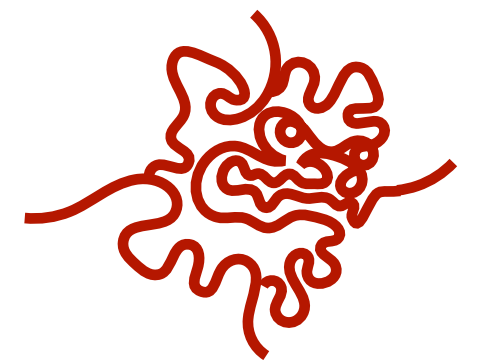


# miniCourse: Terminal

---

## Part I: Introduction

Ryo Nakatani ([ryo.nakatani@oist.jp](mailto:ryo.nakatani@oist.jp))



**OIST**

Okinawa Institute of Science and Technology  
Graduate University

沖縄科学技術大学院大学

# Course Outline

---



## 1. Terminal (Day 1)

- Introduction
- Basics
- Advanced

## 2. Vim (Day 2)

➡ Jeremie Gillet

## 3. High-Performance Computing (Day 3)

➡ Charles Plessy

# Introduction

3 / 38



## What is a terminal?

A terminal is a text only window that can be used as a **Command Line Interface (CLI)** on a **Graphical User Interface (GUI)** based computer

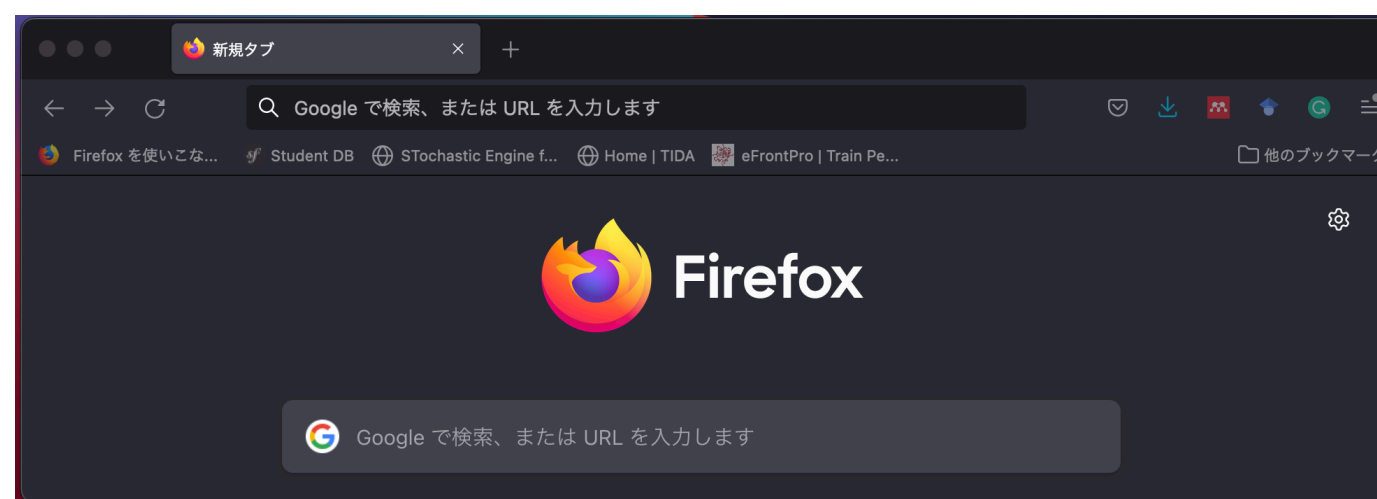
### Graphical User Interface (GUI)



