BINF2111 – Introduction to Bioinformatics Computing

BASH 101 – Loops and conditionals

Richard Allen White III, PhD
RAW Lab

Lecture 9 – Tuesday Sep 20th, 2021

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OUTDATED!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..

Bonus 5 (option 4)

In the doppelganger_names.txt count how many times the name ‘chi’ is left to the name ‘bill’

Using grep only command:

```
grep -oE 'chi'$'\t''bill' doppelganger_names.txt | wc -l
```

Using grep with printf command:

```
grep -oE "chi$(printf '\t')bill" doppelganger_names.txt | wc -l
```

Only awk:

```
awk '/chi\tbill/' doppelganger_names.txt | wc -l
```

Another grep way:

Bonus 5 (option 4)

In the doppelganger_names.txt count how many times the name ‘chi’ is left to the name ‘bill’

Using grep only command:

```
grep -oE 'chi'$'\t''bill' doppelganger_names.txt | wc -l
```

Using grep with printf command:

```
grep -oE "chi$(printf '\t')bill" doppelganger_names.txt | wc -l
```

Only awk:

```
awk '/chi\tbill/' doppelganger_names.txt | wc -l
```

Another grep way:

`grep -oE  ‘chi.bill’ doppelganger_names.txt | wc -l`

Quiz 8 answers

```
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?

T or F

Quiz 8 answers

```
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?

T or F

Quiz 8 answers

```
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
```

Quiz 8 answers

This file is an example of a BLANK type file?

```
>CP001157.1_1 # 101
MSVELWQQCVELLRDELPAQQFNTWIRPLQVEADGDEL*
>CP001157.1_2 # 1566
MKFPDYERVLPRGGDKKVLGDRQLLREAFSRTAILSNEK*
>CP001157.1_3 # 2688
MSLGRVTVTAVRNLHPVTLNPSPRINILYGPNGSGKTSLL*
```

Quiz 8 answers

This file is an example of a BLANK type file?

```
>CP001157.1_1 # 101
MSVELWQQCVELLRDELPAQQFNTWIRPLQVEADGDEL*
>CP001157.1_2 # 1566
MKFPDYERVLPRGGDKKVLGDRQLLREAFSRTAILSNEK*
>CP001157.1_3 # 2688
MSLGRVTVTAVRNLHPVTLNPSPRINILYGPNGSGKTSLL*
```

FAA (FASTA AMINO ACID FILE)

Bonus 7

- Write a BASH code to iterate through the Fruit array?

Fruits=('Apple' 'Banana' 'Orange', ‘Mango’)

ENJOY!

Bonus 7

- Write a BASH code to iterate through the Fruit array?

Fruits=('Apple' 'Banana' 'Orange', ‘Mango’)

```
for i in "${Fruits[@]}"; do
  echo $i
done
```

Assignment 3 lab (most missed)

(Q5) 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:

Assignment 3 lab (most missed)

(Q5) 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:

Assignment 3 lab (most missed)

(Q5) 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:

Assignment 3 lab (most missed)

(Q5) 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'
```

Assignment 3 lab (most missed)

(Q6) Using any command you want convert whitespace into tabs?

- Command 1 here:

```
cat file.txt | tr -s '[:blank:]' '\t' >fixed.txt
```

- Command 2 here:

```
sed 's/[[:blank:]]\+/\t/g' file.txt > fixed.txt
```

- Another way:

```
awk -v OFS="\t" '$1=$1' file.txt > fixed.txt
```

Assignment 3 lab (most missed)

(Q10) 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
ATGGGGGGGCTATCTTGACAACTGACTGCGTAG
>chr1_geneZ
ATGCTAAGGCTATCNNGACAACTGACTAAATAG
```

What is a command to find multiple ‘ATGs’?

- Command:

Assignment 3 lab (most missed)

(Q10) 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:

```
grep "ATG" example2.fasta --color
```

Assignment 3 lab (most missed)

(Q10) 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?

Assignment 3 lab (most missed)

