Introduction to Bash

Danielle Denisko

MBP Tech Talk

October 12, 2018

Template from: www.overleaf.com

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Outline

Introduction
- What is the shell?
- When and how to use Bash

Navigation + Editing files
- The file system
- Navigation
- Absolute vs. relative file paths
- Creating files
- Output redirection
- Expansion
- Quoting

Sed (intro)

Exercises

Other topics

Introduction

"At a high level, computers do four things:

- run programs
- store data
- communicate with each other
- interact with us"

We can interact with computers through:

- Graphical user interface (GUI): windows, icons and pointers
- Command-line interface (CLI): shell or terminal

Adapted from: Introducing the shell, Software Carpentry

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

What is the shell?

The shell is a program that primarily runs other programs by passing keyboard commands to the operating system.

Bash is the most popular shell program. It stands for *Bourne-again shell*.

From a GUI, you open a terminal emulator (often just referred to as "terminal") to interact with the shell.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

When could you use Bash?

- to automate repetitive tasks
- to pipe output of one command to the next
- to parallelize tasks
- to perform more complex tasks than GUI
- to use tools (especially in bioinformatics) that don't implement GUIs
- to interact with remote systems (i.e. compute clusters)

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Basics of using Bash

- write commands at prompt
- usually, start with program followed by flags and then argument(s)... `command -options arguments`
- capitalization and spacing matters!

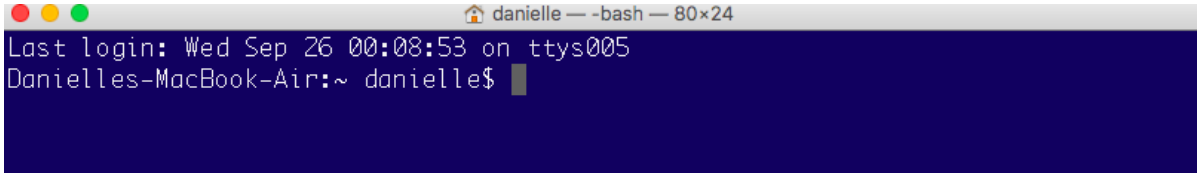


Figure 1: An example of a prompt upon launching a terminal.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Basics of using Bash

Try it!

- open a terminal
- try some simple commands (press enter after typing each one at the prompt):
  - `date`
  - `cal`
  - `ls`
  - `pwd`
- now, try the following keyboard shortcuts:

Ctrl-A (cursor start of line)	up/down arrows (history)
Ctrl-E (cursor end of line)	left/right arrows (previous/
Ctrl-R (search history)	next character)
Ctrl-C (interrupt)	Tab (auto-completion)

