

Basic Unix Tutorial

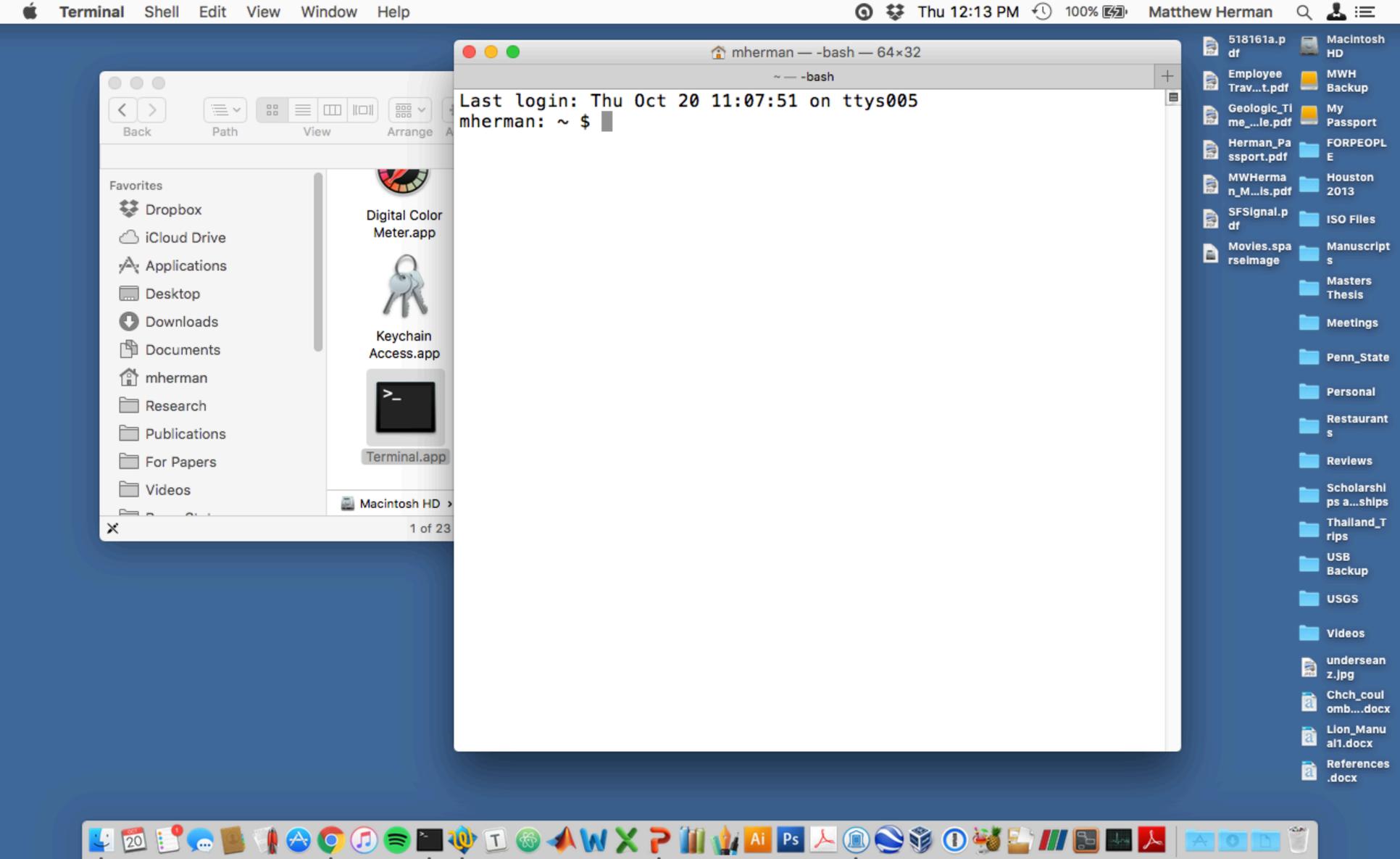
Matt Herman

Geodynamics Research Group

Penn State

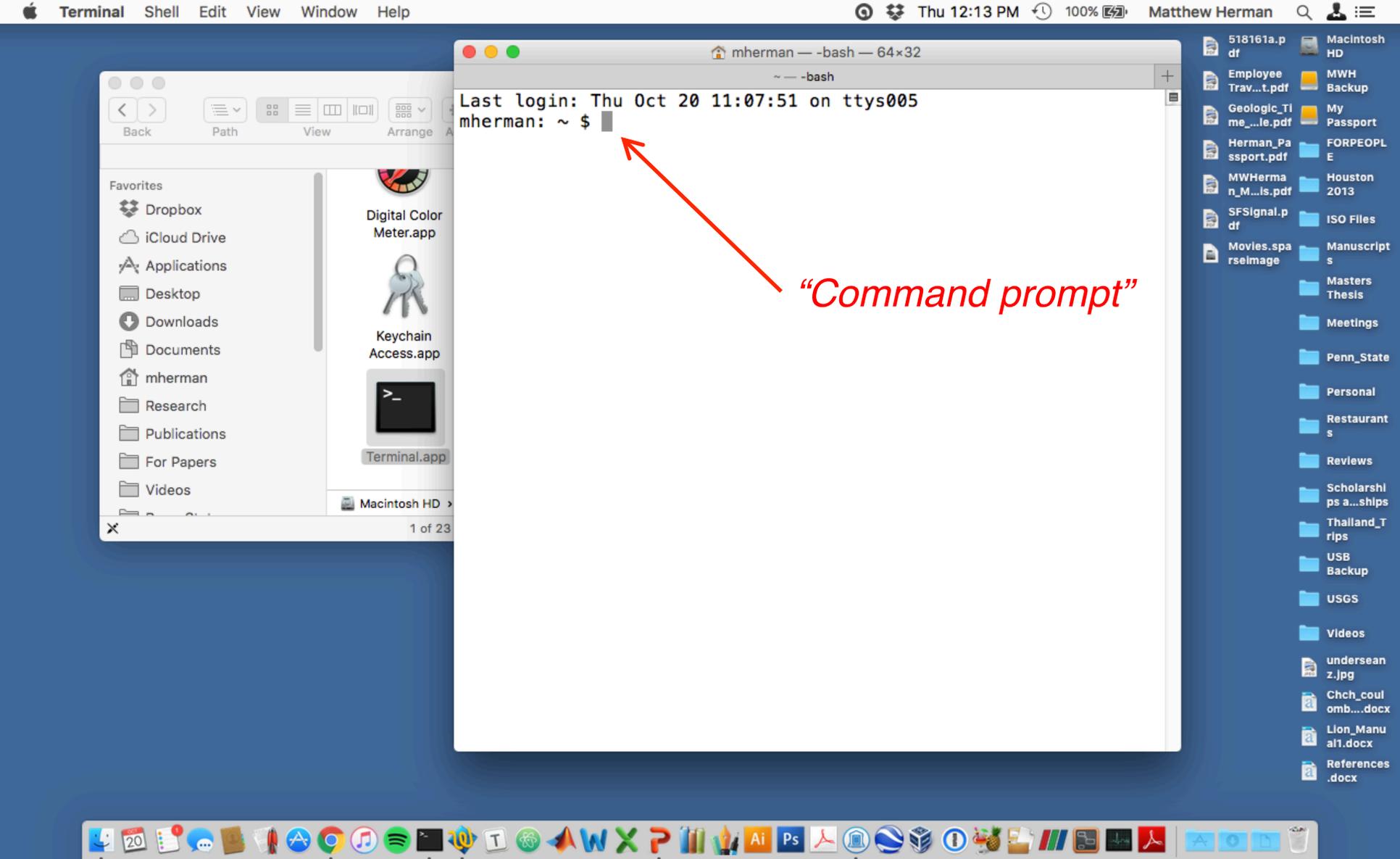
Tutorial Objective

- This is designed to teach you enough to start using the Generic Mapping Tools
- Just the tip of the iceberg!