(Q10) 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
```

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
fi	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

& , | , > , < , ! , =

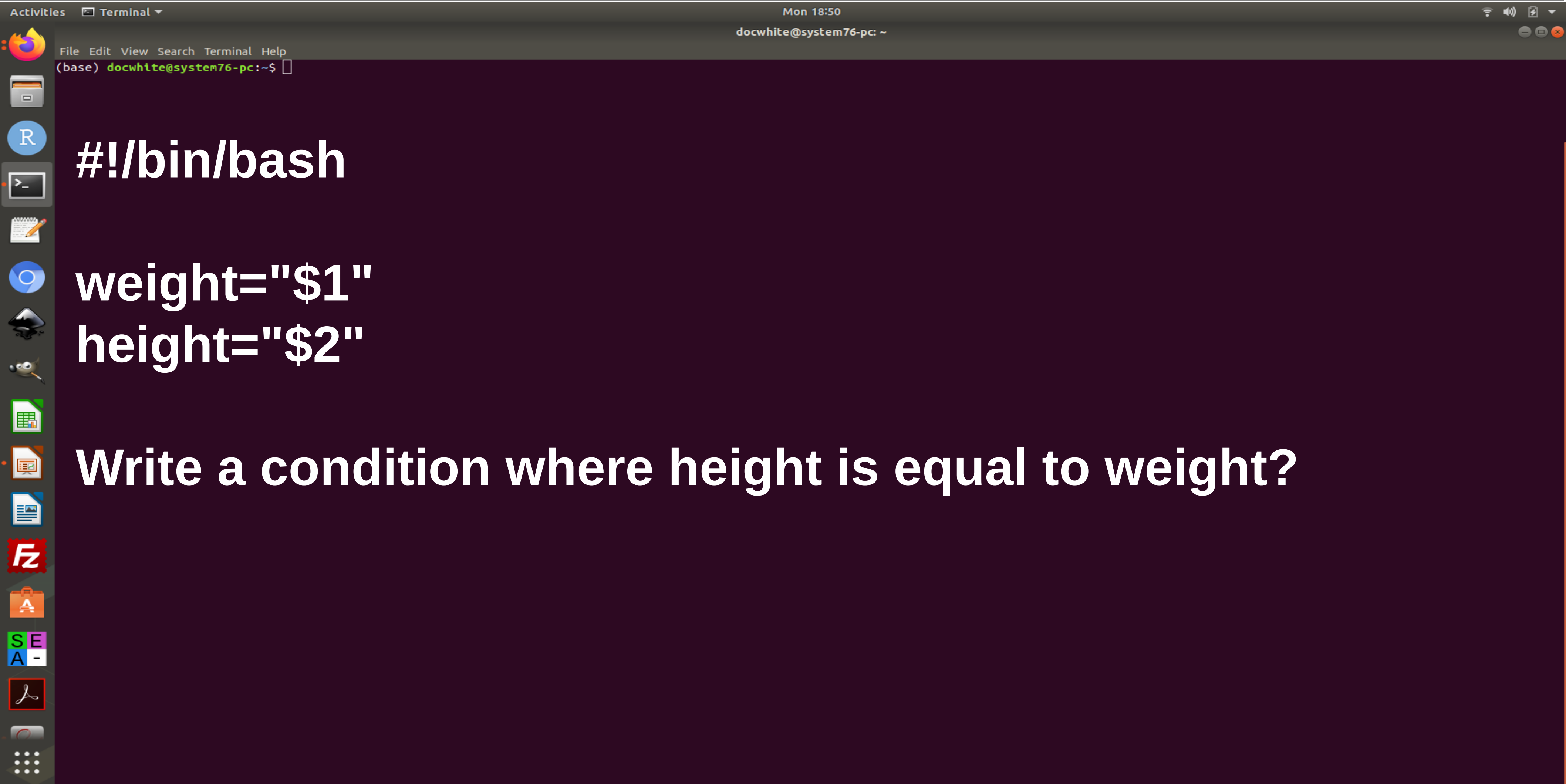
, $, (,) , ; , {, } , [,] , \

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 -lt 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 conditions



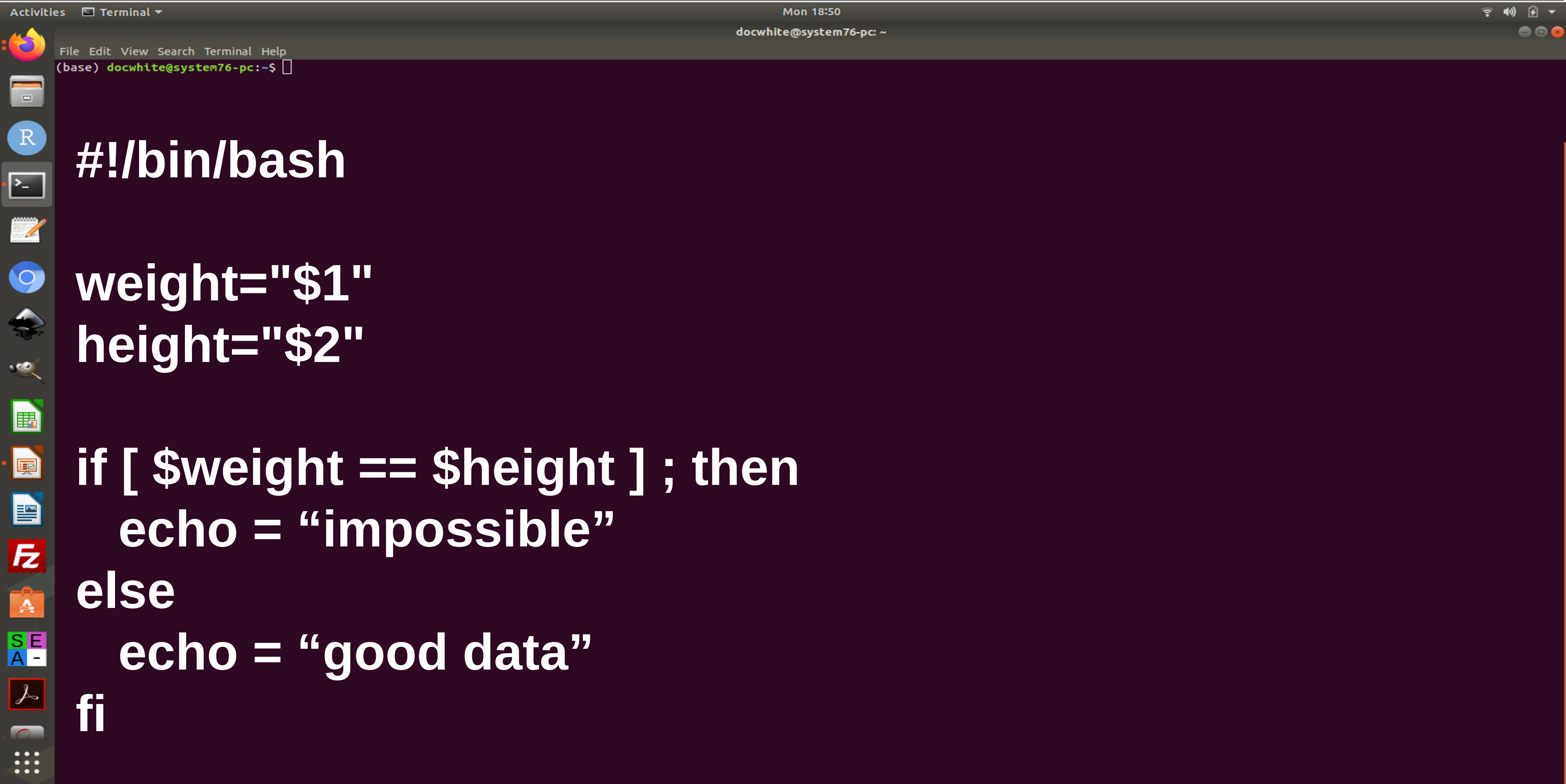
```
(base) docwhite@system76-pc:~$
```