### Command Line Interface (CLI)

```
% open /Applications/Firefox.app
```

opens Firefox



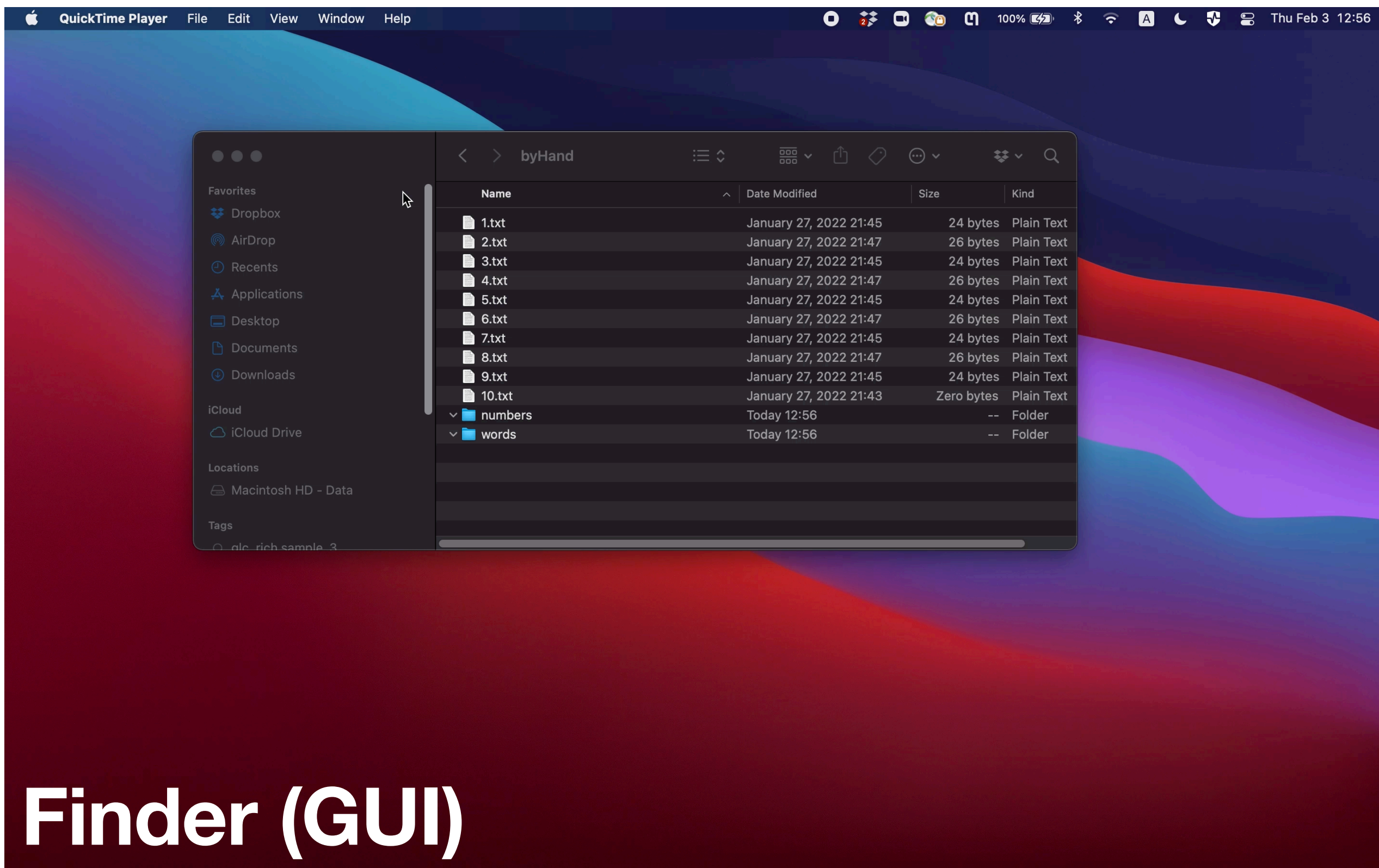
# Introduction

4 /38

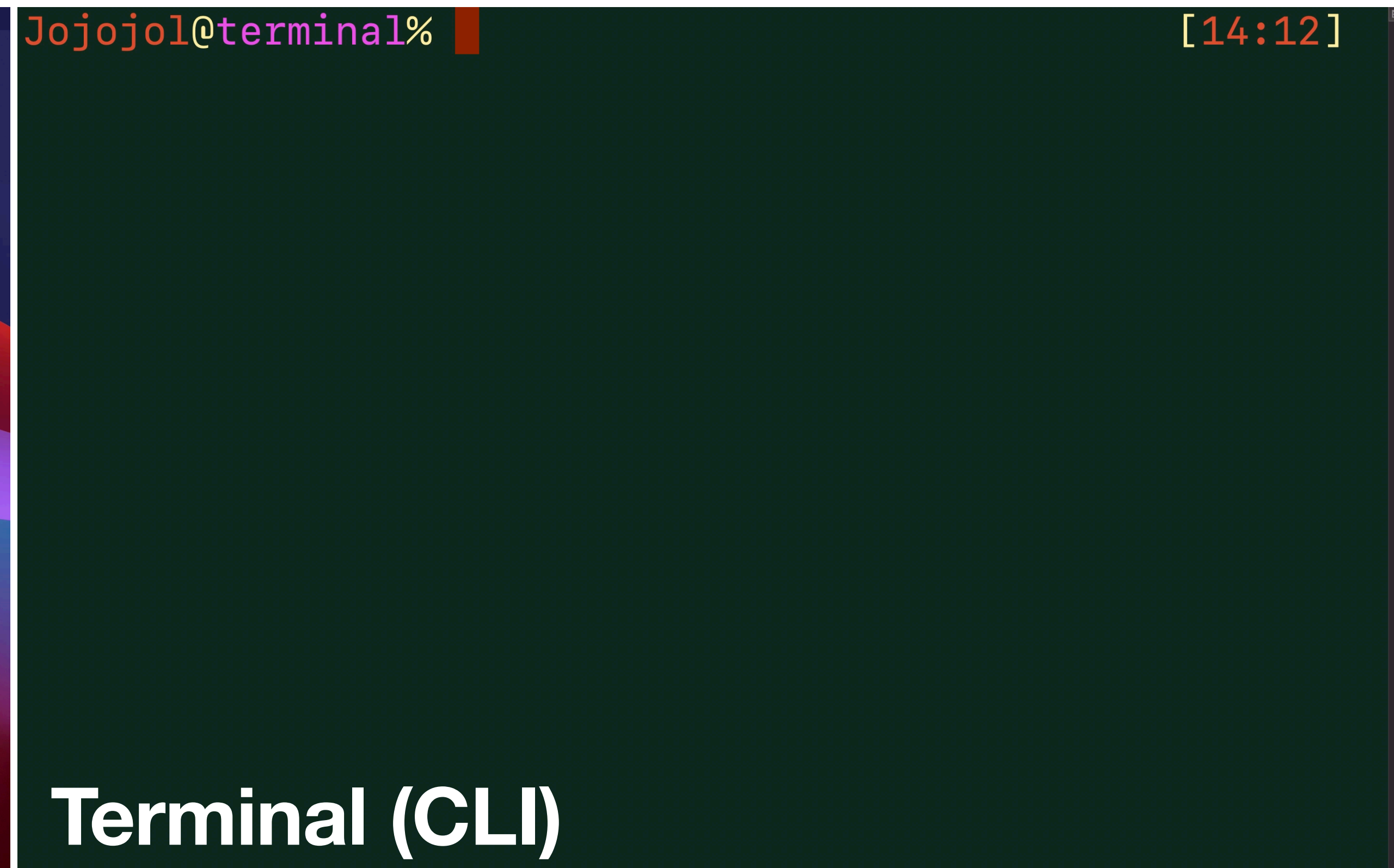


## Terminals are useful

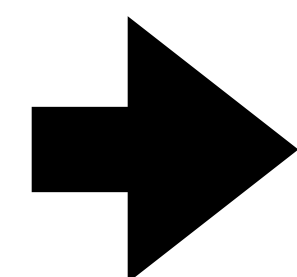
Task: Classify 10 files into folders depending on contents



**Finder (GUI)**



**Terminal (CLI)**



Same task, better faster stronger



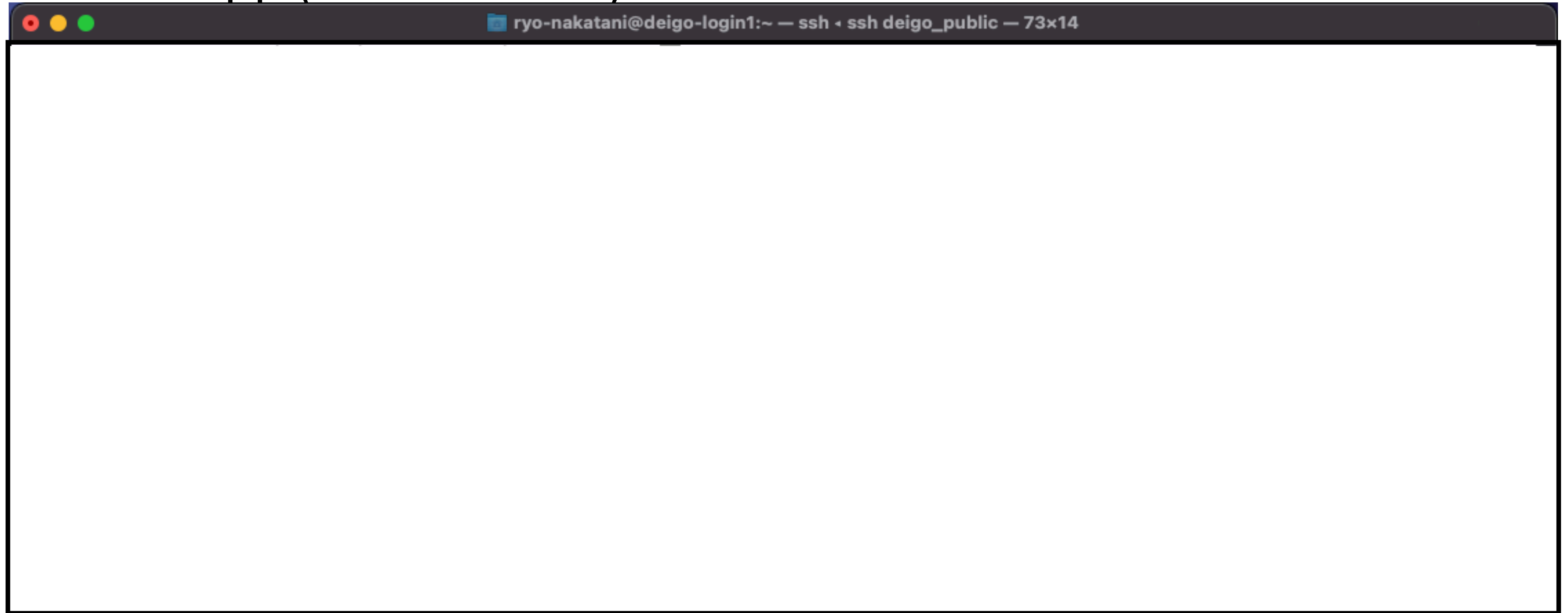
# Introduction

5 /38



## Parts of the terminal screen

Terminal.app (MacOSX GUI)