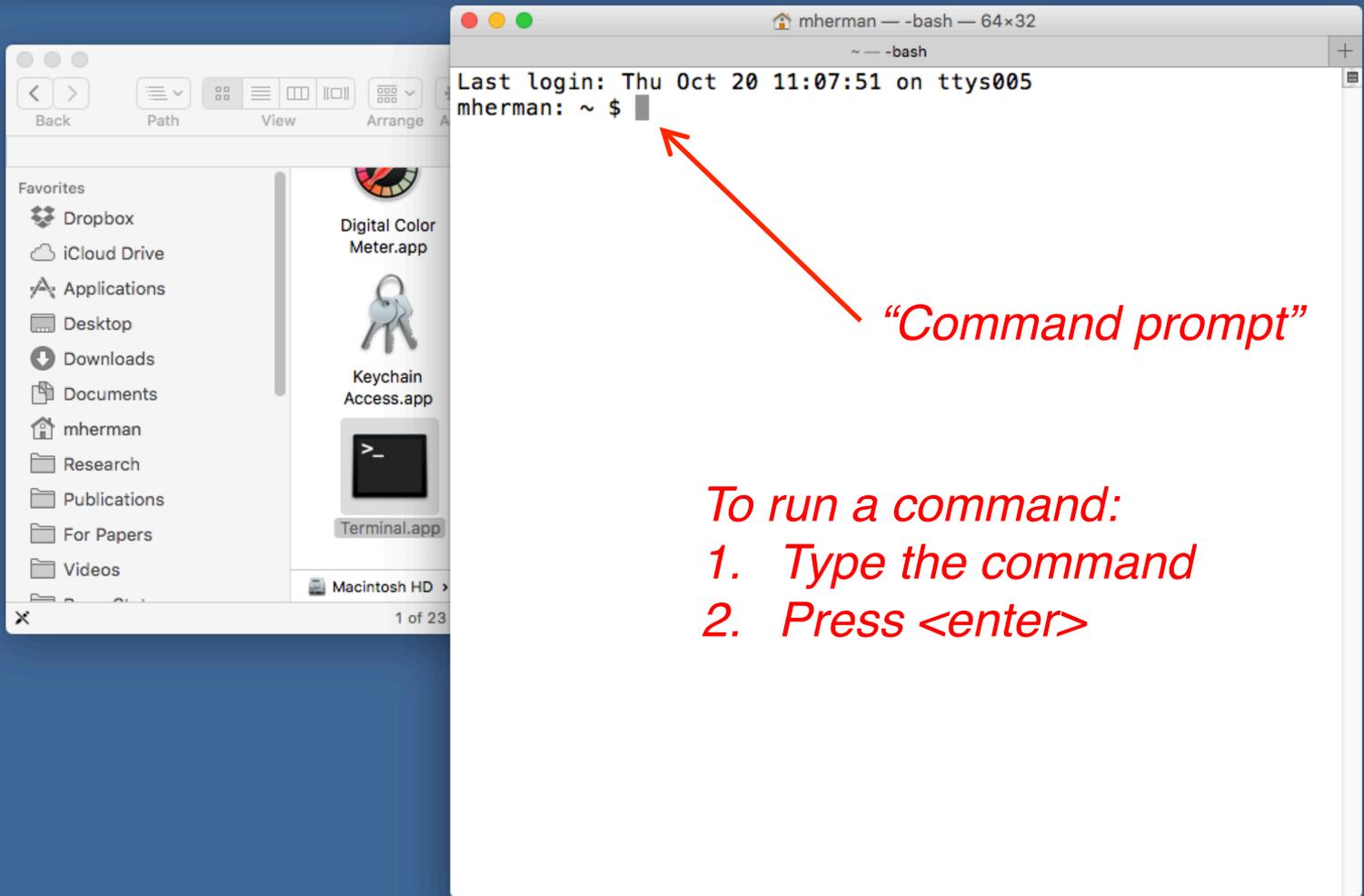
Open the Terminal application

- On a Mac: Applications -> Utilities -> Terminal



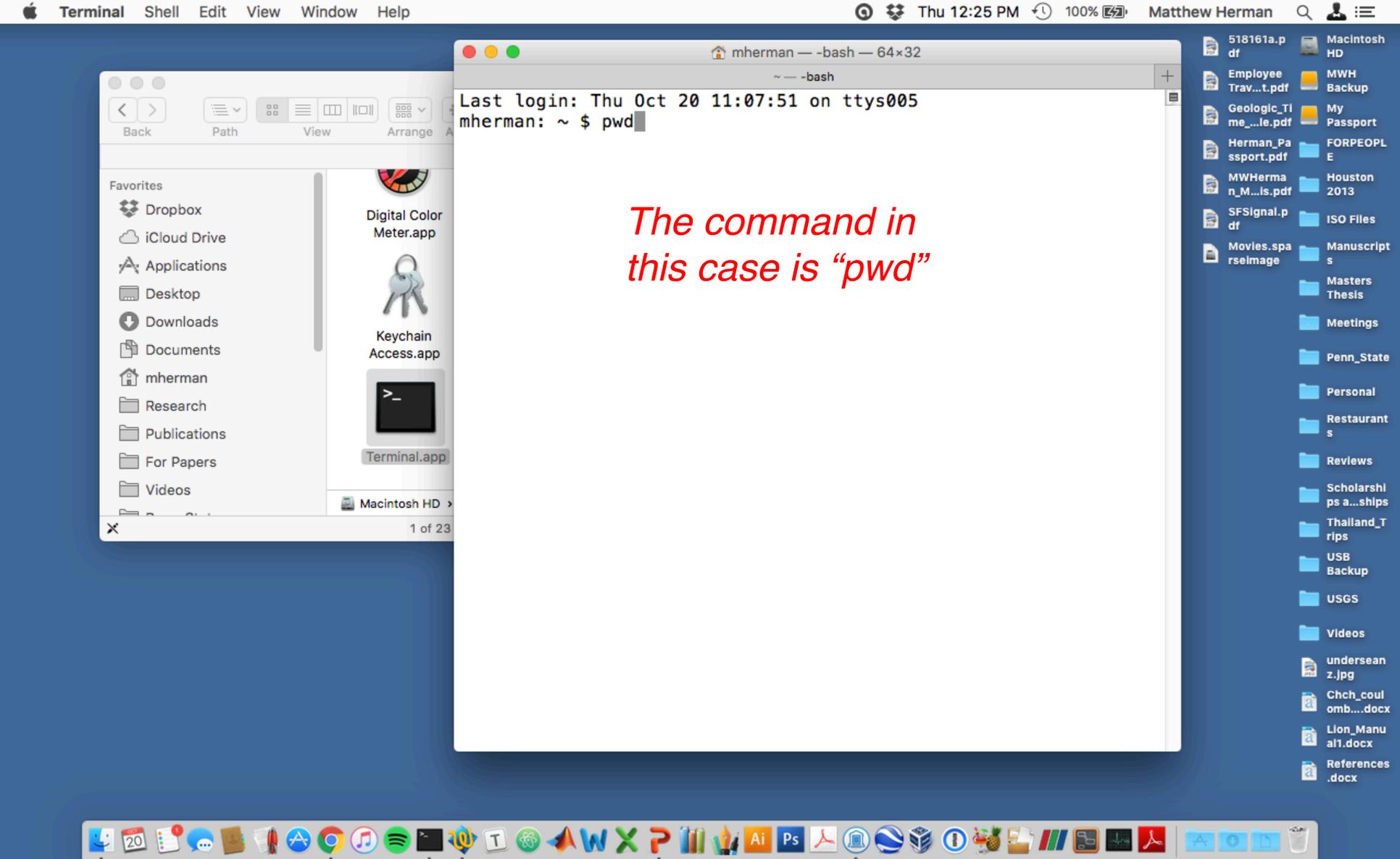
Open the Terminal application

- On a Mac: Applications -> Utilities -> Terminal



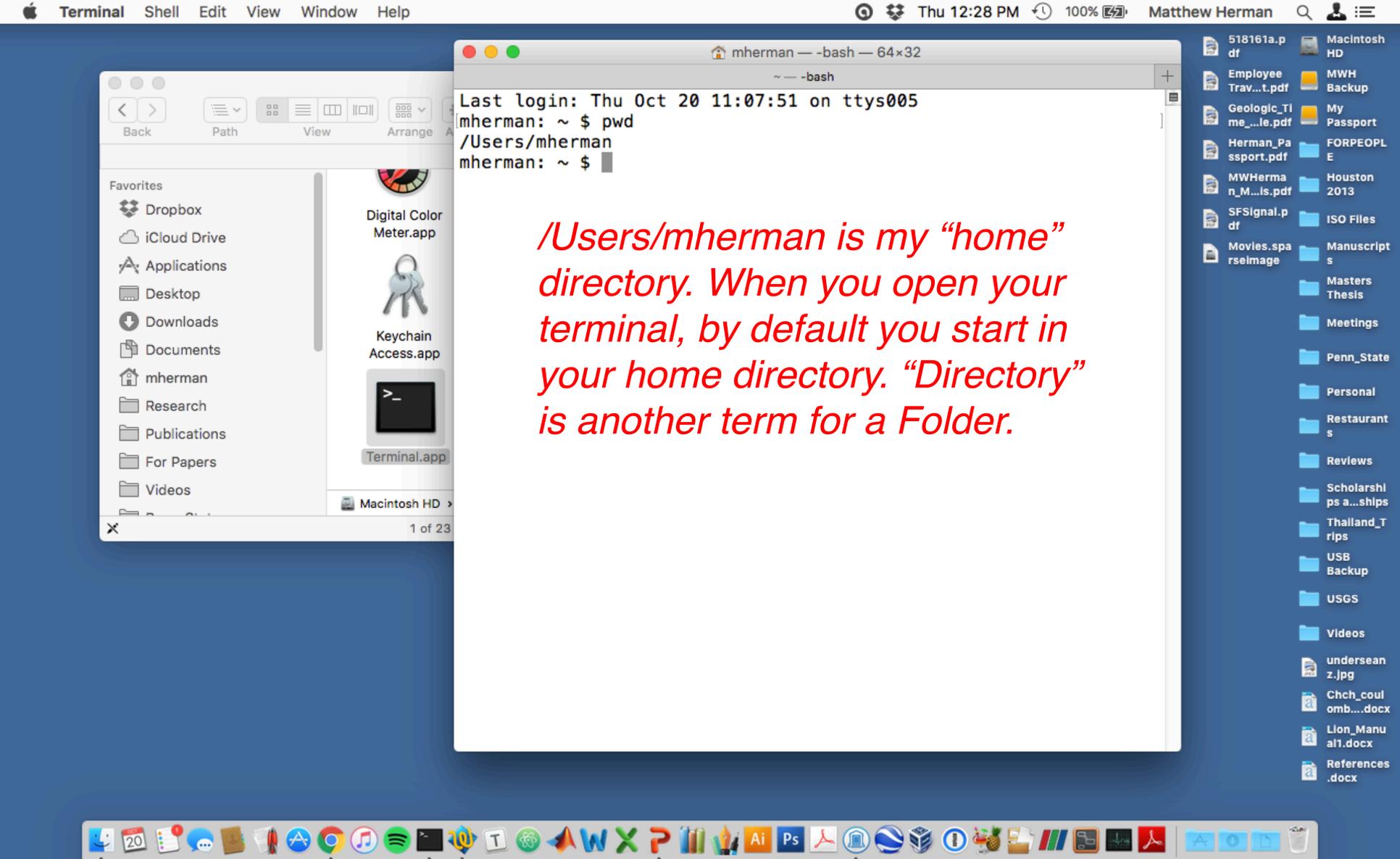
Open the Terminal application

- On a Mac: Applications -> Utilities -> Terminal



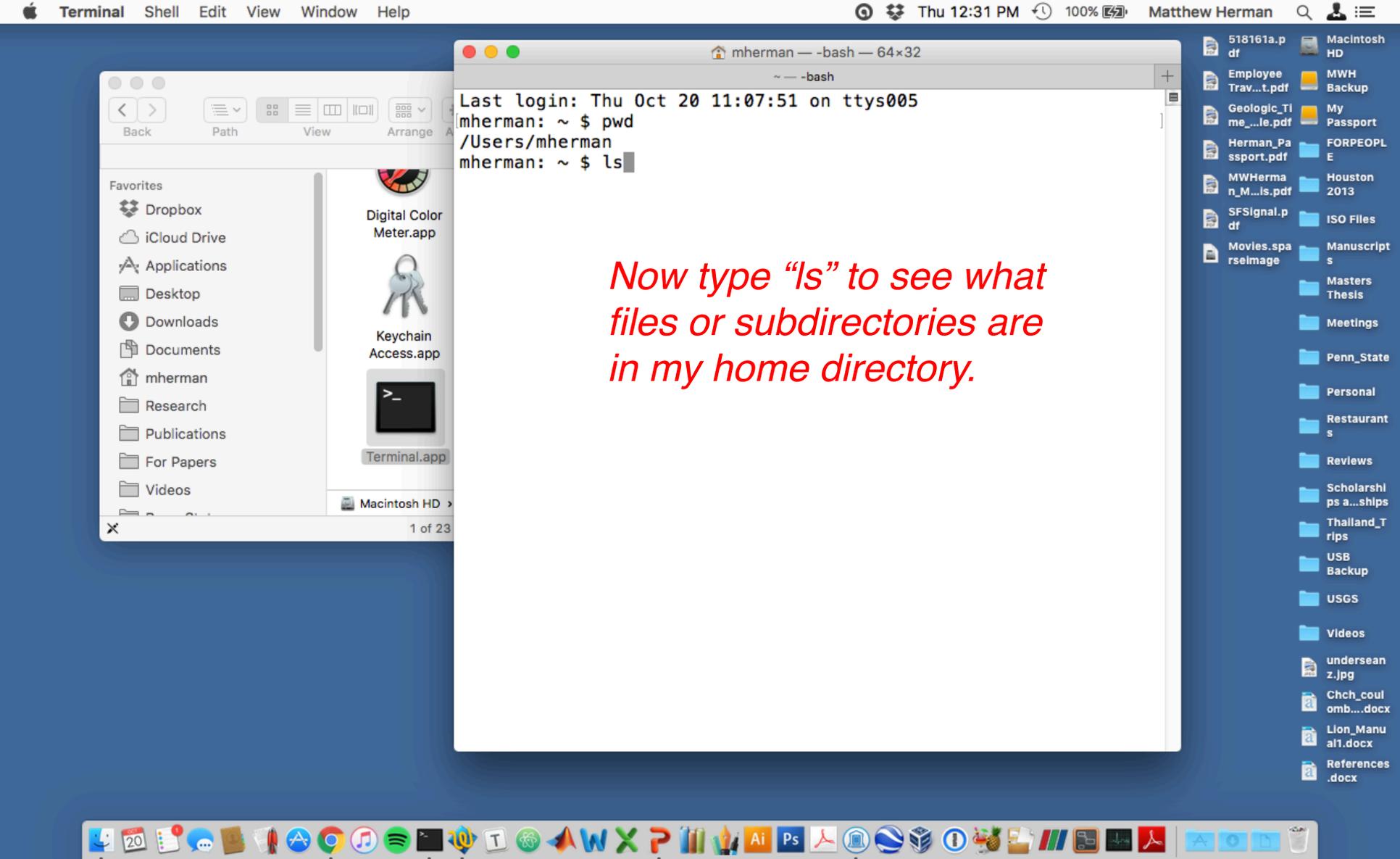
Navigating in the terminal

- pwd: print present working directory



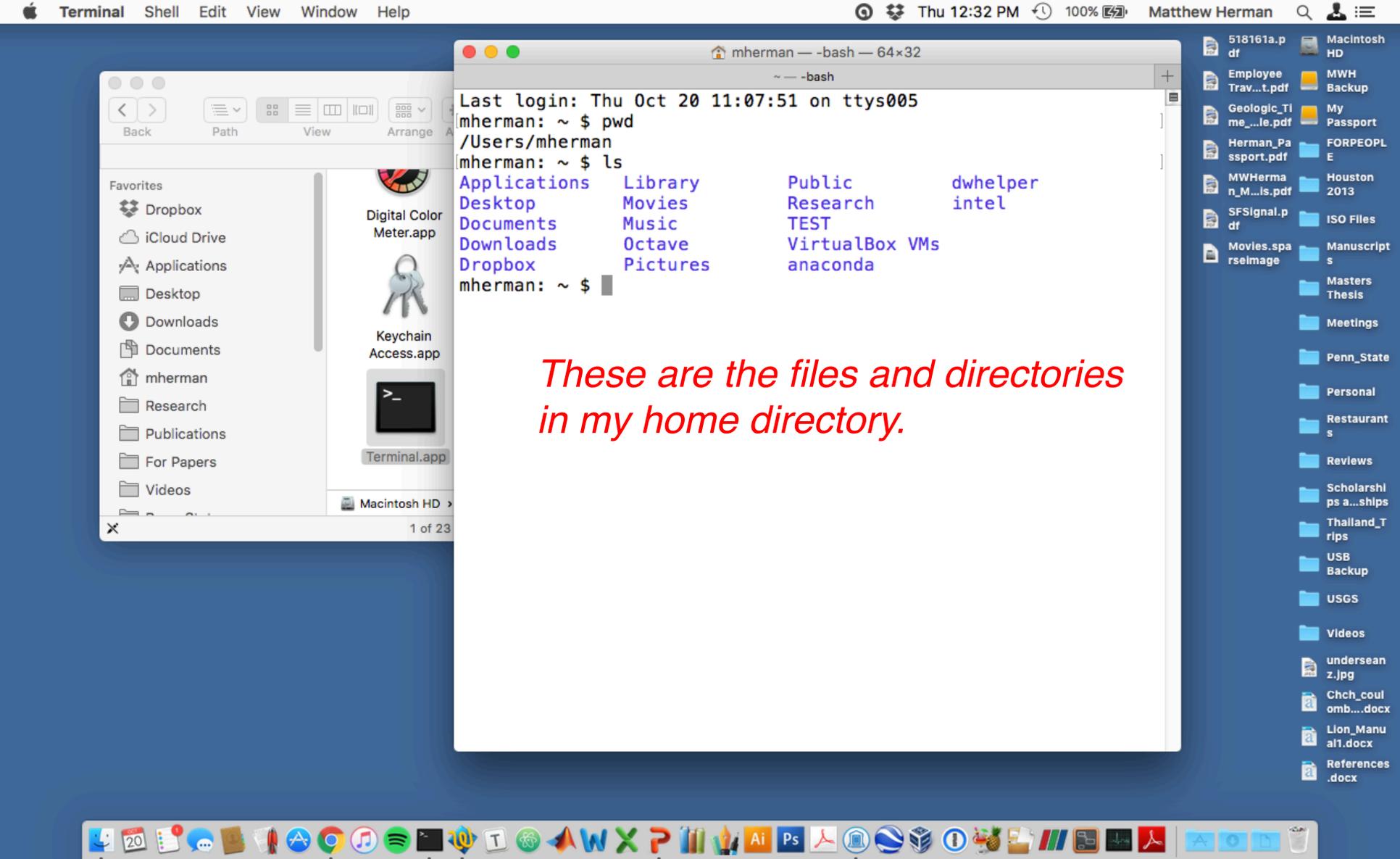
Navigating in the terminal

- pwd: print present working directory