```
#!/bin/bash

weight="$1"
height="$2"
```

Write a condition where height is equal to weight?

BASH conditions



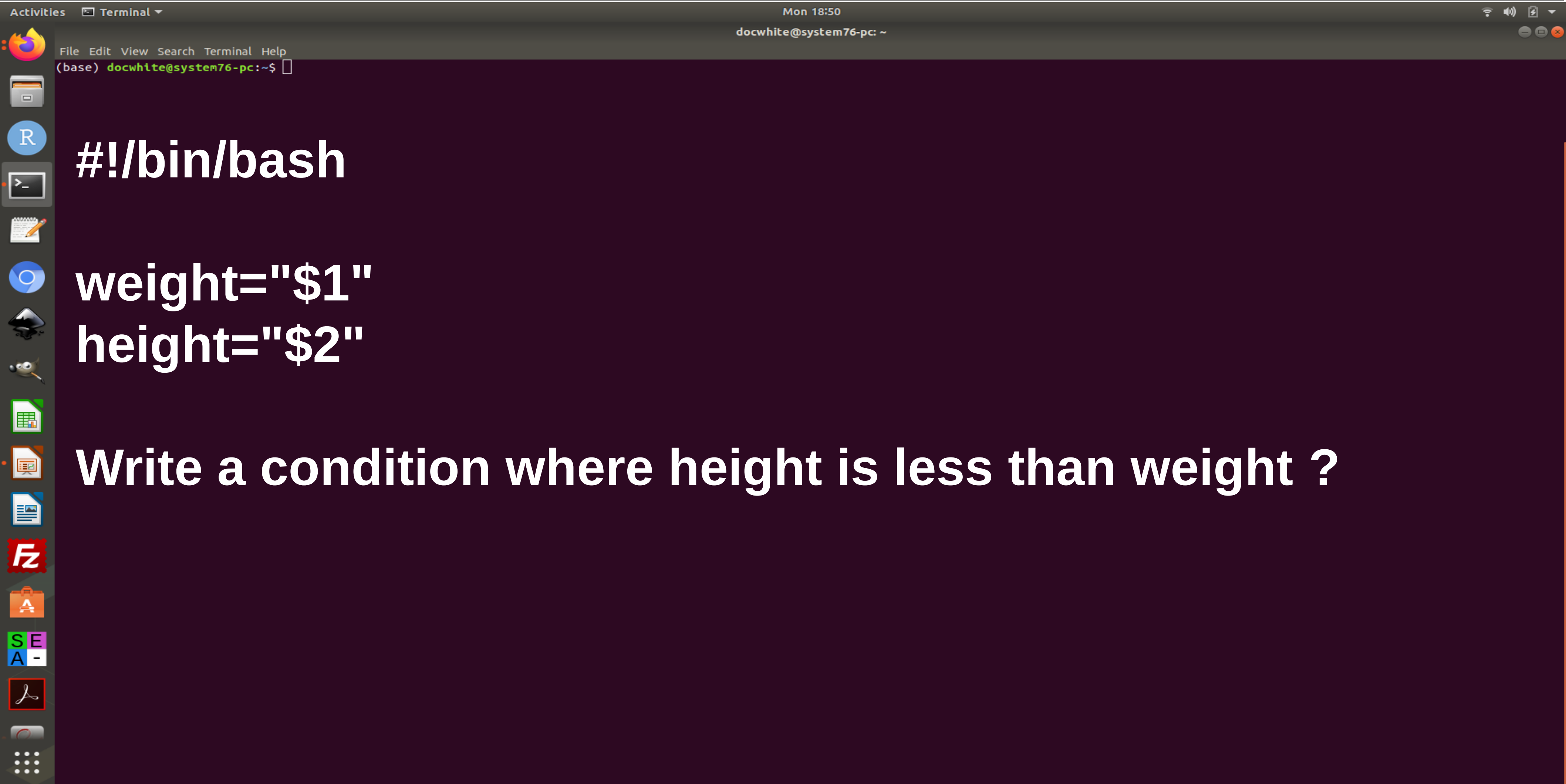
```
(base) docwhite@system76-pc:~$
```

