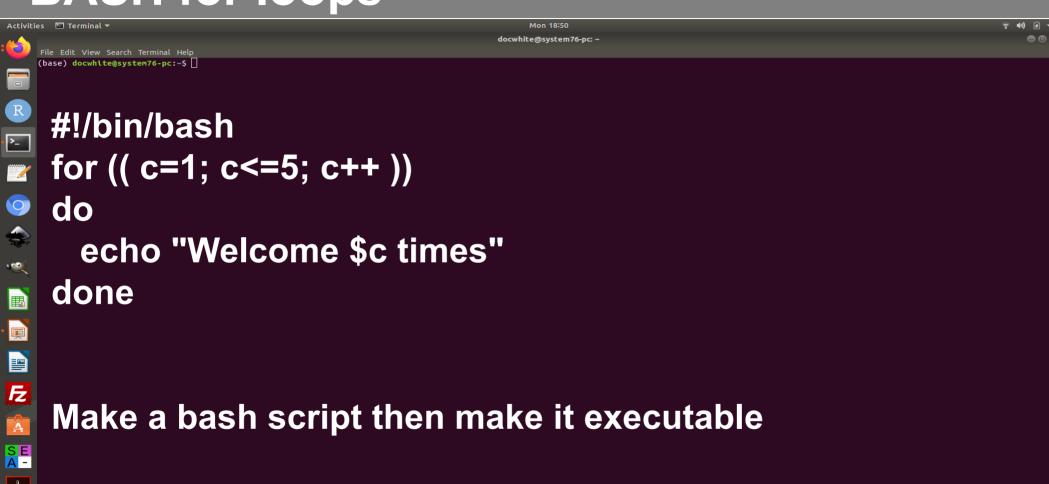






BASH - for loop (C-style)

```
for ((i = 0; i < 100; i++)); do
  command $i
  done</pre>
```



- On canvas now

- Use grep to convert sam file to a fastq file?