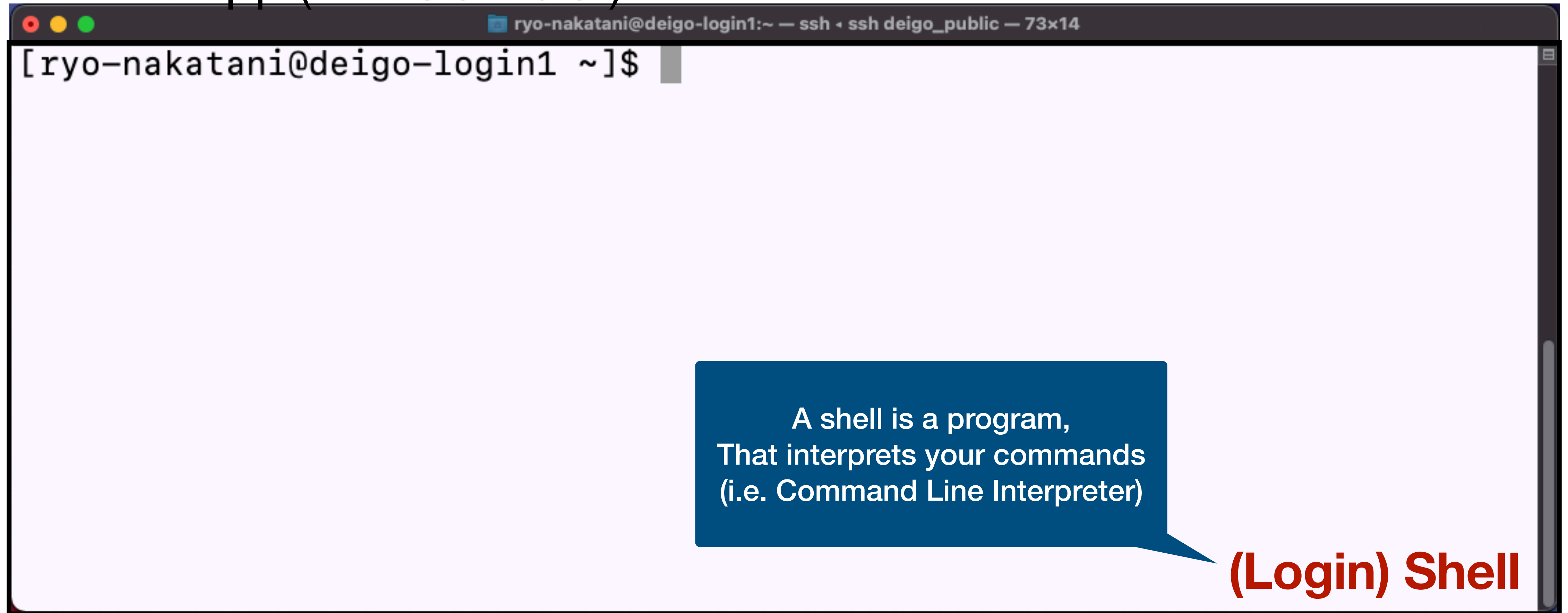
# Introduction

6 /38



## Parts of the terminal screen

### Terminal.app (MacOSX GUI)



# Introduction

7 /38



## Parts of the terminal screen

Terminal.app (MacOSX GUI)

Cursor

```
[ryo-nakatani@deigo-login1 ~]$
```

Prompt (Username @ Machine name directory) \$

**(Login) Shell**

# Introduction

8 / 38



## Parts of the terminal screen

### Terminal.app (MacOSX GUI)

A screenshot of a Terminal window titled "ryo-nakatani@deigo-login1:~ — ssh • ssh deigo\_public — 73x14". The prompt is "[ryo-nakatani@deigo-login1 ~]\$". A large blue callout box explains the components of a command:

- Commands**  
Instruction for the computer to do something
- Option(s)**  
Modifications to the command
- Argument(s)**  
Input to the command

To the right, three labels with callout lines point to parts of the command "\$ say --voice=Alex 'hello'":

- Command** (red text) points to "\$"
- Option(s)** (green text) points to "--voice=Alex"
- Argument(s)** (blue text) points to "'hello'"

The command is displayed on a black background within the blue callout box.

**(Login) Shell**



# Introduction

9 / 38



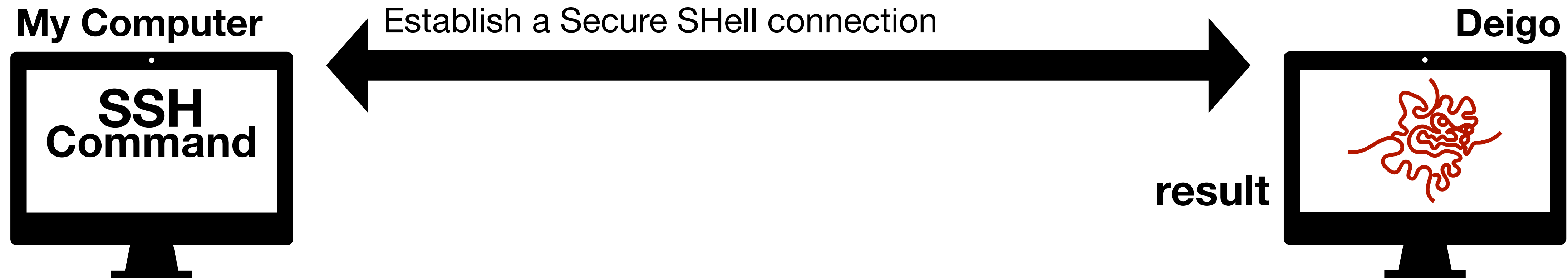
## Connecting to Deigo (SSH)

### What is ssh?

ssh is a way to connect your local terminal to another computer

### Why ssh?

You can use external computer resources or get remote files in a secure fashion



# Introduction

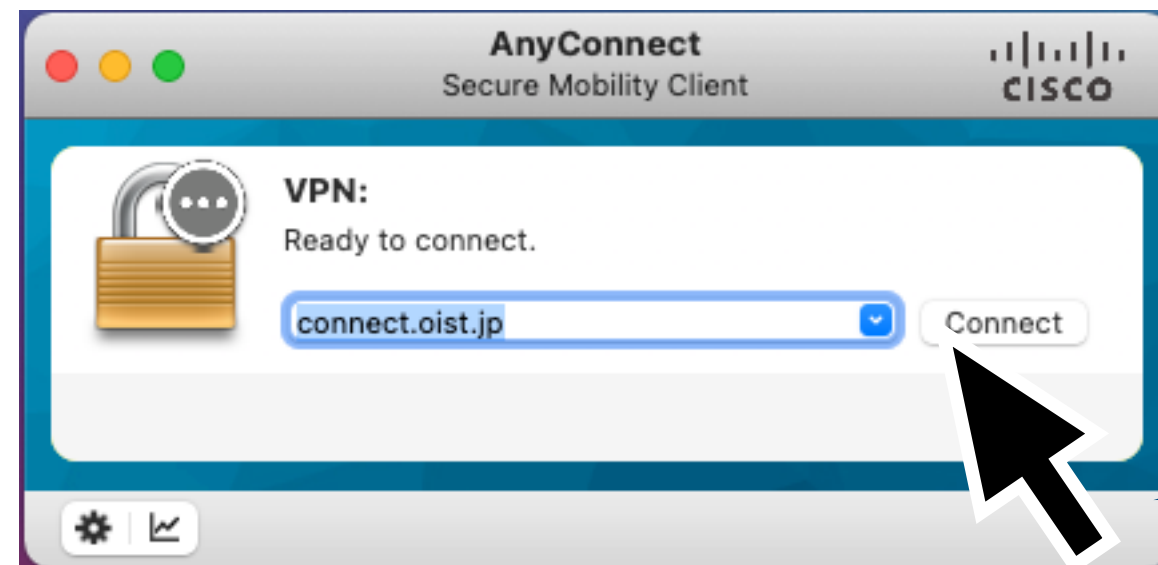
## Connecting to Deigo (SSH)