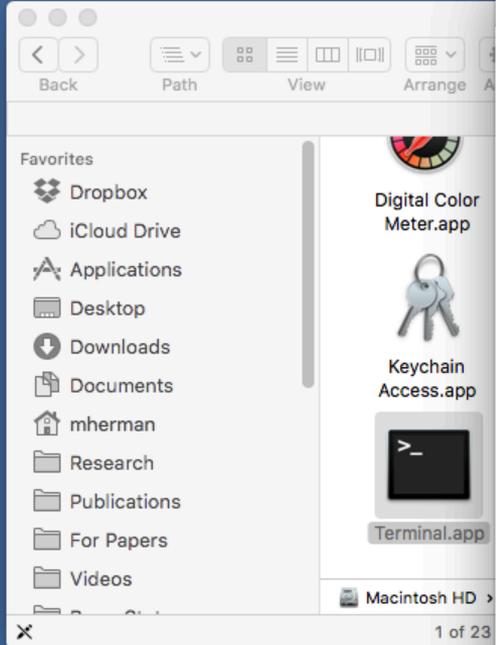
Navigating in the terminal

- `ls`: list contents of current directory



Navigating in the terminal

- `ls`: list contents of current directory



```

mherman — -bash — 64x32
~ — -bash
Last login: Thu Oct 20 11:07:51 on ttys005
mherman: ~ $ pwd
/Users/mherman
mherman: ~ $ ls
Applications  Library          Public           dwhelper
Desktop       Movies           Research         intel
Documents     Music           TEST
Downloads     Octave          VirtualBox VMS
Dropbox       Pictures         anaconda
mherman: ~ $ cd Desktop