```
#!/bin/bash

weight="$1"
height="$2"

if [ $weight == $height ] ; then
   echo = “impossible”
else
   echo = “good data”
fi
```

BASH conditions



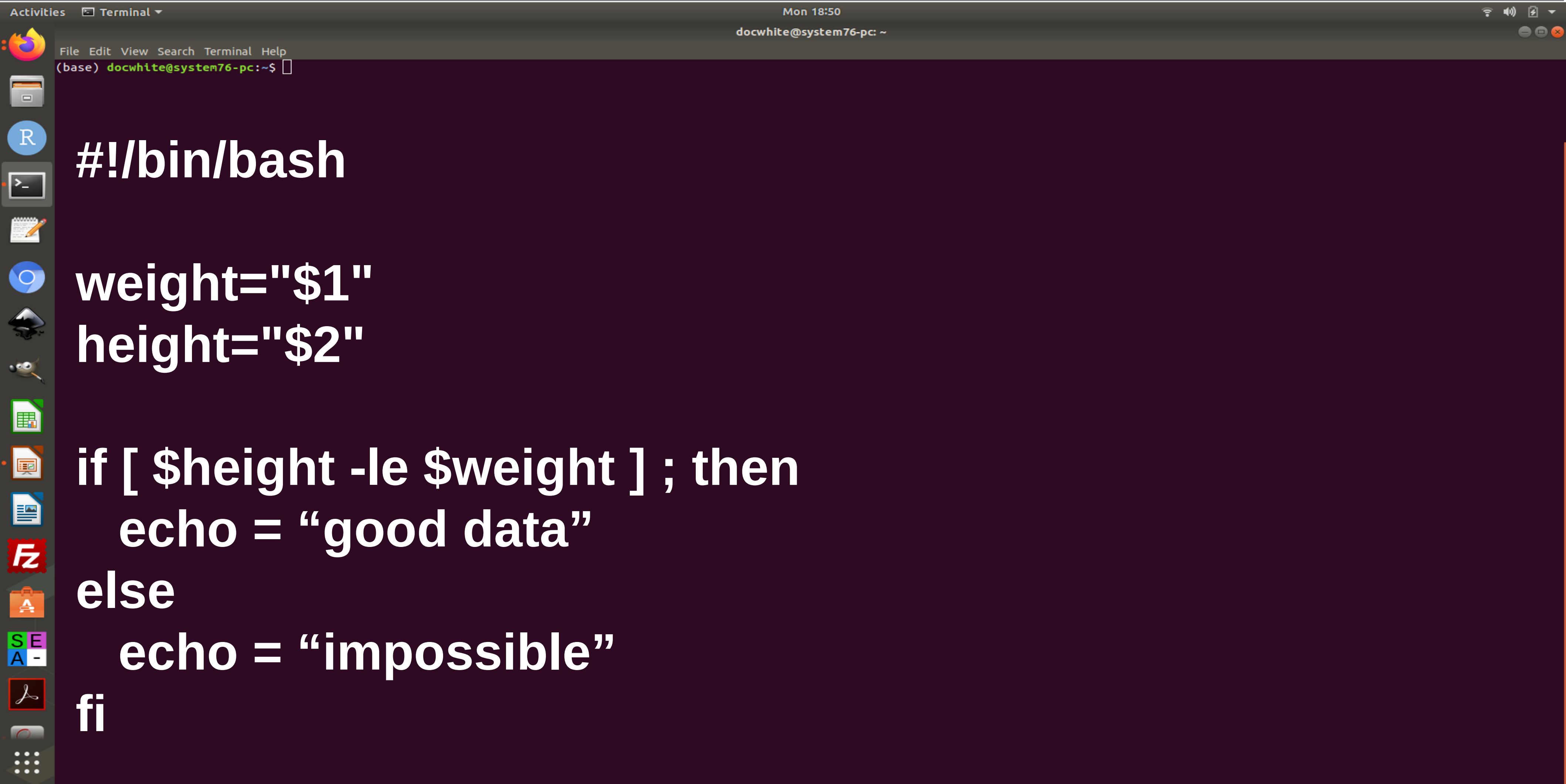
```
(base) docwhite@system76-pc:~$
```