10/38



### Not on campus? (Probably)

In order to connect to deigo you must be on campus or use VPN



Use the Cisco AnyConnect Secure Mobility Client  
(For Installation guide search for VPN on IT service portal)

VPN access can be requested from the IT portal (<https://services.oist.jp/sp>)

# Introduction



## Connecting to Deigo (SSH)

Now let's connect to deigo using the ssh command

The command to enter is... (Linux/Mac/Windows MobaXterm)

```
% ssh -X user-name@deigo.oist.jp
```

**Command**

**Option(s)**

**Argument(s)**

### Commands

Instruction for the computer to do something

### Option(s)

Modifications to the command

### Argument(s)

Input to the command

## Execution Guide

Please try on your computer!

```
% command to try out
```

No need to try now

```
% command not to try out
```

# Introduction

12/38



## Connecting to Deigo (SSH)

To disconnect from deigo...

```
% logout
```

End login shell

```
% [ctrl] - D
```

End input to shell (End shell)

```
% exit
```

Exit any kind of shell

If you disconnected,

Try connecting back to deigo on your own!

## Execution Guide

Please try on your computer!

```
%ssh -X user-name@deigo.oist.jp
```

No need to try now

```
% command not to try out
```



## Tips and Tricks “Entering Commands”

- Ways to enter commands robustly and quickly

Keys to use	Function	Example
[tab]	Finish the rest of your command	logo[tab] —> logout
[up/down arrows]	Select previous commands	[up] —> logout

- Ways to stop when your screen is “frozen”

Keys to use	Function	Example
[ctrl]-C	Kill command	Exercise On Next Slide —>
[ctrl]-D	EOF (end input)	
[ctrl]-Z	Suspend command	





# Exercise

## Tips and Tricks “Entering Commands”

```
$ cat
```

Try the following

[Ctrl]-C

[Ctrl]-Z

[Ctrl]-D

```
$ man cat
```

Try the following

[Ctrl]-C

[Ctrl]-Z

[Ctrl]-D



### TIP

If you used [Ctrl]-Z to suspend,  
The “fg” command will revive it

## Execution Guide

Please try on your computer!

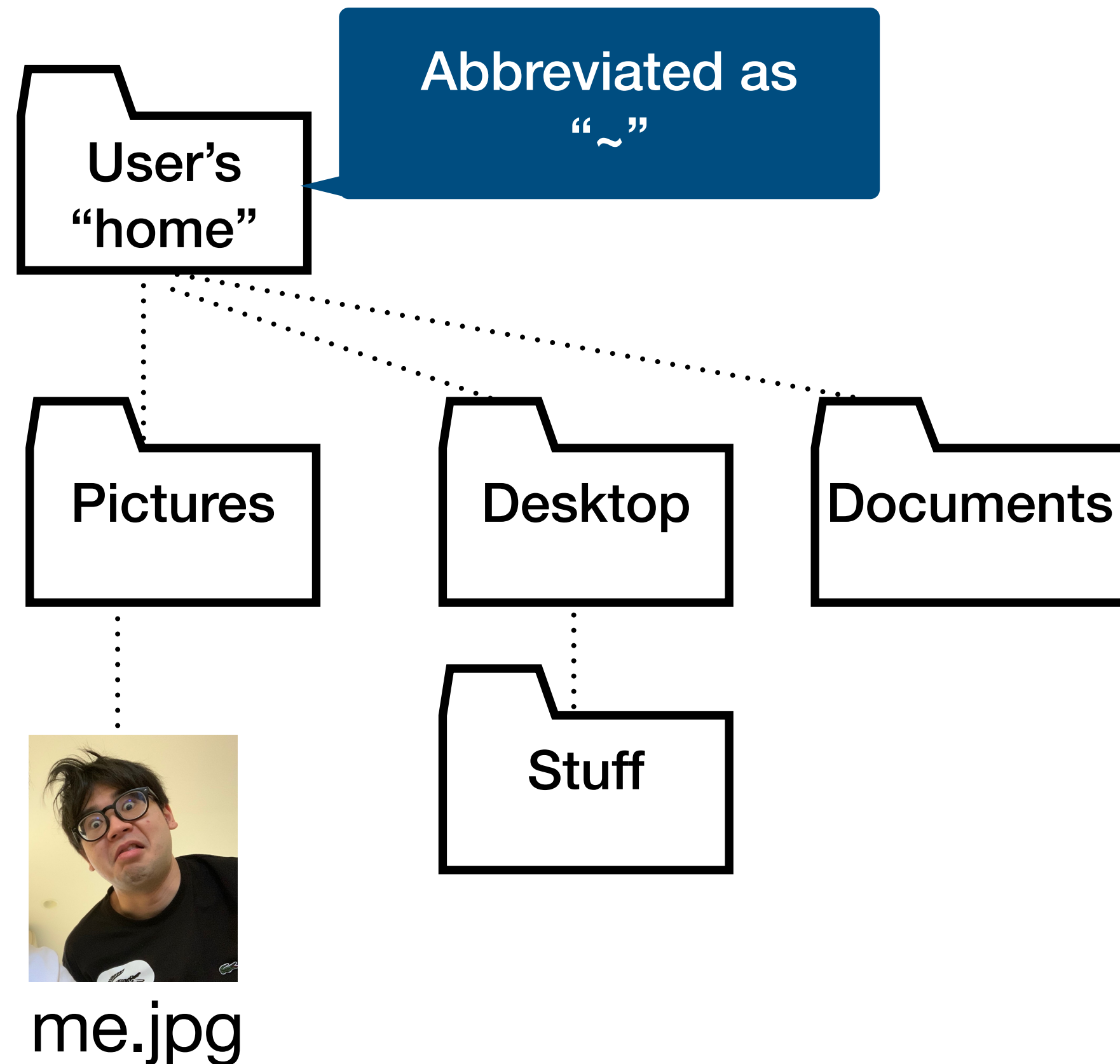
```
% command to try out
```

No need to try now

```
% command not to try out
```

## Directory Structure

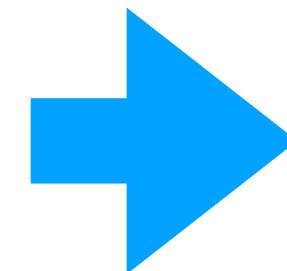
Your files (and all others) are stored in the computer as a directory structure



On a command line if I want to open the image (Mac)

```
% open ~/Pictures/me.jpg
```