```

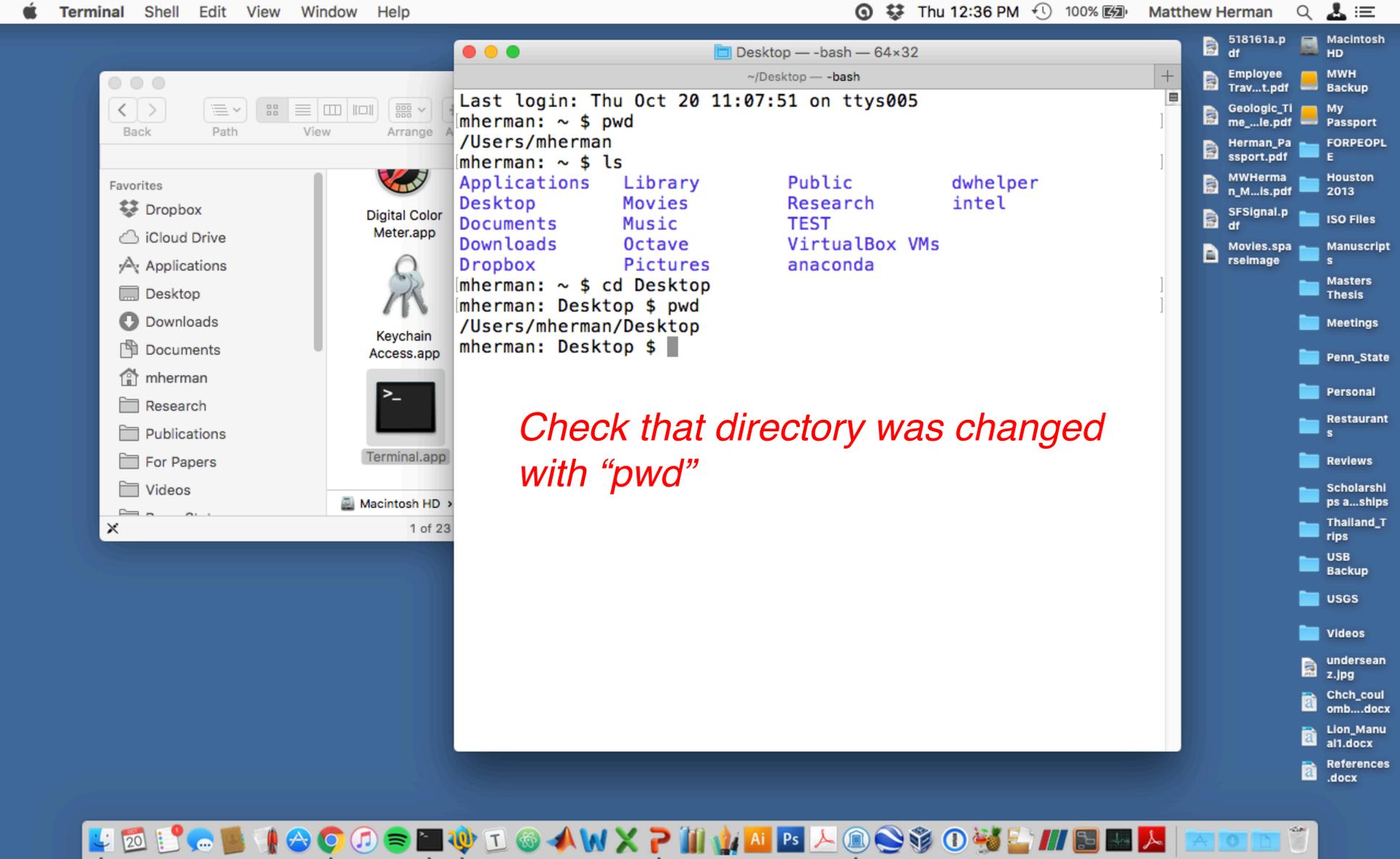
Let's change directory to the Desktop. Type in "cd Desktop"

The syntax is always "cd <directory>". In other words, cd requires the name of a directory as an "argument."



Navigating in the terminal

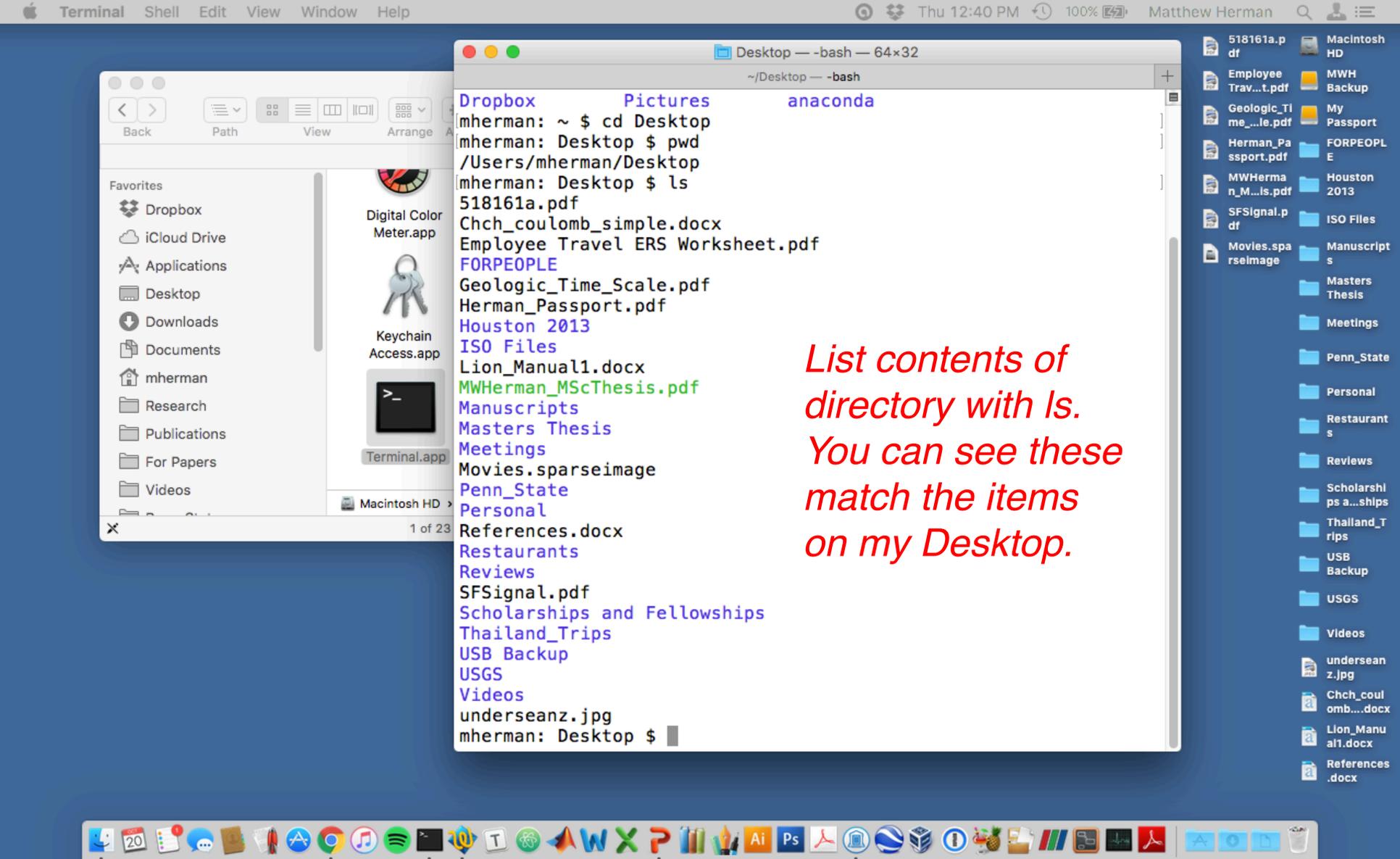
- cd: change directory



Check that directory was changed with "pwd"