```
#!/bin/bash

weight="$1"
height="$2"
```

Write a condition where height is less than weight ?

BASH conditions



```
(base) docwhite@system76-pc:~$
```

```
#!/bin/bash

weight="$1"
height="$2"

if [ $height -le $weight ] ; then
   echo = “good data”
else
   echo = “impossible”
fi
```

BASH Conditionals (file conditions)

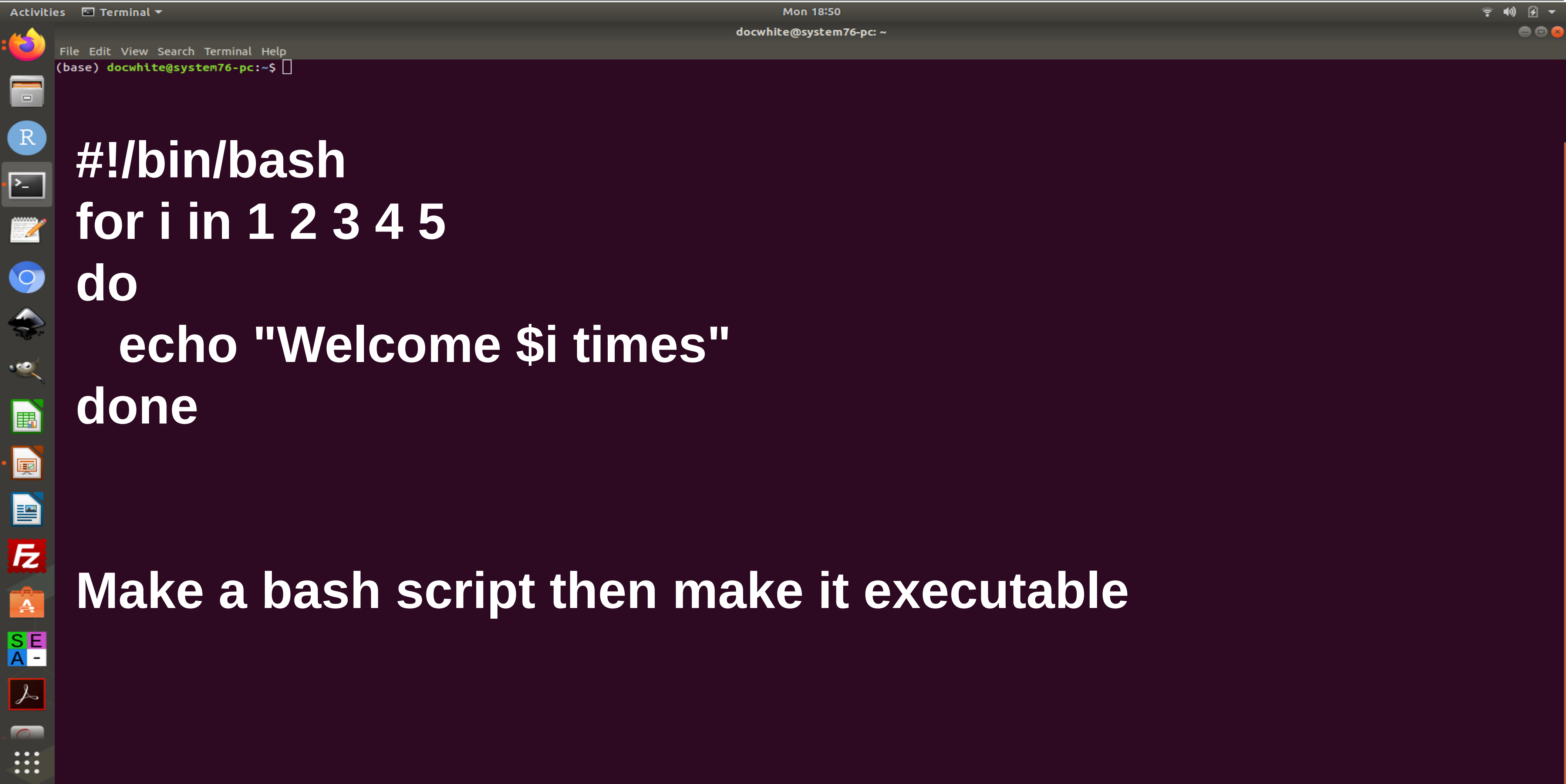
[[-e FILE]]	Exists
[[-r FILE]]	Readable
[[-h FILE]]	Symlink
[[-d FILE]]	Directory
[[-w FILE]]	Writable
[[-s FILE]]	Size is > 0 bytes
[[-f FILE]]	File
[[-x FILE]]	Executable
[[FILE1 -nt FILE2]]	1 is more recent than 2
[[FILE1 -ot FILE2]]	2 is more recent than 1
[[FILE1 -ef FILE2]]	Same files

