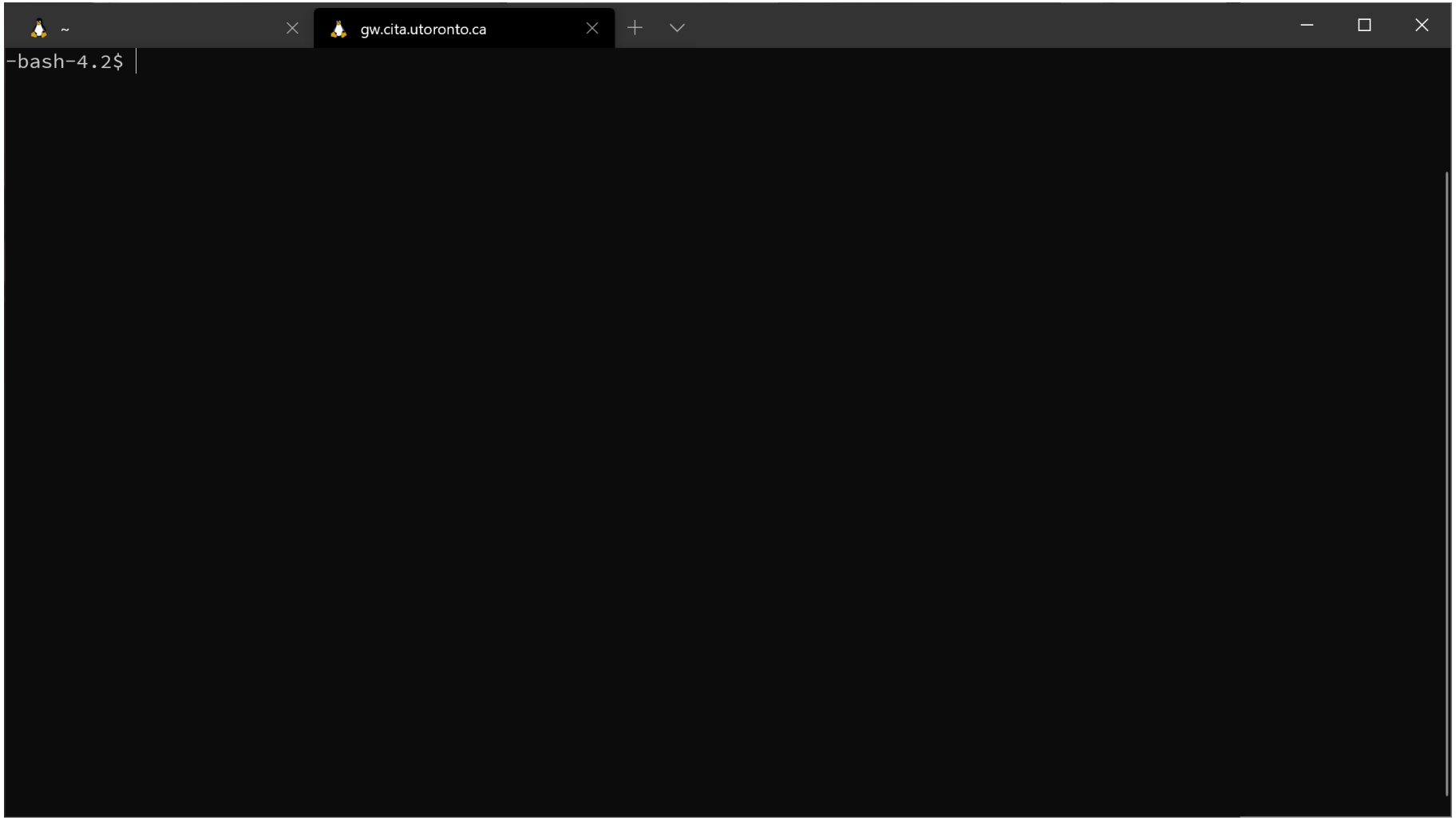
Week 1: Scratching the Surface

SESSION 2: THE COMMAND LINE

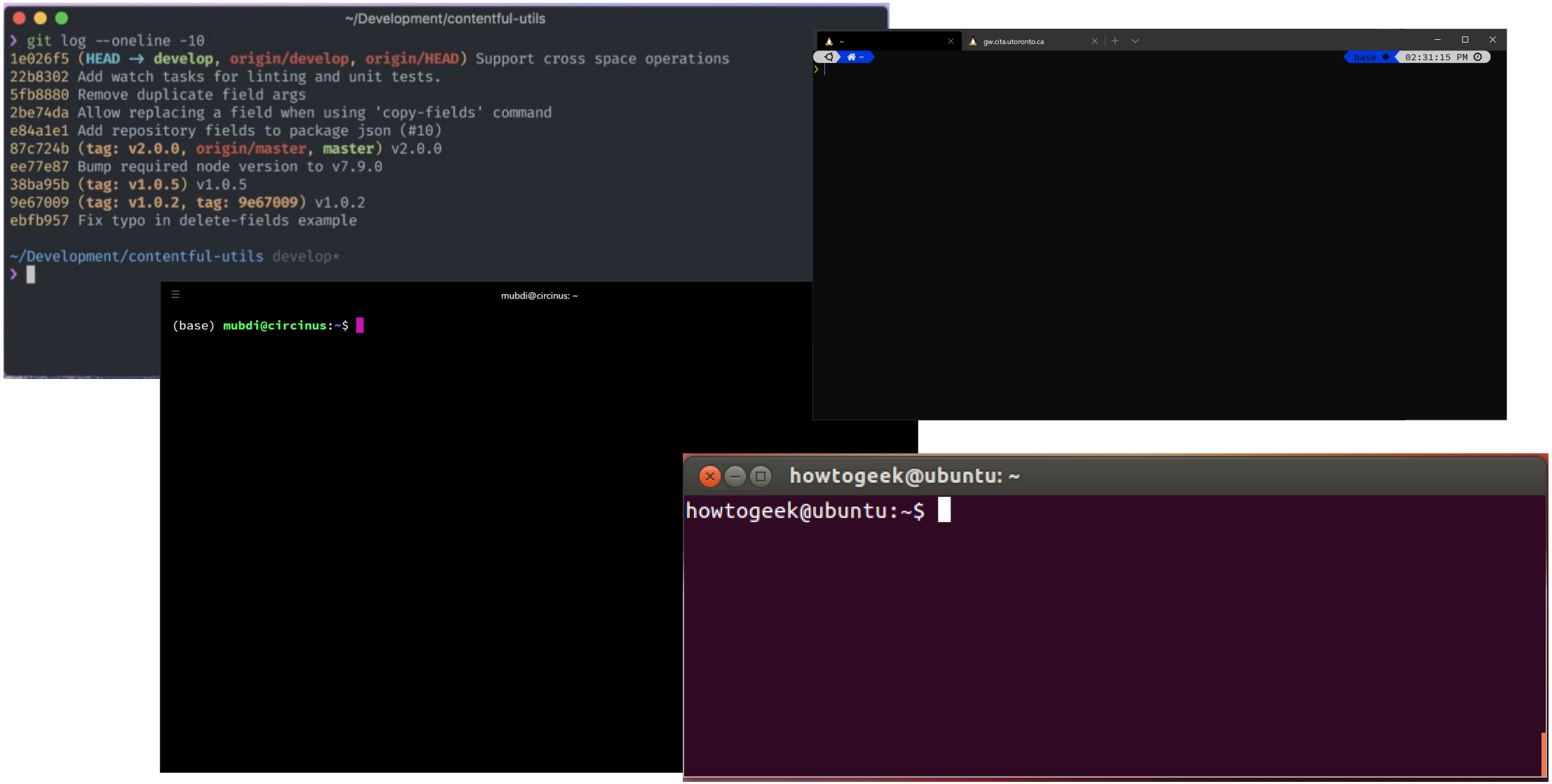
STARFISH SCHOOL 2021

Your friend, the Terminal

This is your terminal. You will come to love this little box!



They come in all sorts of shapes and sizes



What is a Terminal?

A Terminal Program (sometimes also called a “console”) is a way of interacting directly with your computer using text commands. This is an alternative way of interacting with your computer to a mouse, and often, more powerful.

Not all terminals are built the same, and we have the following recommendations for terminal programs:

- **For Windows:** Windows Terminal (available on the Windows Store)
- **For MacOS:** iTerm2 (available on its website: https://iterm2.com)
- **For Linux/or Crossplatform:** Hyper (available on its website: https://hyper.is/)

What is a Terminal?

A Terminal Program (sometimes also called a “console”) is a way of interacting directly with your computer using text commands. This is an alternative way of interacting with your computer to a mouse, and often, more powerful.