Navigating in the terminal

- cd: change directory

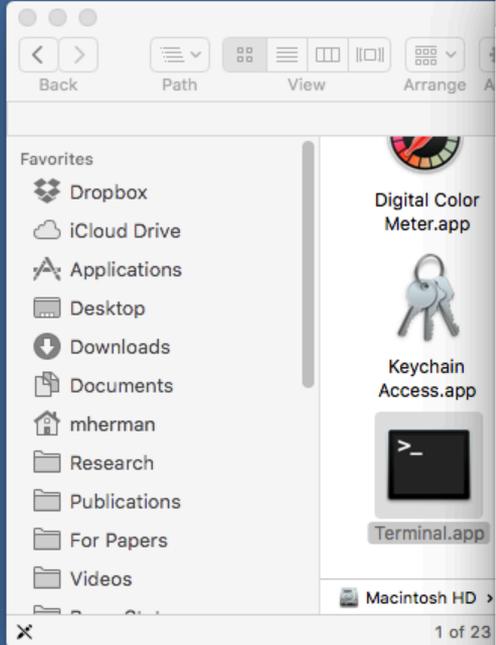


```
Desktop — -bash — 64x32
~/Desktop — -bash
Dropbox      Pictures      anaconda
mherman: ~ $ cd Desktop
mherman: Desktop $ pwd
/Users/mherman/Desktop
mherman: Desktop $ ls
518161a.pdf
Chch_coulomb_simple.docx
Employee Travel ERS Worksheet.pdf
FORPEOPLE
Geologic_Time_Scale.pdf
Herman_Passport.pdf
Houston 2013
ISO Files
Lion_Manual1.docx
MWHerman_MScThesis.pdf
Manuscripts
Masters Thesis
Meetings
Movies.sparseimage
Penn_State
Personal
References.docx
Restaurants
Reviews
SFSignal.pdf
Scholarships and Fellowships
Thailand_Trips
USB Backup
USGS
Videos
underseanz.jpg
mherman: Desktop $
```

List contents of directory with ls. You can see these match the items on my Desktop.

Navigating in the terminal

- cd: change directory



```

Desktop — -bash — 64x32
~/Desktop — -bash
Dropbox      Pictures      anaconda
mherman: ~ $ cd Desktop
mherman: Desktop $ pwd
/Users/mherman/Desktop
mherman: Desktop $ ls
518161a.pdf
Chch_coulomb_simple.docx
Employee Travel ERS Worksheet.pdf
FORPEOPLE
Geologic_Time_Scale.pdf
Herman_Passport.pdf
Houston 2013
ISO Files
Lion_Manual1.docx
MWHerman_MScThesis.pdf
Manuscripts
Masters Thesis
Meetings
Movies.sparseimage
Penn_State
Personal
References.docx
Restaurants
Reviews
SFSignal.pdf
Scholarships and Fellowships
Thailand_Trips
USB Backup
USGS
Videos
underseanz.jpg
mherman: Desktop $

```

Two special directories are "." and ".."

- *“.” is the current directory*
- *“..” is the directory above the current one.*



Navigating in the terminal

- cd: change directory



The image shows a Mac desktop with a file browser window on the left and a terminal window on the right. The file browser shows a directory listing of files and folders. The terminal window shows the following output:

```

Geologic_Time_Scale.pdf
Herman_Passport.pdf
Houston 2013
ISO Files
Lion_Manual1.docx
MWHerman_MScThesis.pdf
Manuscripts
Masters Thesis
Meetings
Movies.sparseimage
Penn_State
Personal
References.docx
Restaurants
Reviews
SFSignal.pdf
Scholarships and Fellowships
Thailand_Trips
USB Backup
USGS
Videos
underseanz.jpg
mherman: Desktop $ cd ..
mherman: ~ $ pwd
/Users/mherman
mherman: ~ $ ls
Applications      Library           Public            dwhelper
Desktop           Movies           Research          intel
Documents         Music            TEST
Downloads        Octave          VirtualBox VMs
Dropbox          Pictures         anaconda
mherman: ~ $

```

A red arrow points from the text annotation to the `cd ..` command in the terminal.

Here I used "cd .." to go up one directory, ending up back in my home directory.

Navigating in the terminal

- cd: change directory

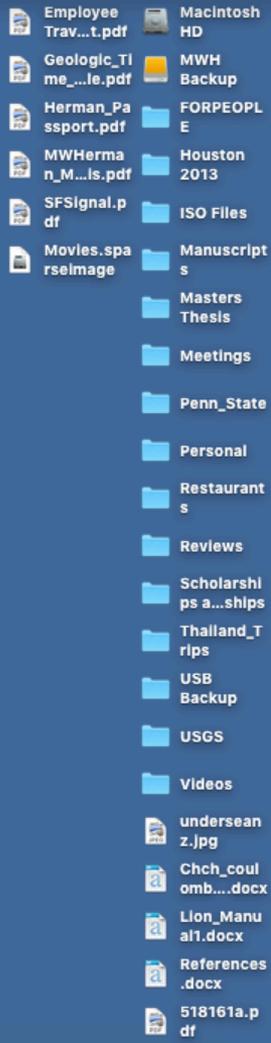
```

mherman: ~ -bash — 64x32
~ -bash
mherman: ~ $ ls
Applications  Library      Public      dwhelper
Desktop       Movies       Research    intel
Documents     Music        TEST
Downloads     Octave       VirtualBox VMs
Dropbox       Pictures     anaconda
mherman: ~ $ mkdir unix_tutorial ←
mherman: ~ $ ls
Applications  Library      Public      dwhelper
Desktop       Movies       Research    intel
Documents     Music        TEST        unix_tutorial
Downloads     Octave       VirtualBox VMs
Dropbox       Pictures     anaconda
mherman: ~ $ █

```

Back in the home directory, I typed “mkdir unix_tutorial” to make a new folder called unix_tutorial.

Note: avoid using spaces for file or directory names.



Making a new directory

- mkdir: make new directory

```
unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash

[mherman: ~ $ ls
Applications      Library           Public            dwhelper
Desktop           Movies           Research          intel
Documents         Music            TEST
Downloads         Octave          VirtualBox VMs
Dropbox           Pictures         anaconda

[mherman: ~ $ mkdir unix_tutorial
[mherman: ~ $ ls
Applications      Library           Public            dwhelper
Desktop           Movies           Research          intel
Documents         Music            TEST             unix_tutorial
Downloads         Octave          VirtualBox VMs
Dropbox           Pictures         anaconda

[mherman: ~ $ cd unix_tutorial
[mherman: unix_tutorial $ ls
mherman: unix_tutorial $
```

*We will work in this new directory.
Change into unix_tutorial and
verify the directory is clean and
empty.*

Making a new directory

- mkdir: make new directory

```

unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash

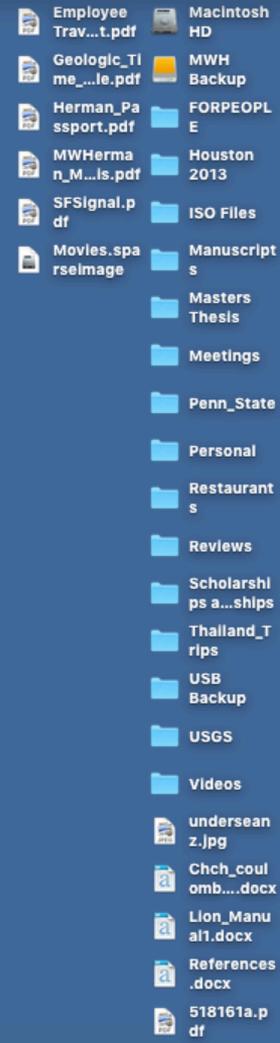
mherman: ~ $ ls
Applications  Library      Public      dwhelper
Desktop       Movies       Research    intel
Documents     Music        TEST
Downloads     Octave       VirtualBox  VMs
Dropbox       Pictures     anaconda

mherman: ~ $ mkdir unix_tutorial
mherman: ~ $ ls
Applications  Library      Public      dwhelper
Desktop       Movies       Research    intel
Documents     Music        TEST        unix_tutorial
Downloads     Octave       VirtualBox  VMs
Dropbox       Pictures     anaconda

mherman: ~ $ cd unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ touch file1
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $

```

Type in the command "touch file1" to create an empty file named "file1." Verify that this file is now in your directory.



Manipulating files

- touch: make a new blank file

```

unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash

mherman: ~ $ ls
Applications  Library      Public      dwhelper
Desktop       Movies       Research    intel
Documents     Music        TEST
Downloads     Octave       VirtualBox  VMs
Dropbox       Pictures     anaconda

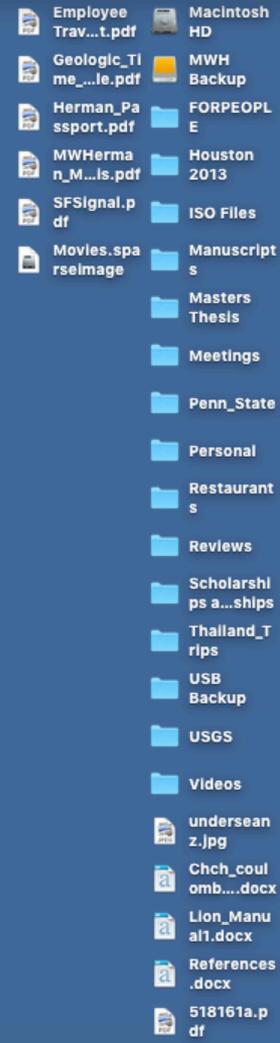
mherman: ~ $ mkdir unix_tutorial
mherman: ~ $ ls
Applications  Library      Public      dwhelper
Desktop       Movies       Research    intel
Documents     Music        TEST        unix_tutorial
Downloads     Octave       VirtualBox  VMs
Dropbox       Pictures     anaconda

mherman: ~ $ cd unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ touch file1
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $ cp file1 file2
mherman: unix_tutorial $ ls
file1 file2
mherman: unix_tutorial $

```

To duplicate a file, use the copy command: "cp file1 file2"

This copies everything in file1 into file2, and keeps file1.