```
bash -x weight.sh 130 178
```

BASH - for loop

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

BASH for loops

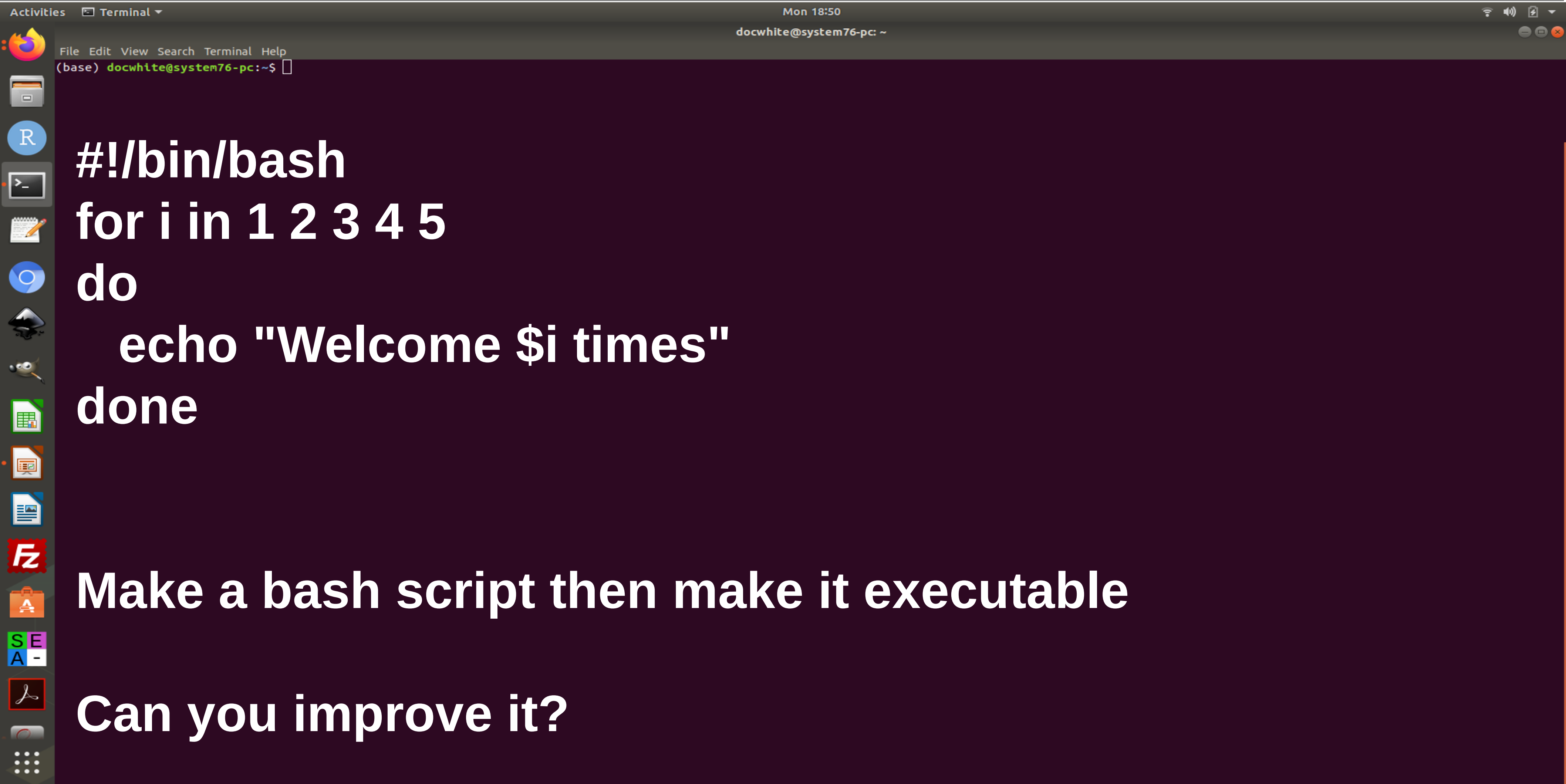