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

File system tree

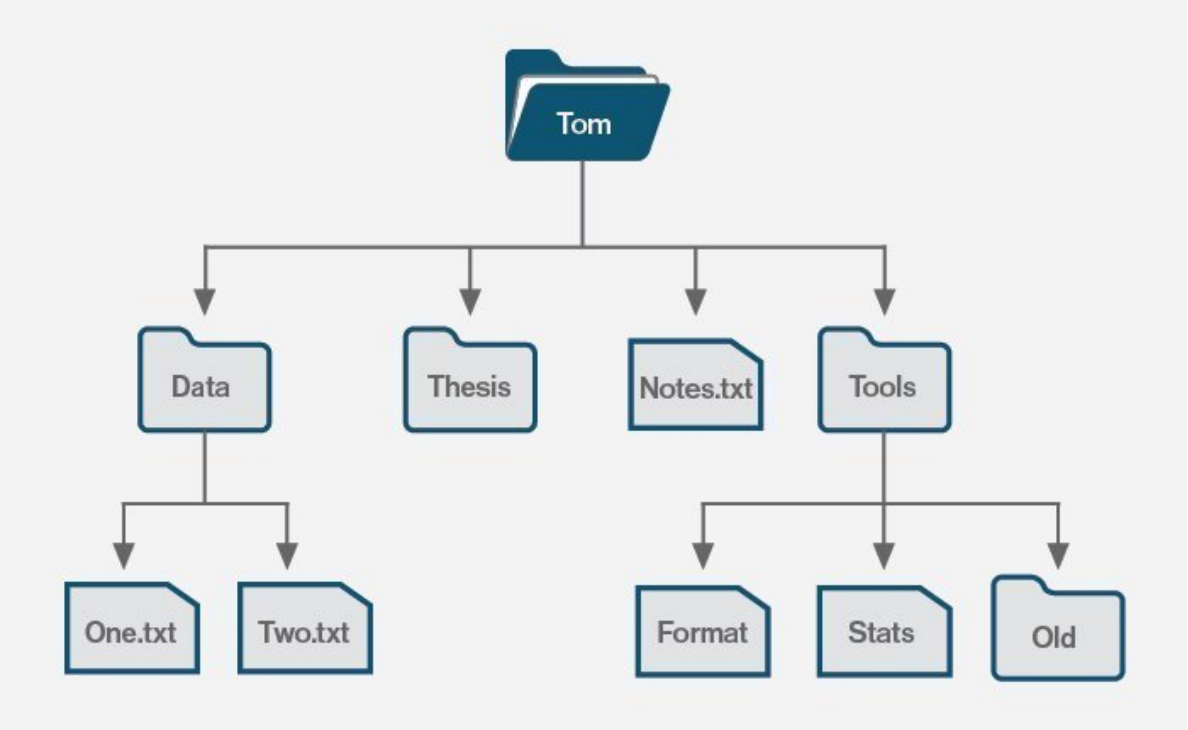


Figure 2: Example of a file system tree. Here, the root directory is "Tom".

Image source: Rouse M. https://searchstorage.techtarget.com/definition/file-system

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Navigation + Manipulating files

File/directory commands:

cp : copy
mv : move
rm : remove (*DANGER!*)

chmod : change permissions
ln : create links

File commands

cat : show contents
less : parse through
wc : view no. of lines, words, etc.

Directory commands

cd : change directory
ls : list directory contents
pwd : print working directory
mkdir : create directory
rmdir: remove directory

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Navigation

Using the "Directory commands" from the previous slide, try this!

- open a terminal

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Navigation

Using the "Directory commands" from the previous slide, try this!

- open a terminal
- find which directory you are in

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Navigation

Using the "Directory commands" from the previous slide, try this!

- open a terminal
- find which directory you are in
- make a new directory called `MBP_tech_talks`

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Navigation

Using the "Directory commands" from the previous slide, try this!

- open a terminal
- find which directory you are in
- make a new directory called `MBP_tech_talks`
- move into that directory

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Navigation

Using the "Directory commands" from the previous slide, try this!

- open a terminal
- find which directory you are in
- make a new directory called `MBP_tech_talks`
- move into that directory
- create a directory called `intro_to_bash`

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Navigation

Using the "Directory commands" from the previous slide, try this!

- open a terminal
- find which directory you are in (`pwd`)
- make a new directory called `MBP_tech_talks` (`mkdir MBP_tech_talks`)
- move into that directory (`cd MBP_tech_talks` or `cd "$_"`)
- create a directory called `intro_to_bash` (`mkdir intro_to_bash`)

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

File paths

```
$ pwd
/Users/danielle
$ cd MBP_tech_talks/intro_to_bash
$ pwd
/Users/danielle/MBP_tech_talks/intro_to_bash
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

File paths

```
$ pwd
/Users/danielle
$ cd MBP_tech_talks/intro_to_bash
$ pwd
/Users/danielle/MBP_tech_talks/intro_to_bash
```

In the example above,

- **relative path:** `(./)MBP_tech_talks/intro_to_bash`
- **absolute path:** `/Users/danielle/MBP_tech_talks/intro_to_bash`
- both paths point to end directory `intro_to_bash`
- '.' is the current working directory
- '..' is the current working directory's parent directory

Try to `cd` into `intro_to_bash` from wherever you are now using first a relative path, and then an absolute path.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Files

Let's make some files inside `intro_to_bash`.

Usually, programmers use text editors such as Emacs and Vim to create or edit text files. However, it takes some time to learn how to use either one (a whole other lesson)! So for our purposes, we'll just `echo` some sentences into two different files.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Files

Let's make some files inside `intro_to_bash`.

Usually, programmers use text editors such as Emacs and Vim to create or edit text files. However, it takes some time to learn how to use either one (a whole other lesson)! So for our purposes, we'll just `echo` some sentences into two different files.

(Try this!) First, `cd` to the desired directory. Second, execute these commands:

- `echo "Hedgehogs are great." > file1.txt`
- `echo "Yay" > file2.txt`

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

`ls` command

Now, we can use `ls` to see the files that are contained within our current working directory.

`ls` is an important command that you will use quite often! Let's look at some of its flags.