Manipulating files

- cp: copy file

```
unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash
mherman: ~ $ ls
Applications  Library          Public           dwhelper
Desktop        Movies           Research         intel
Documents      Music            TEST
Downloads     Octave          VirtualBox VMs
Dropbox        Pictures         anaconda
mherman: ~ $ mkdir unix_tutorial
mherman: ~ $ ls
Applications  Library          Public           dwhelper
Desktop        Movies           Research         intel
Documents      Music            TEST             unix_tutorial
Downloads     Octave          VirtualBox VMs
Dropbox        Pictures         anaconda
mherman: ~ $ cd unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ touch file1
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $ cp file1 file2
mherman: unix_tutorial $ ls
file1 file2
mherman: unix_tutorial $ mv file2 file3
mherman: unix_tutorial $ ls
file1 file3
mherman: unix_tutorial $
```

*To rename a file, deleting the original,
use the move command:
"mv file2 file3"*

Manipulating files

- mv: move file

```

unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash

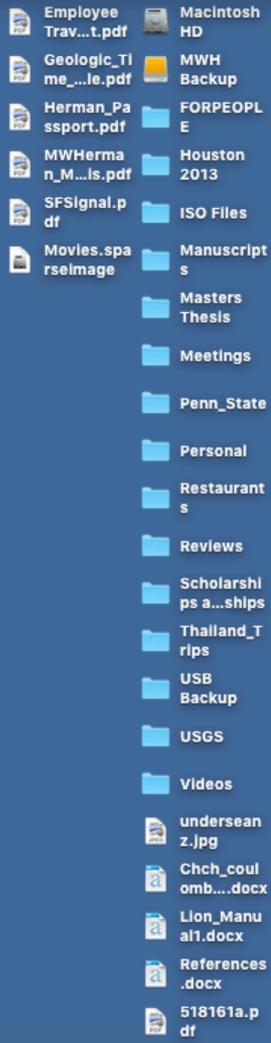
mherman: ~ $ ls
Applications      Library           Public            dwhelper
Desktop           Movies           Research         intel
Documents         Music            TEST
Downloads        Octave          VirtualBox VMs
Dropbox          Pictures        anaconda

mherman: ~ $ mkdir unix_tutorial
mherman: ~ $ ls
Applications      Library           Public            dwhelper
Desktop           Movies           Research         intel
Documents         Music            TEST             unix_tutorial
Downloads        Octave          VirtualBox VMs
Dropbox          Pictures        anaconda

mherman: ~ $ cd unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ touch file1
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $ cp file1 file2
mherman: unix_tutorial $ ls
file1 file2
mherman: unix_tutorial $ mv file2 file3
mherman: unix_tutorial $ ls
file1 file3
mherman: unix_tutorial $ rm file3
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $ █

```

To delete a file, use the rm command: "rm file3"



Manipulating files

- rm: remove file (IRREVERSIBLE!)

```
unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash
mherman: ~ $ ls
Applications  Library      Public      dwhelper
Desktop        Movies       Research    intel
Documents      Music        TEST
```

You cannot undo the “rm” operation. Once a file is deleted, it is gone forever (equivalent to emptying trash). Use with caution.

```
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $
```

To delete a file, use the rm command: “rm file3”

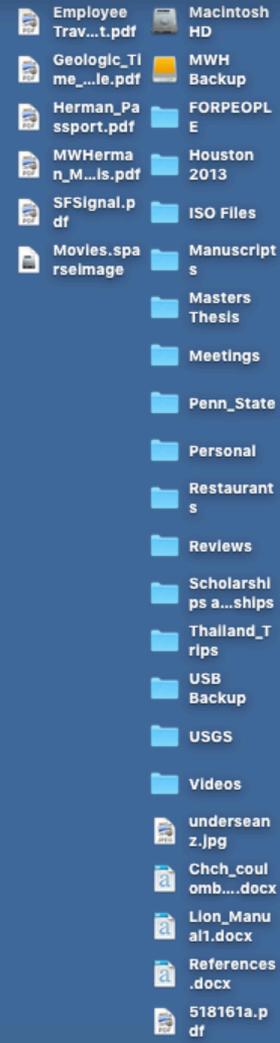


Manipulating files

- rm: remove file (IRREVERSIBLE!)

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ pwd
/Users/mherman/unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $
```

I am starting with a clean unix_tutorial directory. I removed the last file in the directory with "rm file1"



Input and output

- echo: print following values to terminal screen

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ pwd
/Users/mherman/unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $
```

The echo command prints its arguments (whatever follows it) to the terminal screen. For example, typing

“echo 1 2 3 4”

prints “1 2 3 4” on the next line.

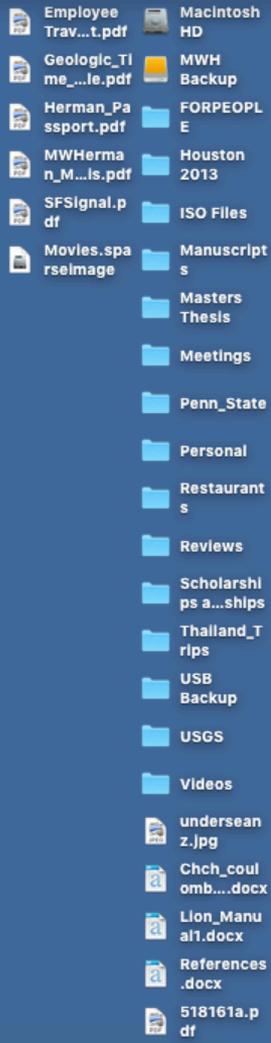
Input and output

- echo: print following values to terminal screen

```
unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash
mherman: unix_tutorial $ pwd
/Users/mherman/unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 1 2 3 4 > file1
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $
```

You can take this output and put it into a file instead of printing it to the terminal. This is done by using “>” (the redirect arrow). For example, to save the previous output into file1, use

“echo 1 2 3 4 > file1”

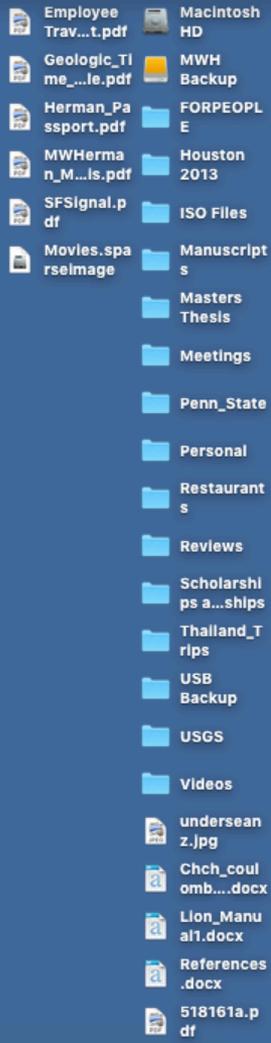


Input and output

- >: redirect output to a file (overwrite)

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ pwd
/Users/mherman/unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 1 2 3 4 > file1
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $ cat file1
1 2 3 4
mherman: unix_tutorial $
```

To verify that the file contains what we expect, type “cat file1”. This prints the contents of a file to the terminal screen.



Input and output

- cat: concatenate (print contents of) file

```
unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash
mherman: unix_tutorial $ pwd
/Users/mherman/unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 1 2 3 4 > file1
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $ cat file1
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8 > file1
mherman: unix_tutorial $ cat file1
5 6 7 8
mherman: unix_tutorial $
```

The single redirect arrow (“>”) overwrites the contents of the file. Verify this by typing in the command

“echo 5 6 7 8 > file1”

Input and output

- >>: redirect output to a file (append)

```
unix_tutorial -- -bash -- 64x32
~/unix_tutorial -- -bash
mherman: unix_tutorial $ pwd
/Users/mherman/unix_tutorial
mherman: unix_tutorial $ ls
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 1 2 3 4 > file1
mherman: unix_tutorial $ ls
file1
mherman: unix_tutorial $ cat file1
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8 > file1
mherman: unix_tutorial $ cat file1
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12 >> file1
mherman: unix_tutorial $ cat file1
5 6 7 8
9 10 11 12
mherman: unix_tutorial $
```

To append (add on) to a file, use the double redirect arrow (“>>”).

“echo 9 10 11 12 >> file1”

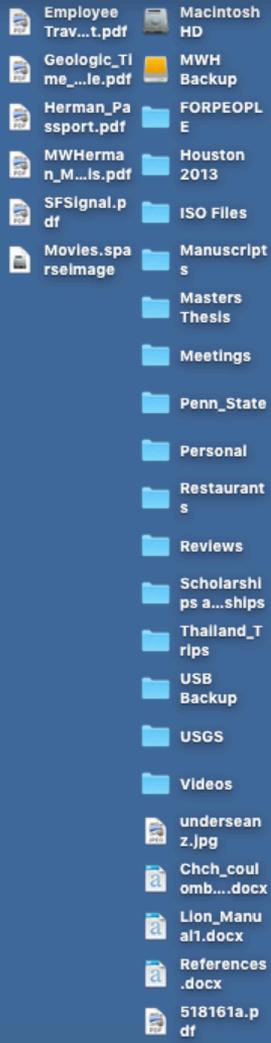
Input and output

- >>: redirect output to a file (append)