```
(base) docwhite@system76-pc:~$
```

```
#!/bin/bash
for i in 1 2 3 4 5
do
   echo "Welcome $i times"
done
```

Make a bash script then make it executable

BASH for loops



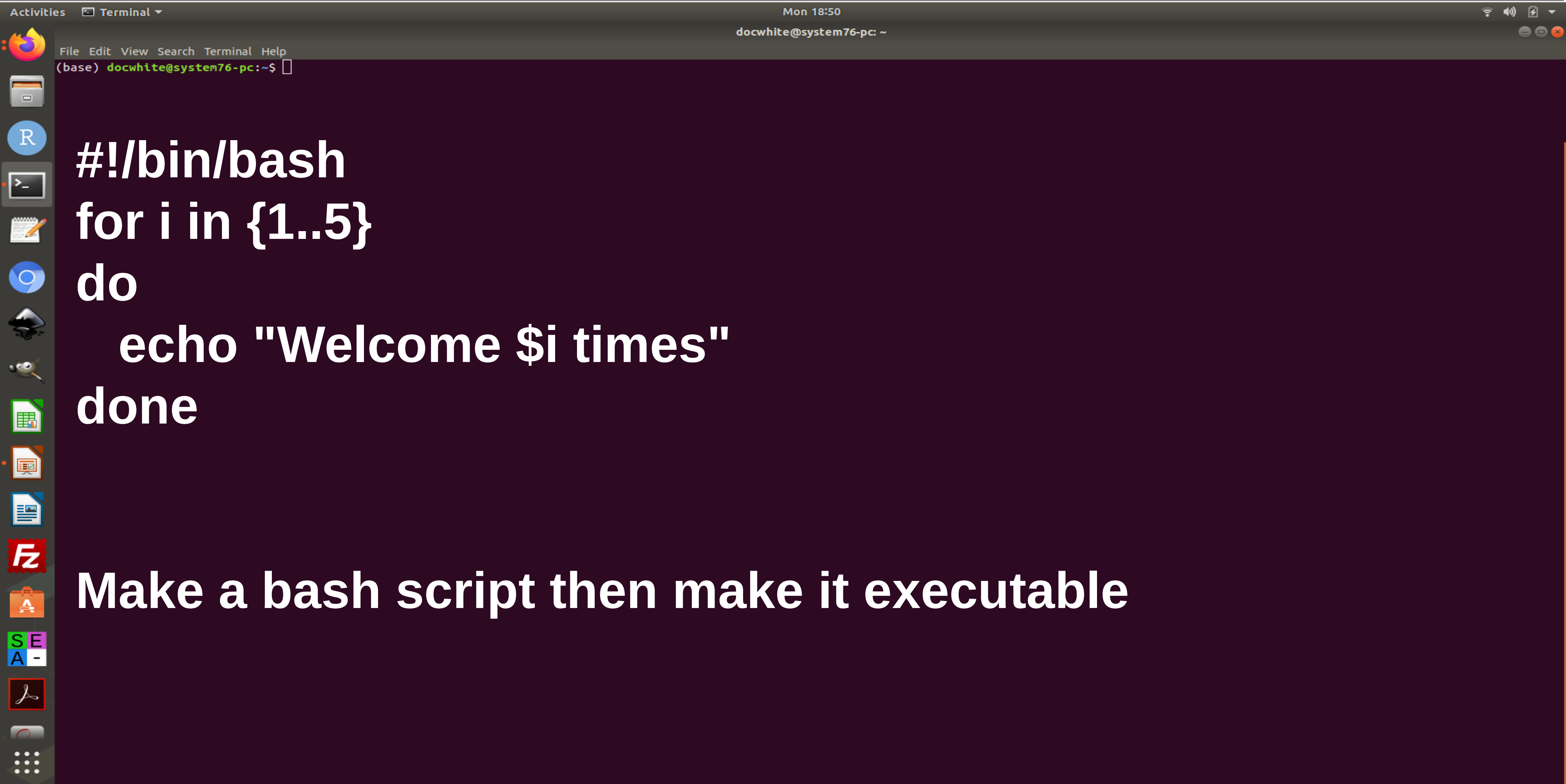
```
(base) docwhite@system76-pc:~$
```