```
$ ls
file1.txt file2.txt
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

`ls` command

Now, we can use `ls` to see the files that are contained within our current working directory.

`ls` is an important command that you will use quite often! Let's look at some of its flags.

```
$ ls
file1.txt file2.txt
$ ls -l
-rw-r--r-- 1 danielle staff 21 8 Oct 22:39 file1.txt
-rw-r--r-- 1 danielle staff 4 8 Oct 22:41 file2.txt
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

`ls` command

Now, we can use `ls` to see the files that are contained within our current working directory.

`ls` is an important command that you will use quite often! Let's look at some of its flags.

```
$ ls
file1.txt file2.txt
$ ls -l
-rw-r--r-- 1 danielle staff 21 8 Oct 22:39 file1.txt
-rw-r--r-- 1 danielle staff 4 8 Oct 22:41 file2.txt
$ ls -a
.  ..  file1.txt file2.txt
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

man

If you ever want to learn more about a command, often you can find an entry in the system's reference manual.

```
$ man ls
```

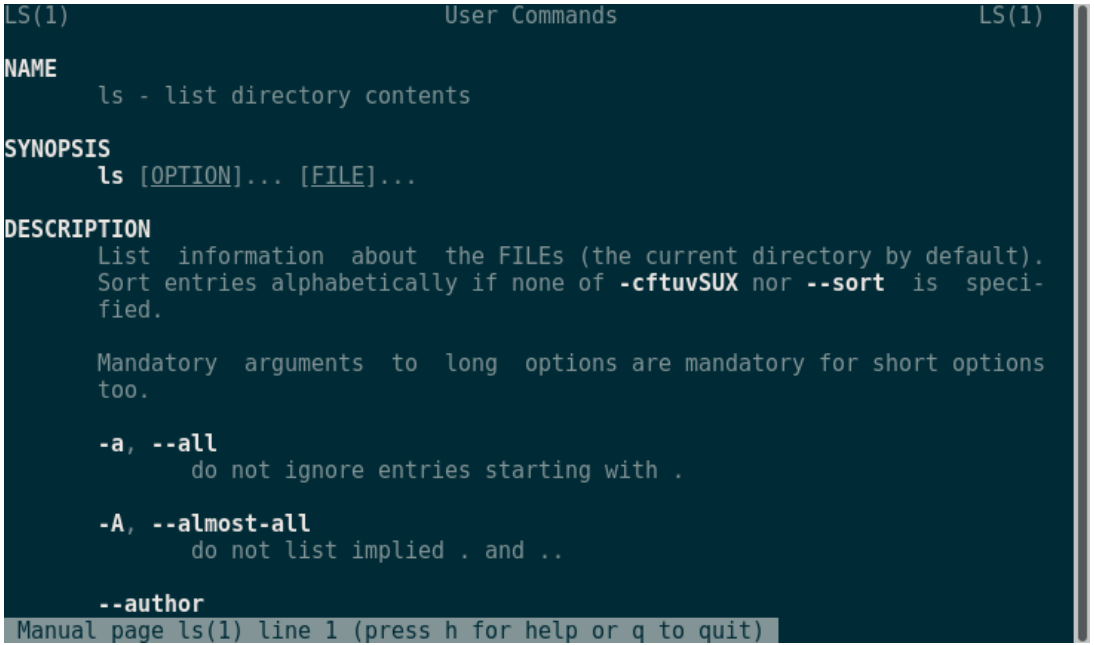


Figure 3: Manual entry for command `ls`. Hit "Enter" or up/down arrows to scroll and "q" to exit.

Redirection

Recall that we created a file using `echo`:

```
echo "Hedgehogs are great." > file1.txt
```

- command is `echo`
- argument is `"Hedgehogs are great."`
- `>` symbol redirects the output to a file

Redirection

There are two types of output.

- **standard output:** results
- **standard error:** status and error messages
- these normally go to the screen

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Redirection

There are two types of output.

- **standard output:** results
- **standard error:** status and error messages
- these normally go to the screen

We can redirect one or both types of streams:

- `>` symbol redirects standard output to a file (overwrites)
- `>>` symbol appends standard output to a file
- `2>` symbol redirects standard error to a file (overwrites)

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Redirection

There are two types of output.

- **standard output:** results
- **standard error:** status and error messages
- these normally go to the screen

We can redirect one or both types of streams:

- `>` symbol redirects standard output to a file (overwrites)
- `>>` symbol appends standard output to a file
- `2>` symbol redirects standard error to a file (overwrites)

Try it!