```
unix_tutorial — less file1 — 64x32
~/unix_tutorial — less file1
5 6 7 8
9 10 11 12
file1 (END)
```

The commands “less” and “more” open a text file instead of printing its contents (like “cat”). I prefer “less” but try both. To exit, press “q”.



Examining files

- less/more: open a file for reading in the terminal

```

unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $

```

Sharing Screenshot
A link to your screenshot has been copied to your clipboard.
Close

- Employee
- Macintosh
- ssport.pdf
- MWHerman_M...ls.pdf
- SFSignal.pdf
- Movies.sparselmage
- Houston 2013
- ISO Files
- Manuscripts
- Masters Thesis
- Meetings
- Penn_State
- Personal
- Restaurants
- Reviews
- Scholarships and ships
- Thailand_Trips
- USB Backup
- USGS
- Videos
- underseanz.jpg
- Chch_coulomb....docx
- Lion_Manual1.docx
- References.docx
- 518161a.pdf

Imagine that you want to run several commands in a row, like shown above. It would be a pain to type them all into the terminal, especially if you wanted to run the series more than once.

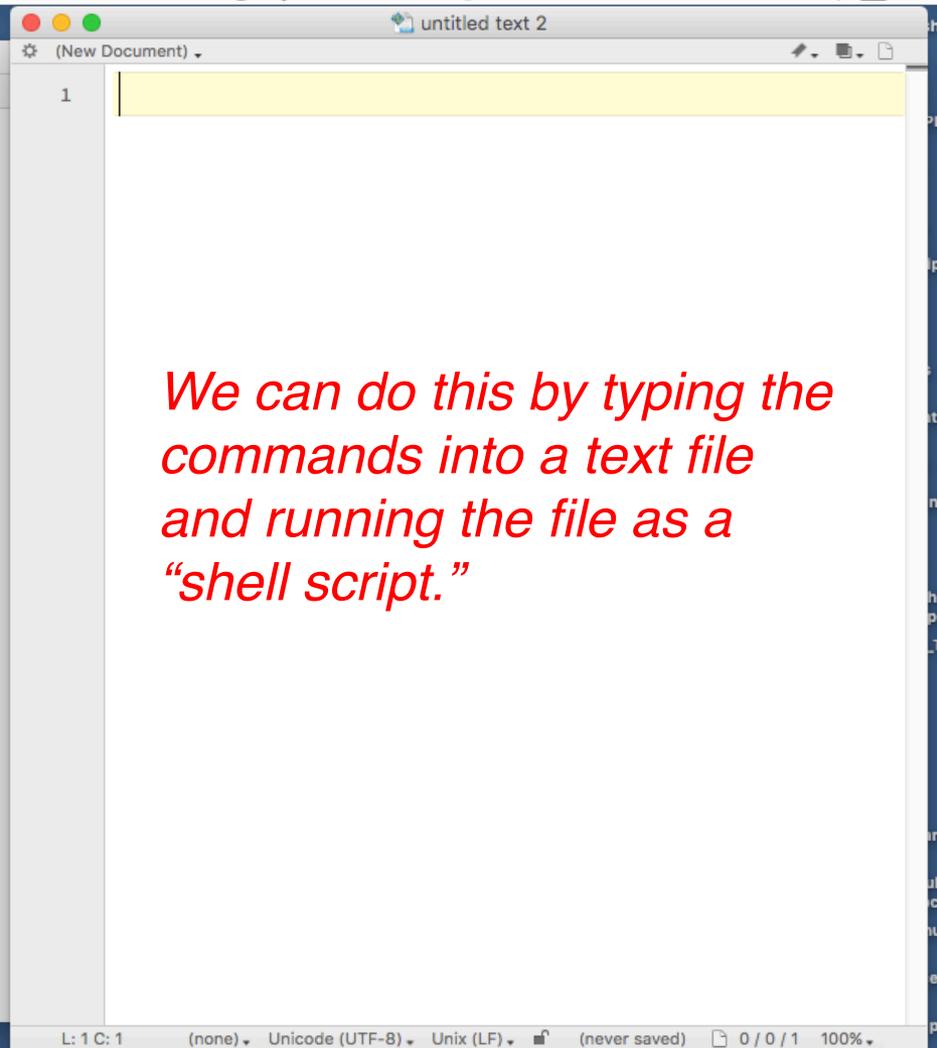
Writing shell scripts

- List commands in a text file, run all at once.



```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $
```

We can do this by typing the commands into a text file and running the file as a "shell script."



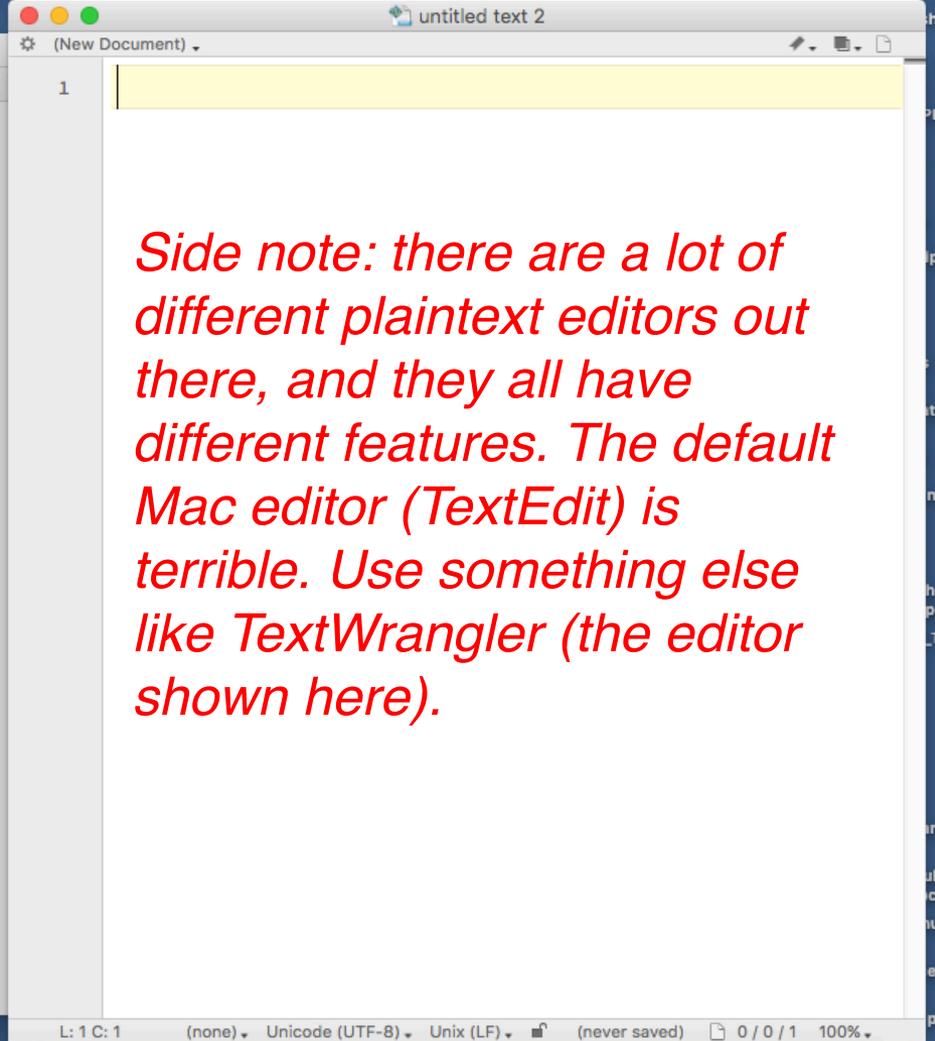
Writing shell scripts

- List commands in a text file, run all at once.

unix_tutorial — -bash — 64x32

~/unix_tutorial — -bash

```
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $
```



Writing shell scripts

- List commands in a text file, run all at once.

unix_tutorial — -bash — 64x32

~/unix_tutorial — -bash

```
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $
```

untitled text 2

(New Document)

1 #!/bin/sh

The first line should always be “#!/bin/sh”. Don’t worry too much about what this means now. Just do it.

L: 1 C: 10 (none) Unicode (UTF-8) Unix (LF) (never saved) 9 / 2 / 1 100%

Writing shell scripts

- List commands in a text file, run all at once.

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $
```

```
untitled text 2
(New Document)
1 #!/bin/sh
2
3 echo 1 2 3 4
4 echo 5 6 7 8
5 echo 9 10 11 12
6 echo 13 14 15 17
```

Type in the commands you want to run, in order.