```
#!/bin/bash
for i in 1 2 3 4 5
do
   echo "Welcome $i times"
done
```

Make a bash script then make it executable

Can you improve it?

BASH for loops



```
(base) docwhite@system76-pc:~$
```

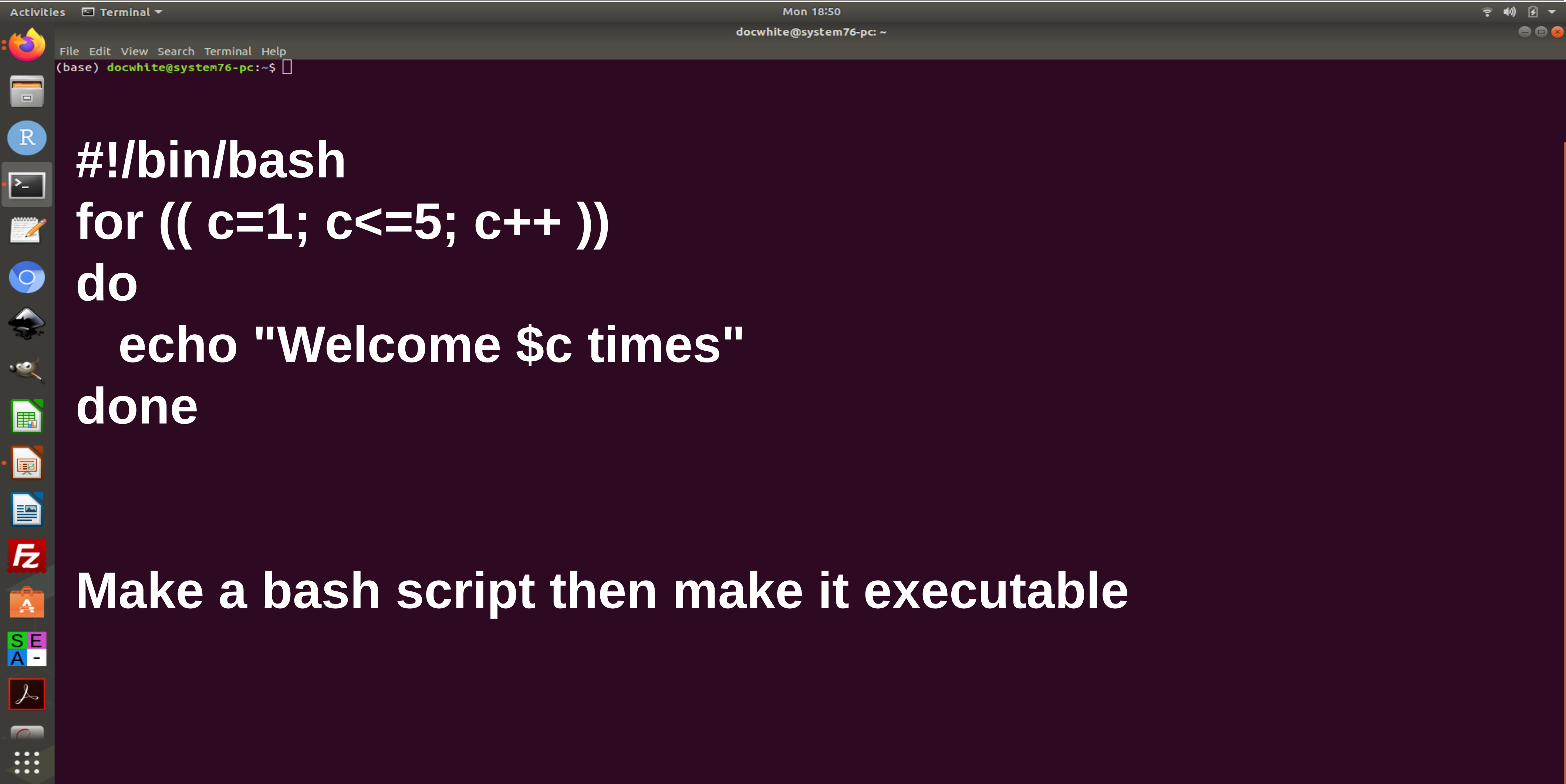
```
#!/bin/bash
for i in {1..5}
do
   echo "Welcome $i times"
done
```

Make a bash script then make it executable

BASH - for loop (C-style)

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

BASH for loops



```
(base) docwhite@system76-pc:~$
```

```
#!/bin/bash
for (( c=1; c<=5; c++ ))
do
   echo "Welcome $c times"
done
```

Make a bash script then make it executable

Quiz 9

- On canvas now

Bonus 8

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