- add 5 lines of text (you choose!) to both `file1.txt` and `file2.txt`.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Redirection

Redirection allows you to do all sorts of cool things, like piping!

```
command 1 | command 2 | command 3
```

Try it: `$ ls | wc -l`

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

View file contents

- `head` (prints first 10 lines of file)
- `tail` (prints last 10 lines of file)
- can modify either of the above with `-n [number of lines]`

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

View file contents

- `head` (prints first 10 lines of file)
- `tail` (prints last 10 lines of file)
- can modify either of the above with `-n [number of lines]`
- `cat` (display whole file)

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

View file contents

- `head` (prints first 10 lines of file)
- `tail` (prints last 10 lines of file)
- can modify either of the above with `-n [number of lines]`
- `cat` (display whole file)
- `less` (view file page by page)

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

View file contents

- `head` (prints first 10 lines of file)
- `tail` (prints last 10 lines of file)
- can modify either of the above with `-n [number of lines]`
- `cat` (display whole file)
- `less` (view file page by page)

Try it! Print the first 3 lines of `file1.txt` and the last two lines of `file2.txt`. How would you combine these into a single file?

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Expansion

- "*" expands to any character (wildcard)

```
$ ls *.txt
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Expansion

- "*" expands to any character (wildcard)

```
$ ls *.txt
```

- "~" expands to home directory path

```
$ echo ~
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Expansion

- "*" expands to any character (wildcard)

```
$ ls *.txt
```

- "~" expands to home directory path

```
$ echo ~
```

- "$(())" executes arithmetic expressions

```
$ echo $((3+4))
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Expansion

- "*" expands to any character (wildcard)

```
$ ls *.txt
```

- "~" expands to home directory path

```
$ echo ~
```

- "$(())" executes arithmetic expressions

```
$ echo $((3+4))
```

Try it! What's $7^2 \times 5$? (hint: "**" is exponentiation, "*" is multiplication)

Intro to Bash
D. Denisko
Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Expansion

- "{}" expands to each option listed

```
$ echo before-{A,B,C}-after
$ echo {1..15}
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Expansion

- "{}" expands to each option listed

```
$ echo before-{A,B,C}-after
$ echo {1..15}
```

- "$" expands variables

```
$ echo "$USER"
$ echo "$PATH"
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Expansion

- "{}" expands to each option listed

```
$ echo before-{A,B,C}-after
$ echo {1..15}
```

- "$" expands variables

```
$ echo "$USER"
$ echo "$PATH"
```

- "$()" expands to output of command

```
$ ls -l $(which cat)
```


Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Quoting

Double quotes: carries out parameter expansion, arithmetic expansion, and command substitution.

Single quotes: suppress all expansions.

Shotts WE. 2012. The Linux Command Line.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Sed

- Sed stands for *streamline editor*
- performs commands on *each line* in stream of text
- commonly used for substitution, insertion, and deletion

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Sed

- Sed stands for *streamline editor*
- performs commands on *each line* in stream of text
- commonly used for substitution, insertion, and deletion

Substitution:
`$ sed 's/regex/replacement/g' file`
The command above would replace all (g=global) occurrences of "regex" with "replacement", both of which you assign.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Try it! Add a few more lines of text (can be different) to file1.txt, each line containing at least one occurrence of the word "blue". Now, replace all occurences of "blue" with "red". How would you replace only the first occurence of "blue" per line?

Intro to Bash
D. Denisko
Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Sed

Deletion:

```
$ sed 'nd' file
```

Delete the nth line of the file.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Sed

Deletion:

```
$ sed 'nd' file
```
Delete the nth line of the file.