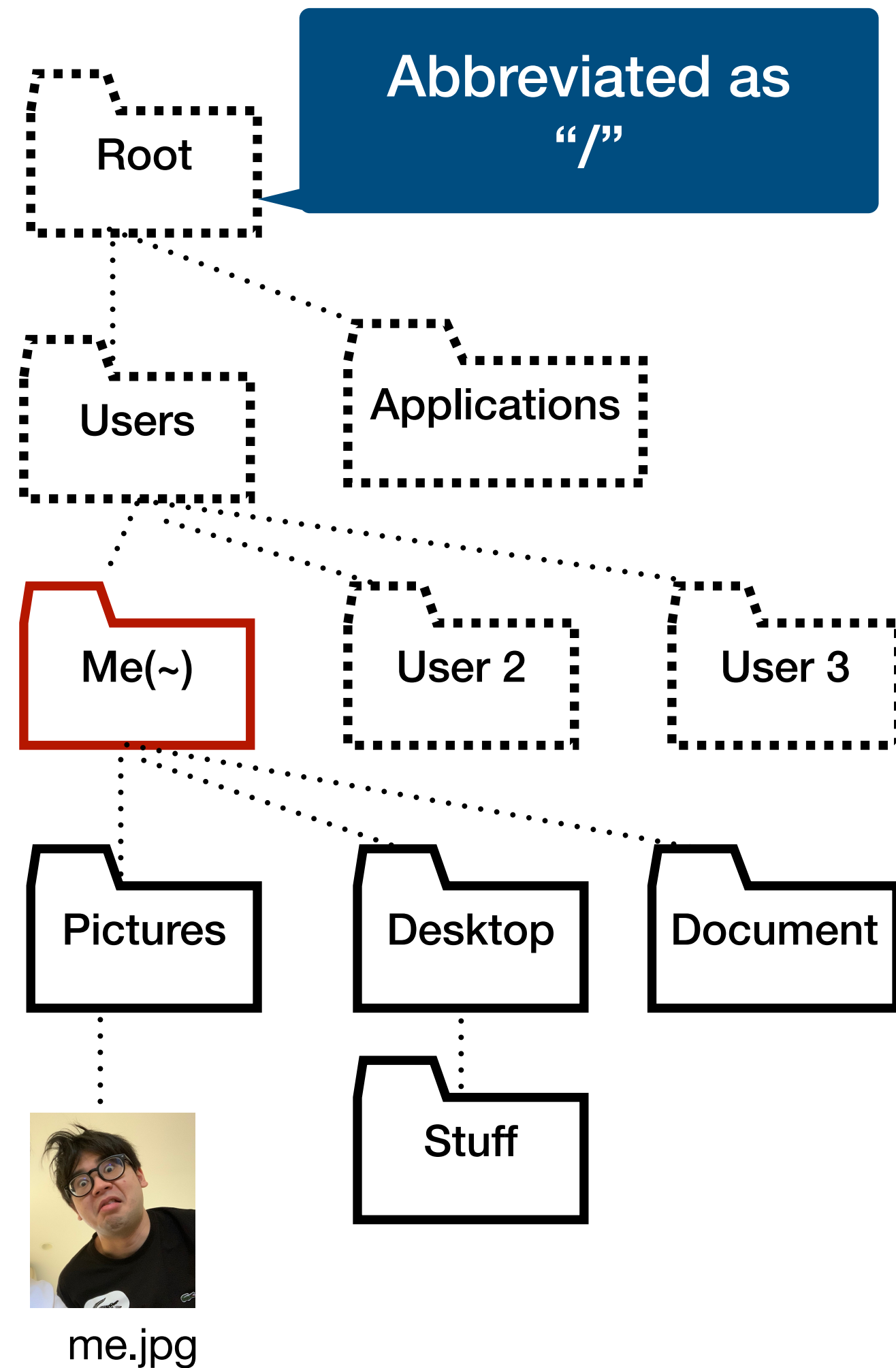
Different folders (directories) are expressed with /



When you login to deigo, you start at home

## Directory Structure

There are other users and files outside your “home”



The home directory for “Me” is equivalent to...

- /Users/Me
- ~

## Directory Commands

- Where am I?