L: 6 C: 17 (none) Unicode (UTF-8) Unix (LF) (never saved) 69 / 22 / 6 100%

Writing shell scripts

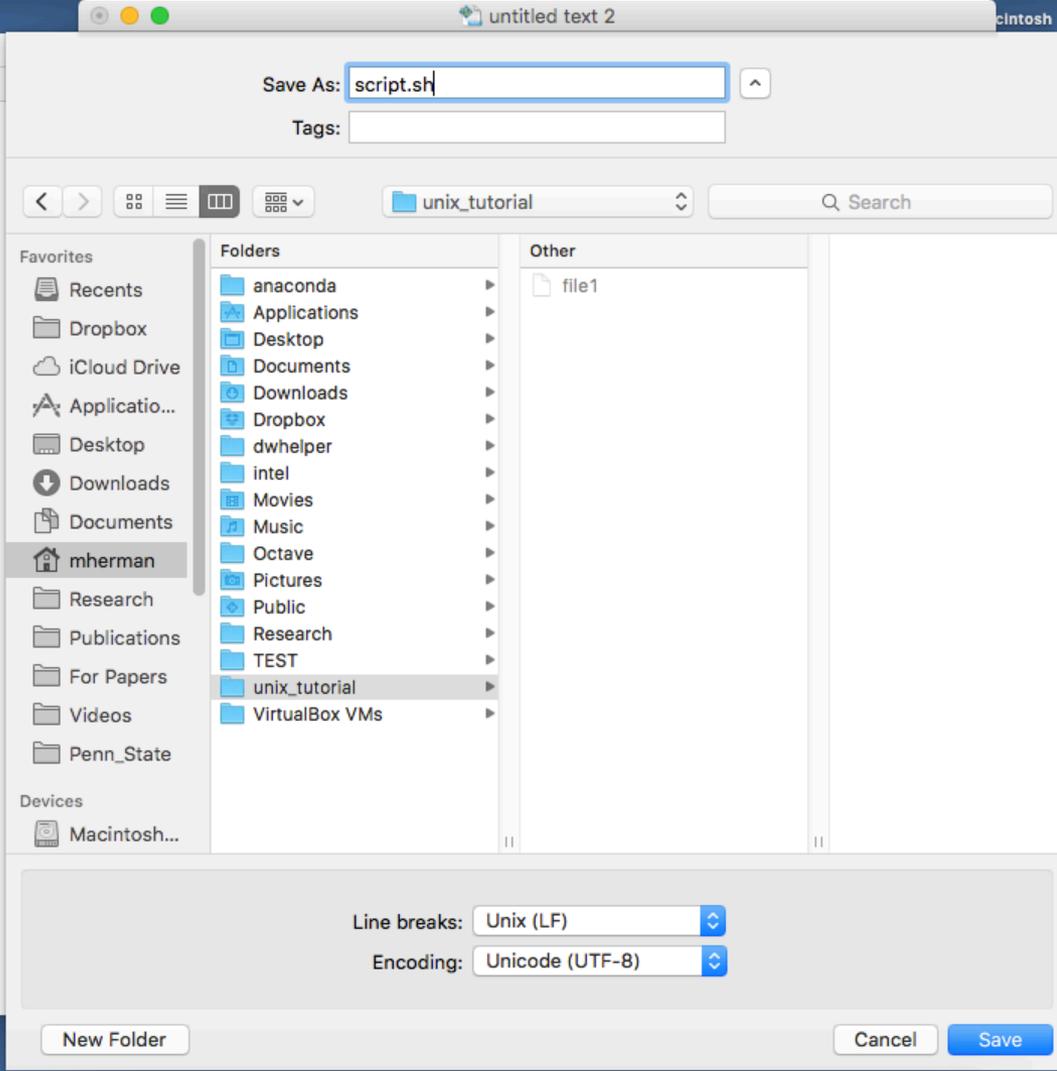
- List commands in a text file, run all at once.

```

unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $

```

Save the file into the folder where you are working. Call it "script.sh"



Writing shell scripts

- List commands in a text file, run all at once.



```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $
```

Return to your terminal, and type in “sh script.sh” (or whatever the name of your script is).

```
unix_tutorial/script.sh
#!/bin/sh
echo 1 2 3 4
echo 5 6 7 8
echo 9 10 11 12
echo 13 14 15 17
```

Li: 6 C: 17 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 10/22/16, 1:36:15 PM

Writing shell scripts

- List commands in a text file, run all at once.

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $
```

Making changes to a sequence of commands is much easier with scripting.

```
script.sh
~/unix_tutorial/script.sh
1  #!/bin/sh
2
3  echo 1 2 3 4
4  echo 5 6 7 8
5  echo 9 10 11 12
6  echo 13 14 15 16
7
```

L: 6 C: 15 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 10/22/16, 1:36:15 PM

Writing shell scripts

- List commands in a text file, run all at once.

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $
```

Making changes to a sequence of commands is much easier with scripting.

```
script.sh
~/unix_tutorial/script.sh
1  #!/bin/sh
2
3  echo 1 2 3 4
4  echo 5 6 7 8
5  echo 9 10 11 12
6  echo 13 14 15 16
7
```

I wrote "17" instead of "16" like I wanted to.

L: 6 C: 15 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 10/22/16, 1:36:15 PM

Writing shell scripts

- List commands in a text file, run all at once.

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $
```

Variables make scripts very powerful and flexible. Use them often.

```
script.sh
~/unix_tutorial/script.sh
1  #!/bin/sh
2
3  echo 1 2 3 4
4  echo 5 6 7 8
5  echo 9 10 11 12
6  echo 13 14 15 16
7
8  echo Matt wants to go to bed
9  VAR1="Matt"
10 VAR2="bed"
11 echo $VAR1 wants to go to $VAR2
12
```

L: 12 C: 1 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 10/22/16, 1:41:43 PM

Writing shell scripts

- List commands in a text file, run all at once.

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $
```

Variables make scripts very powerful and flexible. Use them often.

```
script.sh
~/unix_tutorial/script.sh
1  #!/bin/sh
2
3  echo 1 2 3 4
4  echo 5 6 7 8
5  echo 9 10 11 12
6  echo 13 14 15 16
7
8  echo Matt wants to go to bed
9  VAR1="Matt" ←
10 VAR2="bed"
11 echo $VAR1 wants to go to $VAR2
12
```

Define variables with an "=" (no spaces allowed)

L: 12 C: 1 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 10/22/16, 1:41:43 PM

Writing shell scripts

- List commands in a text file, run all at once.

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $
```

Variables make scripts very powerful and flexible. Use them often.

```
script.sh
~/unix_tutorial/script.sh
1  #!/bin/sh
2
3  echo 1 2 3 4
4  echo 5 6 7 8
5  echo 9 10 11 12
6  echo 13 14 15 16
7
8  echo Matt wants to go to bed
9  VAR1="Matt"
10 VAR2="bed"
11 echo $VAR1 wants to go to $VAR2
12
```

Call variables with a "\$" sign in front of the name.

L: 12 C: 1 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 10/22/16, 1:41:43 PM

Writing shell scripts

- List commands in a text file, run all at once.

```

unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Matt wants to go to bed
Matt wants to go to bed
mherman: unix_tutorial $

```

```

unix_tutorial/script.sh
#!/bin/sh

echo 1 2 3 4
echo 5 6 7 8
echo 9 10 11 12
echo 13 14 15 16

echo Matt wants to go to bed
VAR1="Matt"
VAR2="bed"
echo $VAR1 wants to go to $VAR2

```

Save the file!
Run it again!



Writing shell scripts

- List commands in a text file, run all at once.

unix_tutorial — -bash — 64x32

~/unix_tutorial — -bash

```
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Matt wants to go to bed
Matt wants to go to bed
mherman: unix_tutorial $
```

script.sh

~/unix_tutorial/script.sh

```
1  #!/bin/sh
2
3  echo 1 2 3 4
4  echo 5 6 7 8
5  echo 9 10 11 12
6  echo 13 14 15 16
7
8  echo Matt wants to go to bed
9  VAR1="Frodo"
10 VAR2="Mt. Doom"
11 echo $VAR1 wants to go to $VAR2
12
```

*Try changing the variables,
saving the file, and running
again.*

L: 9 C: 12 Unix Shell Script Unicode (UTF-8) Unix (LF) Saved: 10/22/16, 1:47:11 PM

Writing shell scripts

- List commands in a text file, run all at once.

```
unix_tutorial — -bash — 64x32
~/unix_tutorial — -bash
mherman: unix_tutorial $ echo 1 2 3 4
1 2 3 4
mherman: unix_tutorial $ echo 5 6 7 8
5 6 7 8
mherman: unix_tutorial $ echo 9 10 11 12
9 10 11 12
mherman: unix_tutorial $ echo 13 14 15 17
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 17
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Matt wants to go to bed
Matt wants to go to bed
mherman: unix_tutorial $ sh script.sh
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
Matt wants to go to bed
Frodo wants to go to Mt. Doom
mherman: unix_tutorial $
```

```
script.sh
#!/bin/sh

echo 1 2 3 4
echo 5 6 7 8
echo 9 10 11 12
echo 13 14 15 16

echo Matt wants to go to bed
VAR1="Frodo"
VAR2="Mt. Doom"
echo $VAR1 wants to go to $VAR2
```

Try changing the variables, saving the file, and running again.



Writing shell scripts

- List commands in a text file, run all at once.

Tutorial Complete

- You now have enough basic knowledge to begin learning GMT through our GMT tutorials.
- Remember, this is just the beginning. There is so much more to the Unix operating system and the more you know, the more cool things you can accomplish.

Good luck!