```
$ sed '$d' file
```
Delete the last line of the file.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Sed

Deletion:

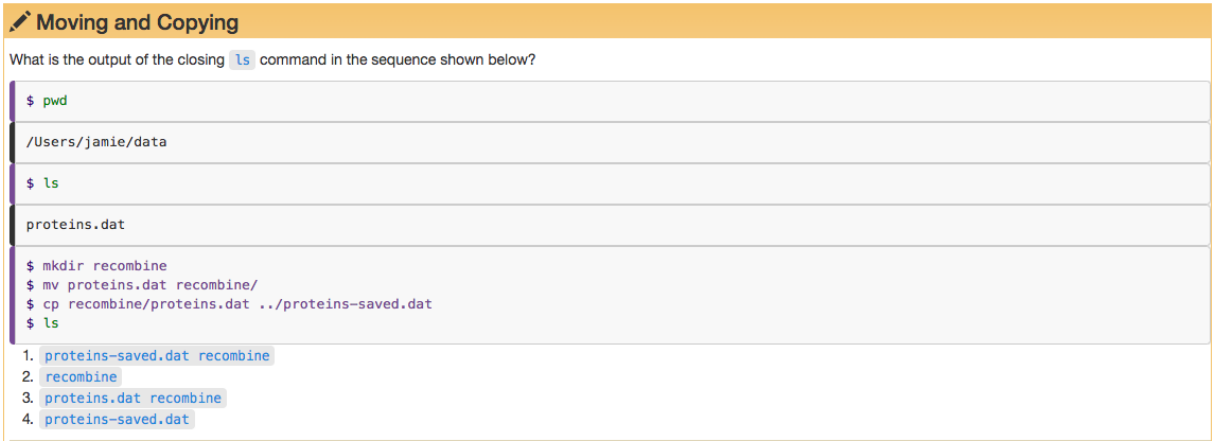
```
$ sed 'nd' file
```
Delete the nth line of the file.

```
$ sed '$d' file
```
Delete the last line of the file.

```
$ sed 'x,yd' file
```
Delete lines x through y of the file.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Exercises



Moving and Copying

What is the output of the closing `ls` command in the sequence shown below?

```
$ pwd
```

```
/Users/jamie/data
```

```
$ ls
```

```
proteins.dat
```

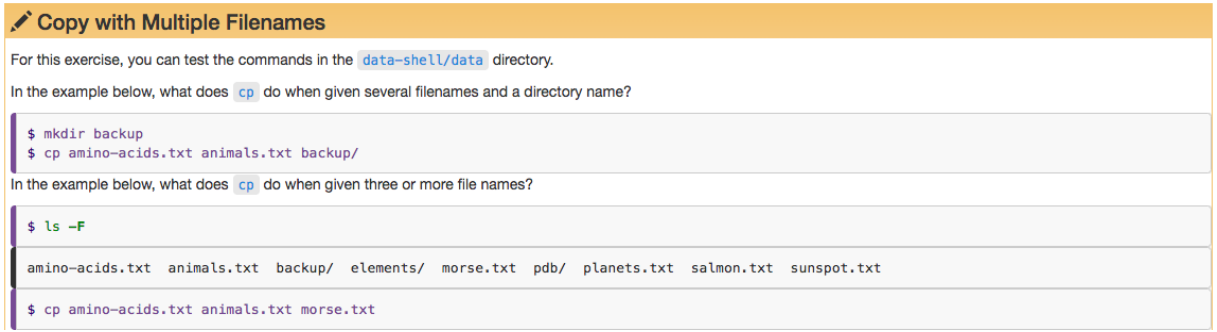
```
$ mkdir recombine
$ mv proteins.dat recombine/
$ cp recombine/proteins.dat ../proteins-saved.dat
$ ls
```

1. `proteins-saved.dat recombine`
2. `recombine`
3. `proteins.dat recombine`
4. `proteins-saved.dat`

Figure 4: Source: Working with Files and Directories, Software Carpentry.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Exercises



Copy with Multiple Filenames

For this exercise, you can test the commands in the `data-shell/data` directory.

In the example below, what does `cp` do when given several filenames and a directory name?

```
$ mkdir backup
$ cp amino-acids.txt animals.txt backup/
```

In the example below, what does `cp` do when given three or more file names?

```
$ ls -F
```

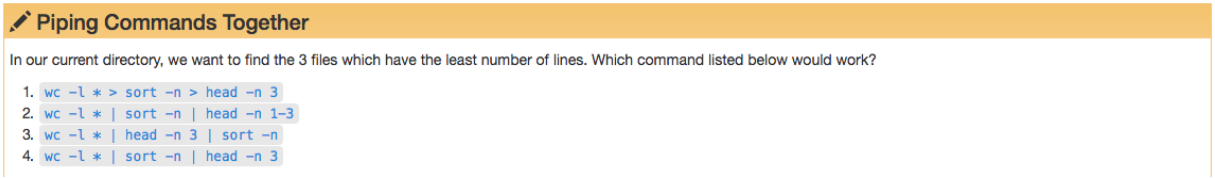
```
amino-acids.txt  animals.txt  backup/  elements/  morse.txt  pdb/  planets.txt  salmon.txt  sunspot.txt
```