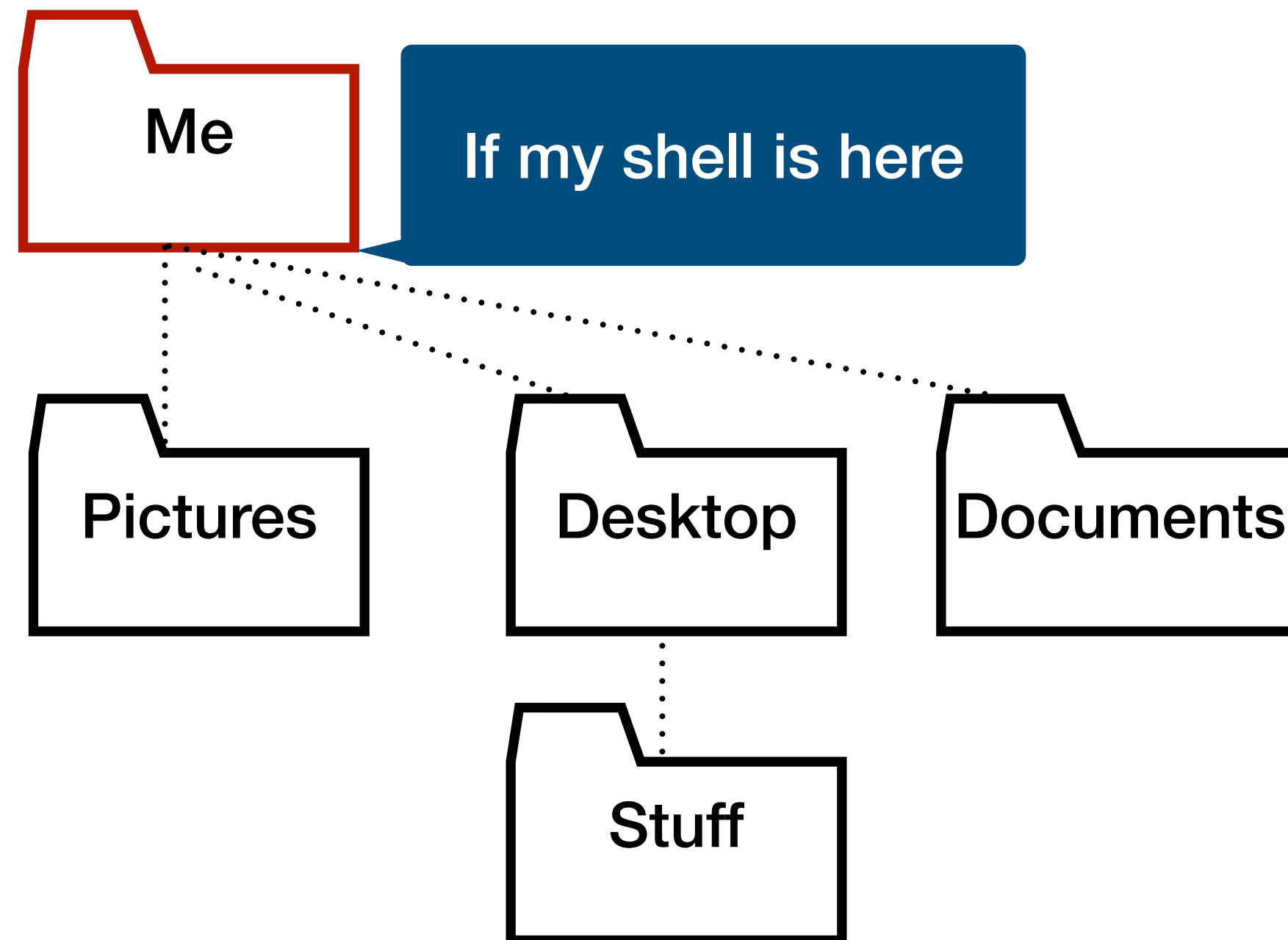
```
% pwd
```

Print the current location the shell is in

```
% ls
```

Print the directories/files in the current location the shell is in

Visual Example)



```
% pwd
/Users/Me
% ls
Pictures  Desktop  Documents
```

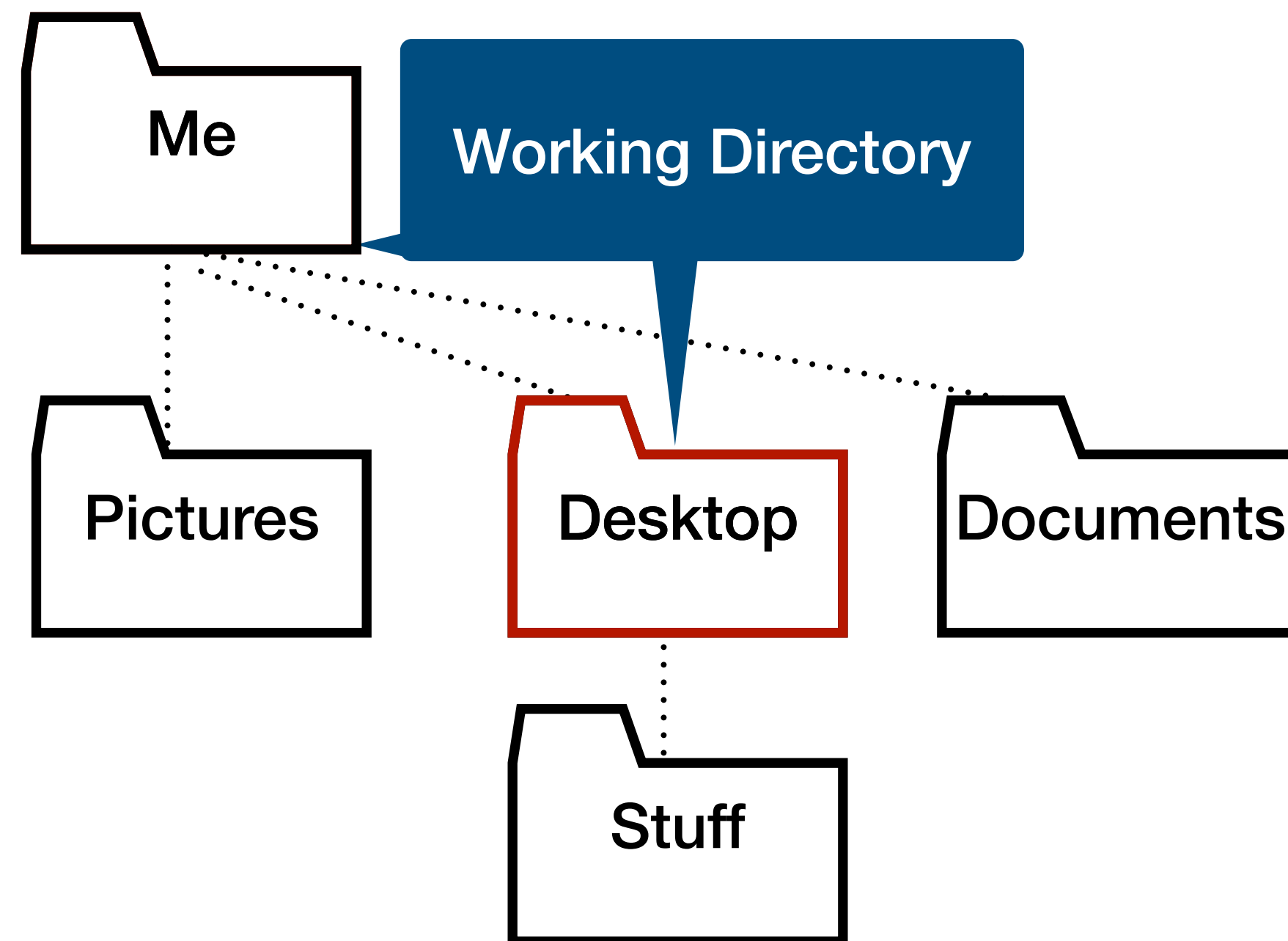
## Directory Commands

- Moving about

```
% cd /Directory/I/will/move/to
```

Will move the shell's working directory (current directory)  
To /Directory/I/will/move/to

Visual Example)



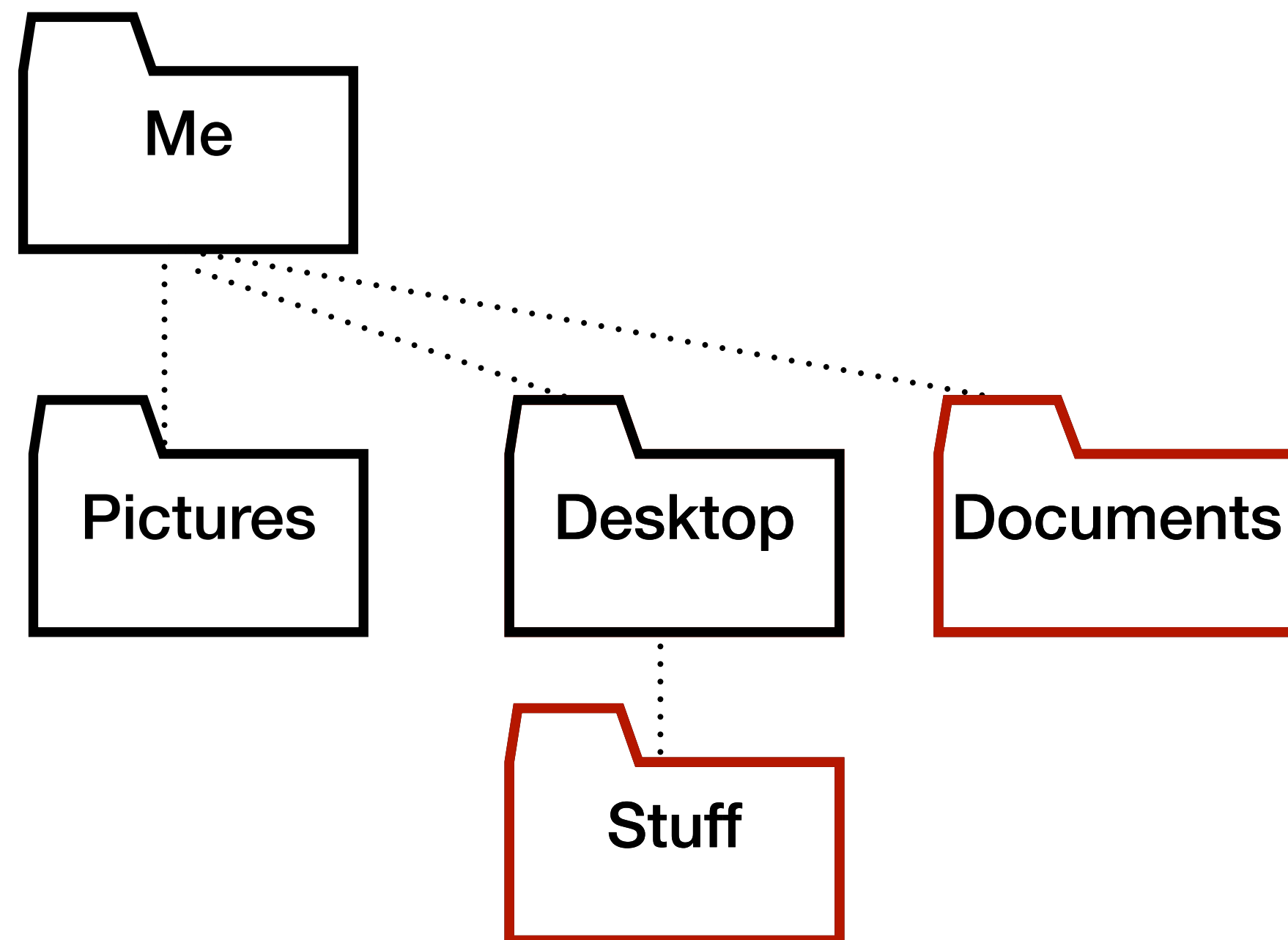
```
% pwd  
/Users/Me  
% cd /Users/Me/Desktop  
% ls  
Stuff
```