Not all terminals are built the same, and w r terminal programs:

- **For Windows:** Windows Terminal (availabl
- **For MacOS:** iTerm2 (available on its websi
- **For Linux/or Crossplatform:** Hyper (availabl https://hyper.is/)

HOT TIP

The terminal programs we’re recommending all have the ability to open multiple tabs and panes, and hot keys to switch between them. This makes it easy to compare two things side-by-side.

Some terminal basics...

```
> cd <directory name> # change your current directory to <directory name>

> cd ~ # change your current directory to your home directory

> ls # list all the files in the current directory

> ls -ahl # the same, but show me all the files in a list

> ls <directory name> # list all the files in <directory name>

> mkdir <directory name> # make a directory called <directory name>

> rmdir <directory name> # remove the directory called <directory name>
(only works if it's empty)

> pwd # list what directory you're in
```

Some terminal basics...

```
> cd <directory name> # change your current directory to <directory name>

> cd ~ # change your current directory to your home directory

> ls # list all the files in the cu

> ls -ahl # the same, but show me a

> ls <directory name> # list all th

> mkdir <directory name> # make a d

> rmdir <directory name> # remove t                                    e>
(only works if it's empty)

> pwd # list what directory you're in
```

HOT TIP

This is just a very basic list to give you a flavour, and everyone has their own versions/options of these commands that they love. Use what makes the most sense!

Some terminal basics...

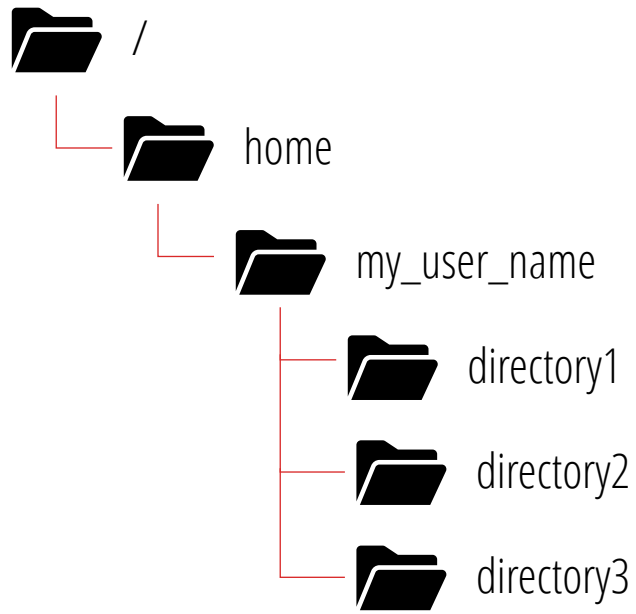
```
> cd <directory name> # change your current directory to <directory name>

> cd ~ # change your current directory to your home directory

> ls # list all the files in the current directory

> ls -ahl # the same, but show me all the files in a list

> ls <directory name> # list all the files

> mkdir <directory name> # make a directory

> rmdir <directory name> # remove the directory
(only works if it's empty)

> pwd # list what directory you're in
```

HOT TIP

You can type in "man <command name>" to get the **man**ual page of the command, which shows you all of the options for that command and how to use it.