```
$ cp amino-acids.txt animals.txt morse.txt
```

Figure 5: Source: Working with Files and Directories, Software Carpentry.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Exercises



Piping Commands Together

In our current directory, we want to find the 3 files which have the least number of lines. Which command listed below would work?

1. `wc -l * > sort -n > head -n 3`
2. `wc -l * | sort -n | head -n 1-3`
3. `wc -l * | head -n 3 | sort -n`
4. `wc -l * | sort -n | head -n 3`

Figure 6: Source: Pipes and Filters, Software Carpentry.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Exercises

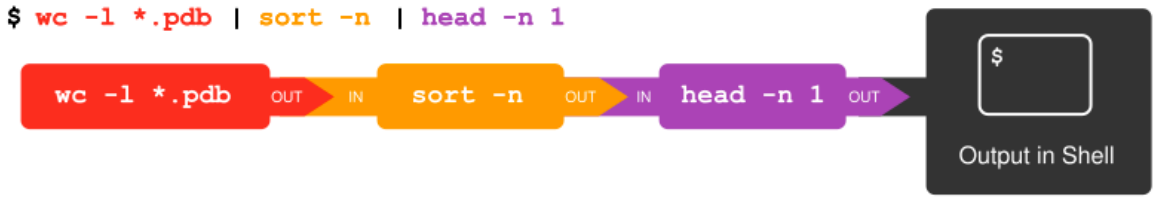
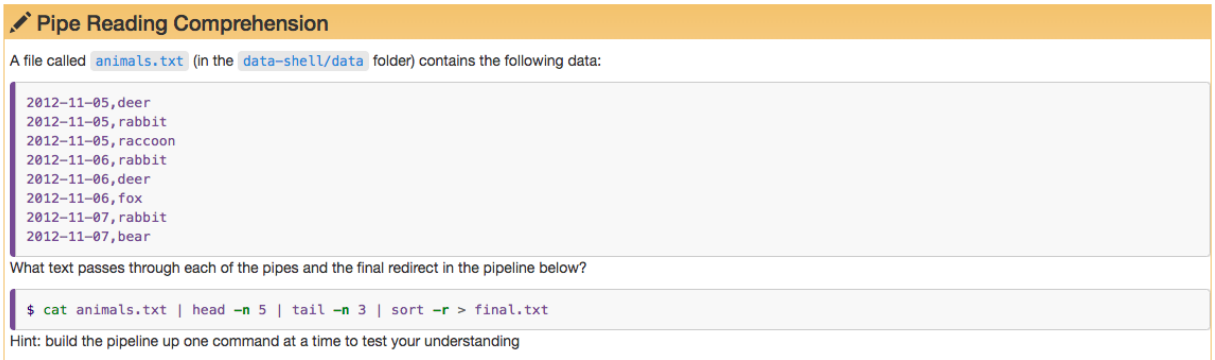


Figure 7: Source: Pipes and Filters, Software Carpentry.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Exercises



Pipe Reading Comprehension

A file called `animals.txt` (in the `data-shell/data` folder) contains the following data:

```
2012-11-05,deer
2012-11-05,rabbit
2012-11-05,raccoon
2012-11-06,rabbit
2012-11-06,deer
2012-11-06,fox
2012-11-07,rabbit
2012-11-07,bear
```

What text passes through each of the pipes and the final redirect in the pipeline below?

```
$ cat animals.txt | head -n 5 | tail -n 3 | sort -r > final.txt
```

Hint: build the pipeline up one command at a time to test your understanding

Figure 8: Source: Pipes and Filters, Software Carpentry.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Other topics

- the environment (~/`.bashrc`)
- loops, if statements
- shell scripts
- regular expressions (i.e. regexes)
- symbolic links
- more useful commands (i.e. Coreutils)
- submitting jobs to cluster queue

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References

Resources and further readings


Software Carpentry. *The Unix Shell*. 2018. URL: https://swcarpentry.github.io/shell-novice/.


Introduction to Bash. URL: http://cs.lmu.edu/~ray/notes/bash/.


William E. Shotts Jr. *The Linux Command Line: A Complete Introduction*. San Francisco, CA, USA: No Starch Press, 2012. ISBN: 9781593273897.

Introduction
What is the shell?
When and how to use Bash
Navigation + Editing files
The file system
Navigation
Absolute vs. relative file paths
Creating files
Output redirection
Expansion
Quoting
Sed (intro)
Exercises
Other topics
References