## Directory Commands

Like ~ (home directory) you can use `‘.’` and `‘..’` to specify current working directory(`‘.’`) and the directory above(`‘..’`)

Visual Example)



```
% pwd
/Users/Me/Desktop
% cd ./Stuff
% cd ../../Documents
```

Absolute path

Writing all directories from root

Relative path

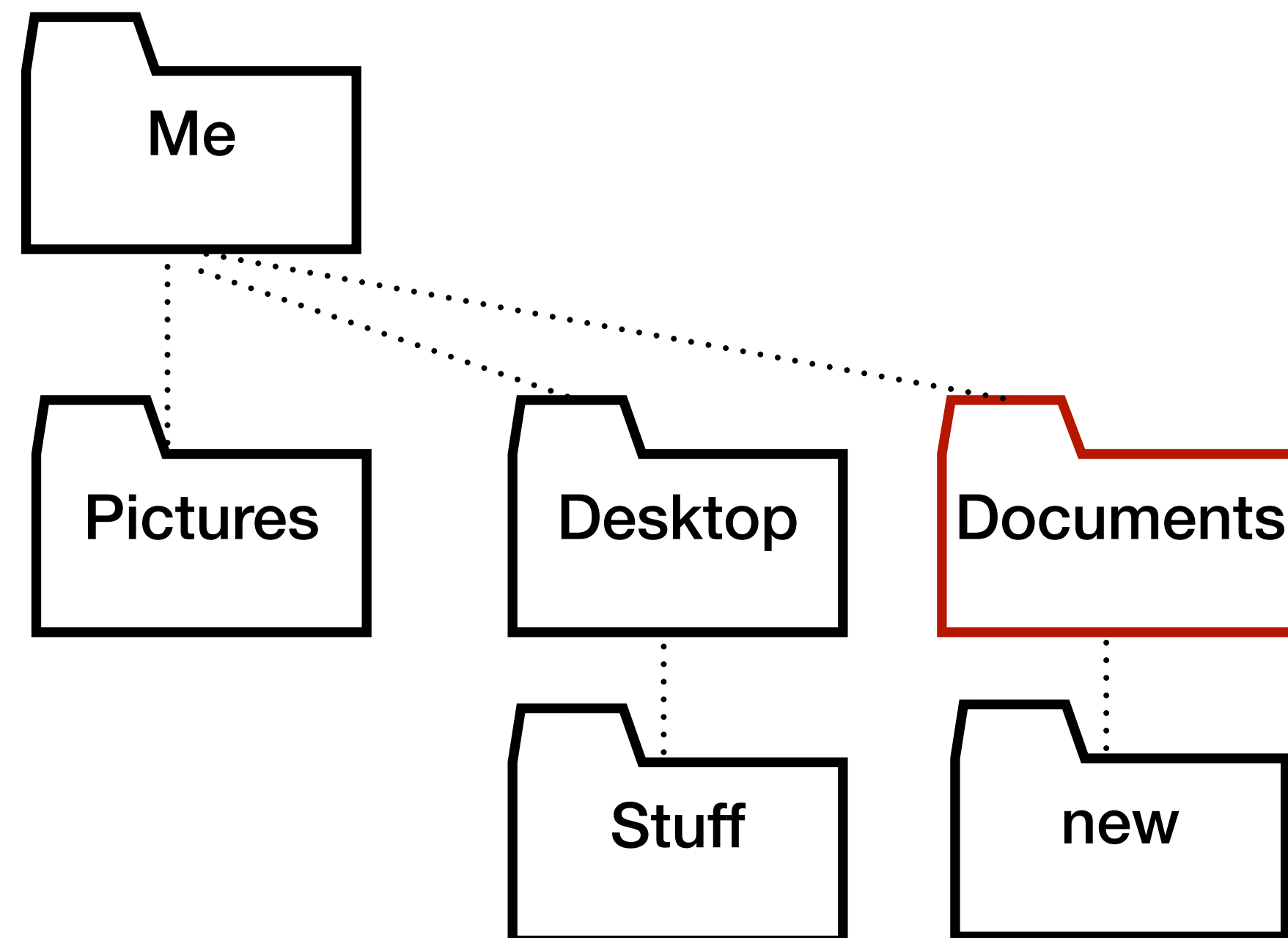
Writing directories using `“.”` And `“..”`

## Directory Commands

You can also make new directories

```
% mkdir ./newDirectory
```

Visual Example)



```
% pwd  
/Users/Me/  
% cd Documents  
% mkdir new
```

# Exercise

## Directory Commands

- Check where you are, go to the root, and see what's there

```
% pwd
% cd /
% ls
```



### TIP

pwd → cd → ls → pwd ...

- Check where you are, if you are not home go back home

```
% cd ~
- or -
% cd $HOME
- or -
% cd
- or -
% cd -
```

The last one is go back to the previous directory

Directory	Abbrev.
Home	~
Root	/
Current Directory	.
Parent Directory	..

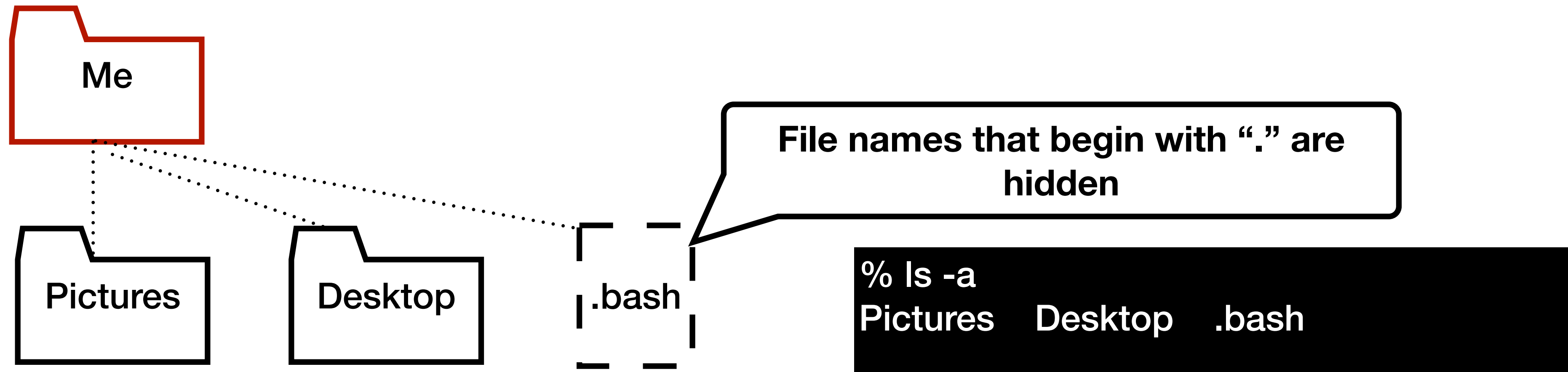
# Coffee Break

22/38



## Did you know (Hidden Files)?

Files that don't usually show up with ls, and on GUI exist in your computer



Usually these files are configuration files (settings) so don't touch them unless you know what you are doing



## Tip and Tricks “Terminal Lingo”

Special characters used in terminal

Character	Function	Example
*	All	<div>*.txt : all .txt files in your pwd</div> <div>N* : all files starting with N in your pwd</div>
\$	Variable	<div>\$HOME: variable with the abs path of ~</div> <div>\$SHELL: variable with the name of your shell</div>
“”	Treat as Text	<div>“Hello”: “Hello” is text, not name of a file or path</div> <div>“\$HOME”: Contents of \$HOME is treated as text</div>
‘’	Treat as Raw Text	<div>‘Hello’: ‘Hello’ is text, not name of a file or path</div> <div>‘\$HOME’: ‘\$HOME’ is text not variable</div>
\	Treat as Normal character	<div>\\$: treated as dollar sign not special variable</div> <div>\*: treated as asterisk not special character</div>



Often there are things that you want to do with your file, directory, etc...

- Copying
- Moving
- Looking
- Removing
- Writing

## Copying files

Copying files from one directory to another (2 arguments necessary)

```
% cp [This] [toThere]
```

To move directories,  
You must use -r option

Copying exercise directory to your home

```
% cp /apps/unit/GradschoolD/terminal ~  
% cp -r /apps/unit/GradschoolD/terminal ~
```

The first one should give you  
an error message



Carelessly using cp will overwrite all files permanently,  
use -i option for the command to always ask before overwriting

## Moving files

Moving files from one directory to another (2 arguments necessary)

```
% mv [This] [toThere]
```

**No -r option necessary,  
This will move files and directories**

Make a new directory in home and move terminal directory there

```
% pwd  
% mkdir Dir  
% mv terminal Dir  
% mv Dir newDir
```



Carelessly using mv will overwrite all files permanently,  
use -i option for the command to always ask before overwriting



## Looking at files/directories

Often times you want to check the content of the file/directory or even whether it is a file or a directory

### Checking file types

```
% ls -l
```

First 4 letters define type

d: Directory

r: readable (You can check it)

w: writable (You can modify it)

x: executable (You can run it; its a program)

Example)

Owner

File name

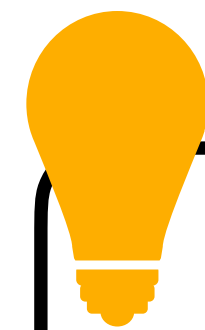
```
% ls -l
drwxr-xr-x 3 ryo-nakatani allstudents 26 Feb 4 01:39 directory
-rw-r--r-- 1 ryo-nakatani allstudents 4 Feb 4 01:48 file.txt
-rwxr-xr-x 1 ryo-nakatani allstudents 4 Feb 4 01:48 code.py
```

## Looking at files/directories

Often times you want to check the content of the file/directory or even whether it is a file or a directory

### Checking contents of a directory

```
% ls -a ~  
% ls -r ~  
% ls -l ~  
% ls -l -t ~  
% ls -lth ~  
% ls -arlth ~
```



#### TIP

Interesting options for ls...

- a : Show hidden files
- l : Show long list
- r : Also show contents of subdirectories
- t : Show in chronological order
- h : Show file sizes with units (use with -l)





## Looking at files/directories

Often times you want to check the content of the file/directory or even whether it is a file or a directory

### Checking contents of a text file (cat)

Let's check the contents of

~/newDir/terminal/materials/cat\_test/\*.txt

- change directory to ~/newDir/terminal/materials/cat\_test
- Examine the names of files using ls
- Check the contents of each one

```
% cat CATchMEifYOUcan.txt  
% cat catmultiple1.txt  
% cat catmultiple2.txt  
% cat catmultiple*
```

## Looking at files/directories

Often times you want to check the content of the file/directory or even whether it is a file or a directory

### Checking contents of a text file (less)

Let's check the contents of  
~/terminal/materials/less\_test/AliceInWonderland.txt

```
% cat ~/terminal/materials/less_test/AliceInWonderland.txt  
% less ~/terminal/materials/less_test/AliceInWonderland.txt
```



#### TIP

While cat spits out everything as output,  
Less makes it possible to interact with the text

Keys to use	Function	Keys to use	Function
[up/down arrows]	Scroll through the text	/words	Find words in text
h	Get help screen	[space]	Next page
q	Quit less	b	Back one page

## Removing files and folders

removing files

```
% rm [This] ...
```

The -r option will erase  
all files and the directory

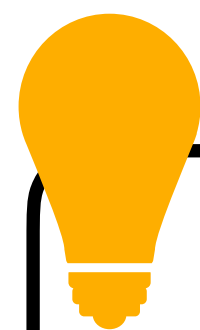
removing directories

```
% rmdir [This] ...
```

Will remove empty directories



Carelessly using rm will remove all files permanently,  
use -i option for the command to always ask before removing each file



**TIP**

As a failsafe when executing `% rm *` it makes you wait ten seconds.  
But to be on the safe side, do not use \* in rm commands.

## Removing files and folders

### removing files



Carelessly using `rm` will remove all files permanently, use `-i` option for the command to always ask before removing each file

- change directory to `~/newDir/terminal/materials/remove_test`
- Examine the names of files using `ls`
- Remove the file if it is empty (check with `less` or `cat` to make sure)

```
% rm -i [fileName]
```

## Writing to a file

We will mainly do this in tomorrow's lecture!!

You can write outputs to text files using redirection

**Write** Hello World in output.txt (will overwrite)

```
% echo "Hello World" > output.txt
% cat output.txt
Hello World
```

**Append** Hello World in output.txt

```
% echo "Hello World" >> output.txt
% cat output.txt
Hello World
Hello World
```



### TIP

“echo” command prints text and contents of variables.

It is like the print in programming languages.

Ex) echo “Hello World” —> Hello World  
echo \$USER —> your-user-name

## Useful Commands (text manipulation)

| (pipe)

Pass output of one command to the next

```
% cat "Hello" | cat
```

grep

Catch certain patterns in text

```
% grep 'Hello' *  
% ls | grep .txt
```

awk

Modify text (uses awk language)

```
% ls -l | awk '{print $1}'
```



## Useful Commands (text outputs)

### wc

**Get number of words/lines of a file**

```
% wc someText.txt  
% wc -l someText.txt
```

Count number of words,  
Or with -l option the lines

### head

**Get first 10 lines of text**

```
% head -5 someText.txt
```

With -[number] option  
You can change the number of  
Lines to be shown  
Ex) -1, -3, -50

### tail

**Get last 10 lines of text**

```
% tail someText.txt  
% head -5 someText.txt | tail -4
```

# Advanced



## Useful Commands

**pbcopy (Mac), xclip(Linux), clip(Windows)**

**Copy to clipboard**

```
% cat someText.txt | pbcopy
```

Copy the contents of text file to  
clipboard

**open (Mac/Linux), xdg-open(windows)**

**Open GUI for directory, file, application etc.**

```
% open .
```

Open current directory in finder

# Review



## Commands to know

Commands	Function
[ctrl]-C	Kill command
[ctrl]-D	EOF (end input)
[ctrl]-Z	Suspend command
pwd	Print present working directory
ls	List files in present working directory
cd	Change Directory
cp	Copy files or directory
mv	Move file or directory
rm	Remove file or directory
cat	Output file contents
less	View file contents interactively



# Can't Remember?

## Don't google, try man

Its obvious that you won't remember everything,

In that case use the -h,--help option or the “man” command (short for manual)

```
% man ls  
% man cat
```



### TIP

Same keys as less to interact

Keys to use	Function	Keys to use	Function
[up/down arrows]	Scroll through the text	/words	Find words in text
h	Get help screen	[space]	Next page
q	Quit man	b	Back one page