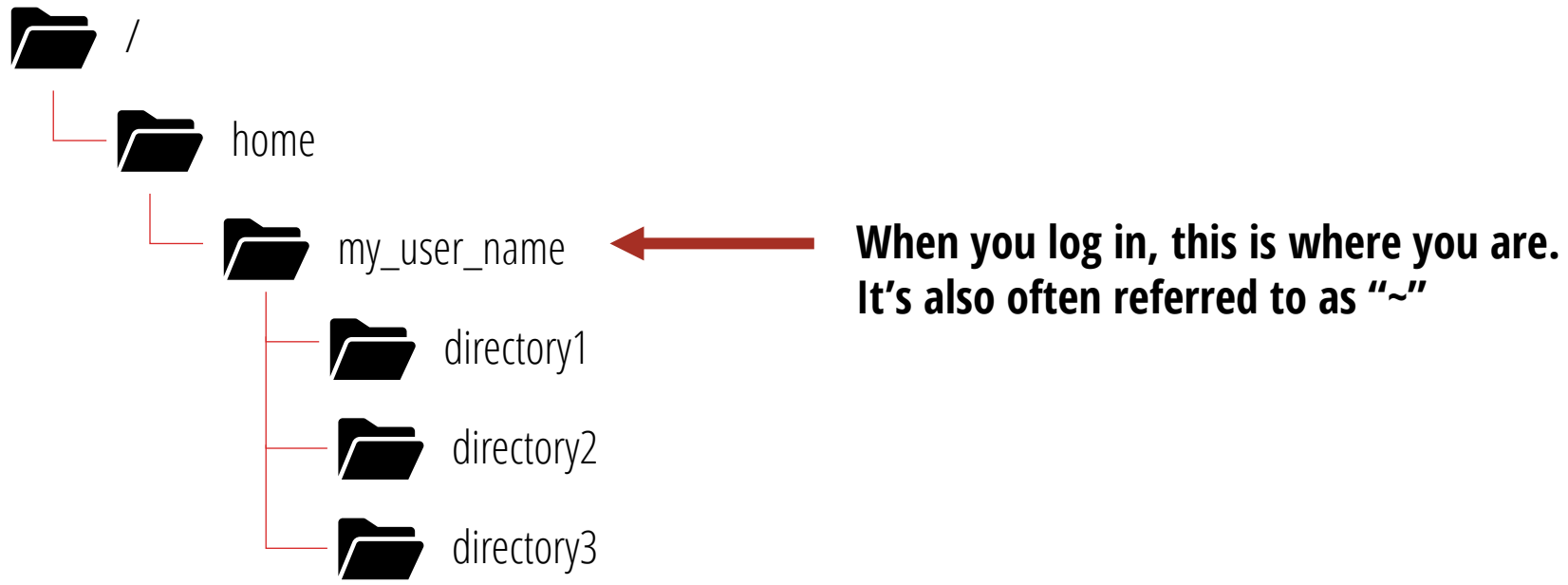
Navigating your directories

Typically, on your computer, there's a tree of directories:



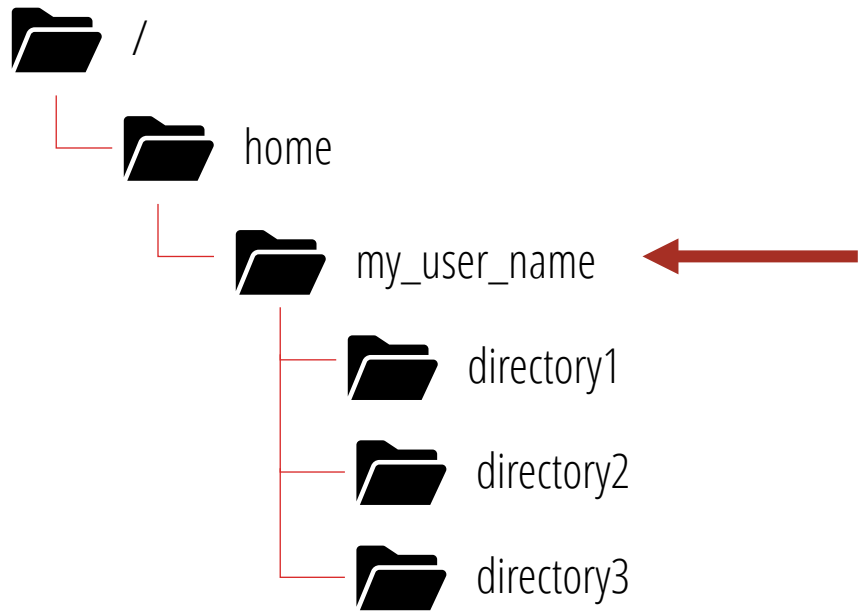
Navigating your directories

Typically, on your computer, there's a tree of directories:



Navigating your directories

Typically, on your computer, there's a tree of directories:



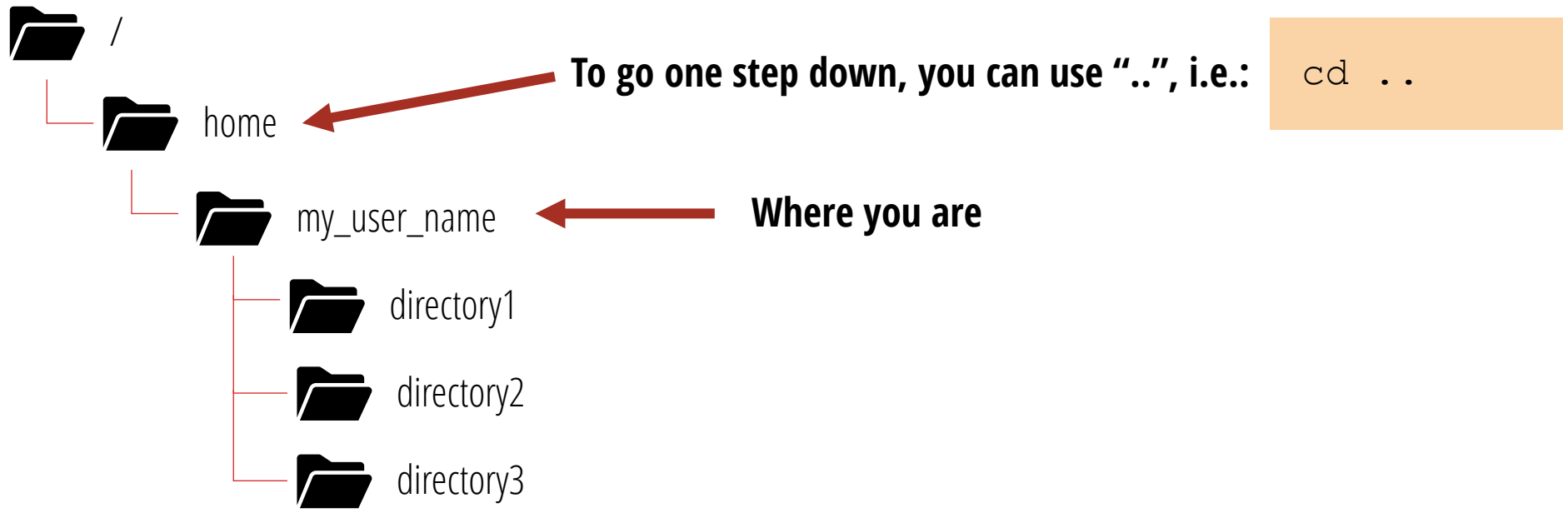
When you log in, this is where you are.
It's

COOL CATCH

In Windows (not using WSL), this location is usually in your User directory. For instance, for me, it is located in "C:\Users\mubdi"

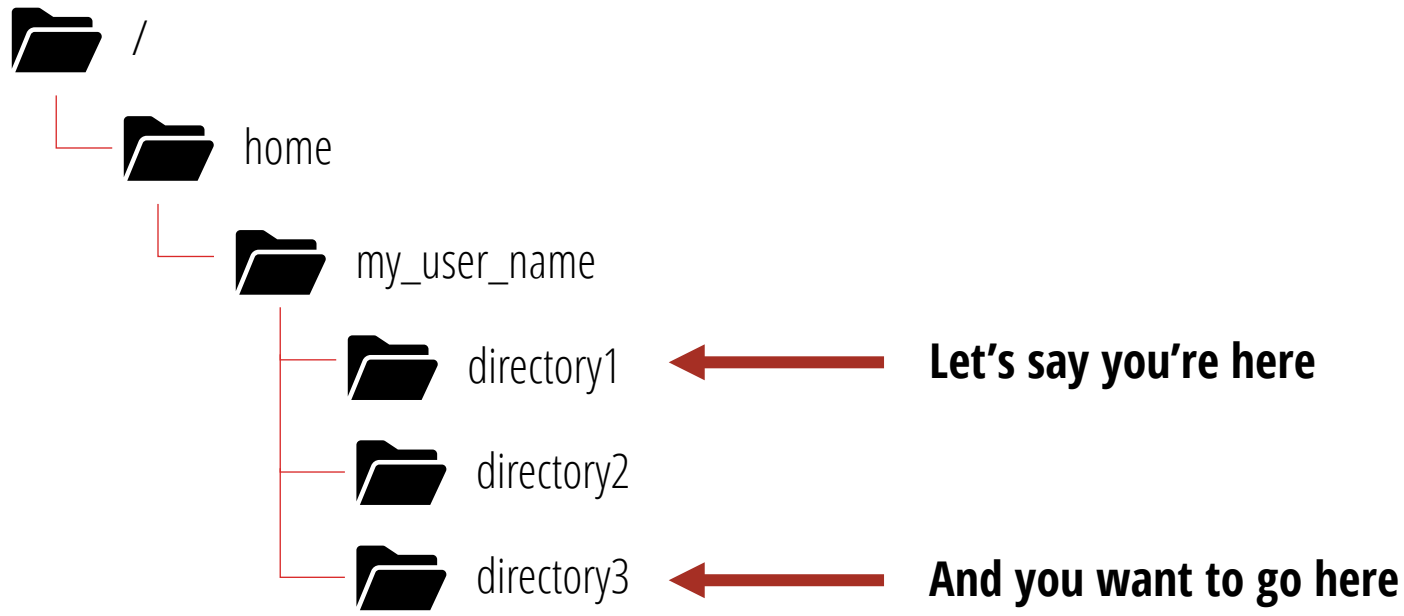
Navigating your directories

Typically, on your computer, there's a tree of directories:



Navigating your directories

Typically, on your computer, there's a tree of directories:



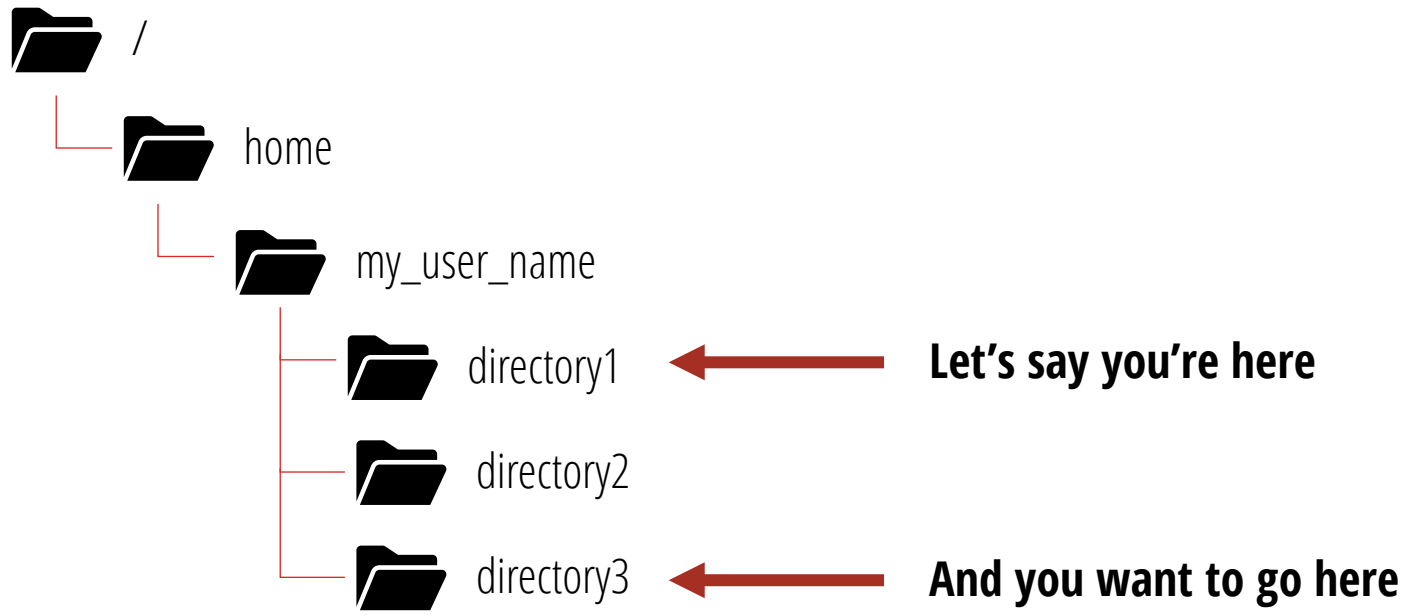
Navigating your directories

Typically, on your computer, there's a tree of directories:


```
cd ../directory3
```

Navigating your directories

Typically, on your computer, there's a tree of directories:

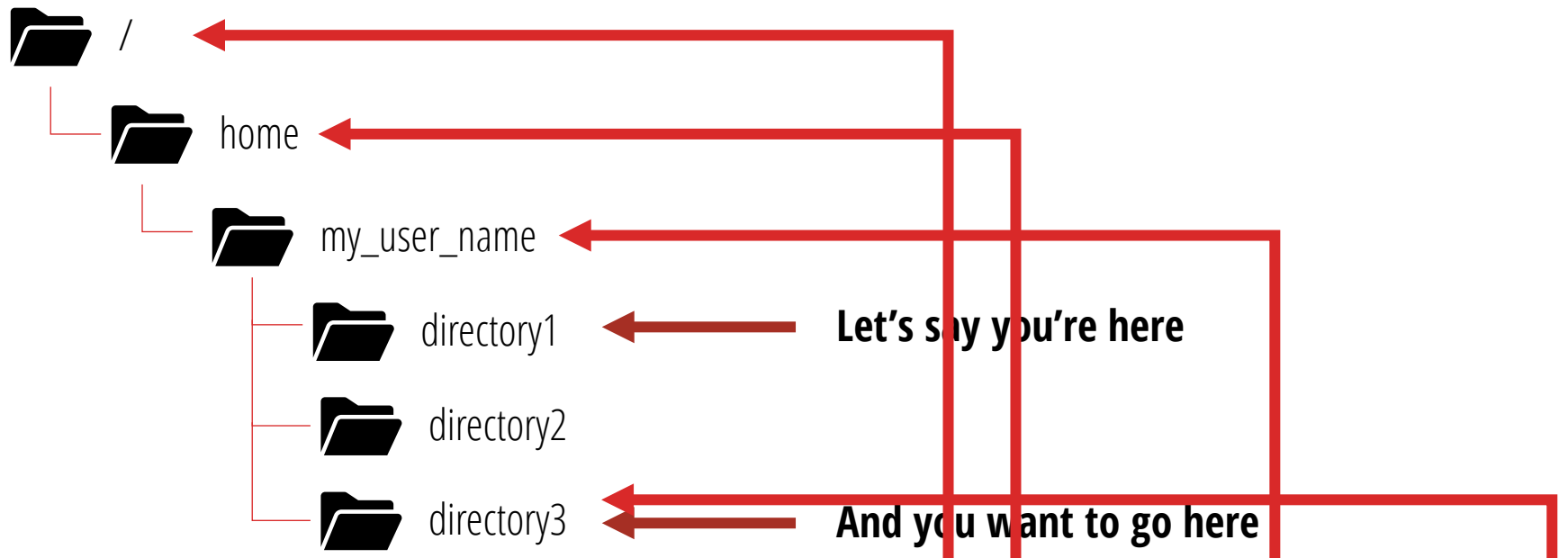


Or you could use absolute paths (i.e., right from the top):

```
cd /home/my_user_name/directory3
```

Navigating your directories

Typically, on your computer, there's a tree of directories:

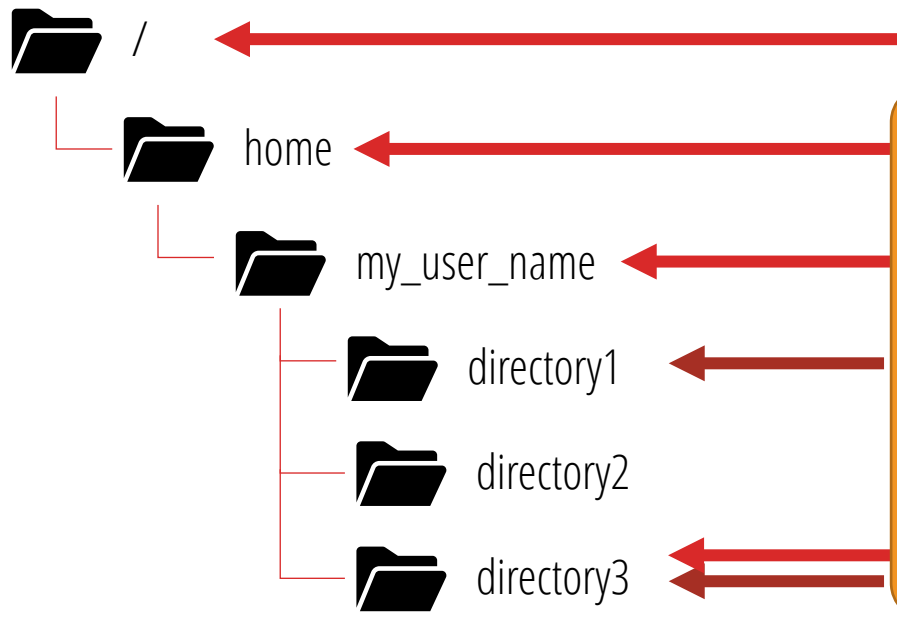


Or you could use absolute paths (i.e., right from the top):

```
cd /home/my_user_name/directory3
```

Navigating your directories

Typically, on your computer, there's a tree of directories:



HOT TIP

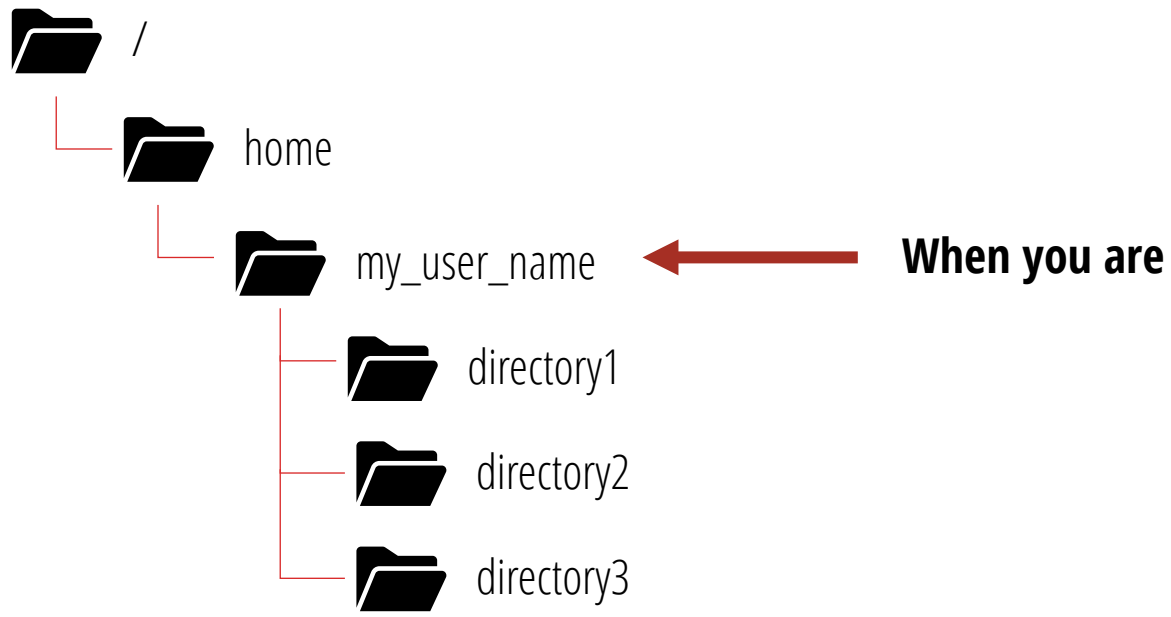
It's okay to be lazy, and something you want to be lazy about is typing. You can usually press "tab" to try to complete the name of directories and/or files, if you start typing the first few letters.

Or you could use absolute paths (i.e., right from the top):

```
cd /home/my_user_name/directory3
```

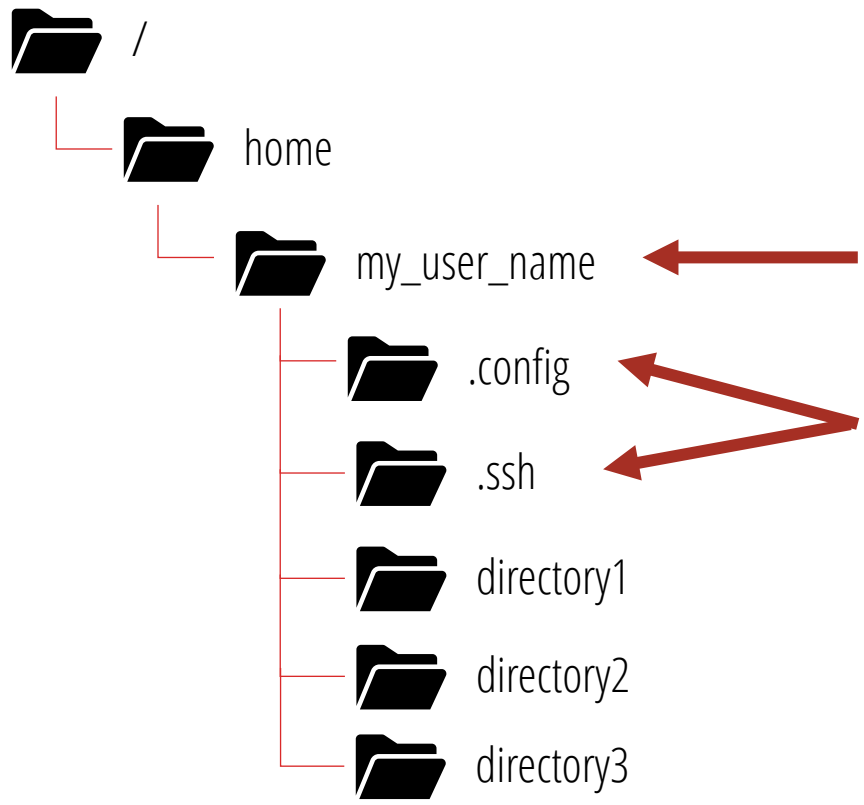
Hidden Directories

When you run an “ls”, you don’t see everything. In your home directory, try running “ls –a”



Hidden Directories

When you run an “ls”, you don’t see everything. In your home directory, try running “ls –a”

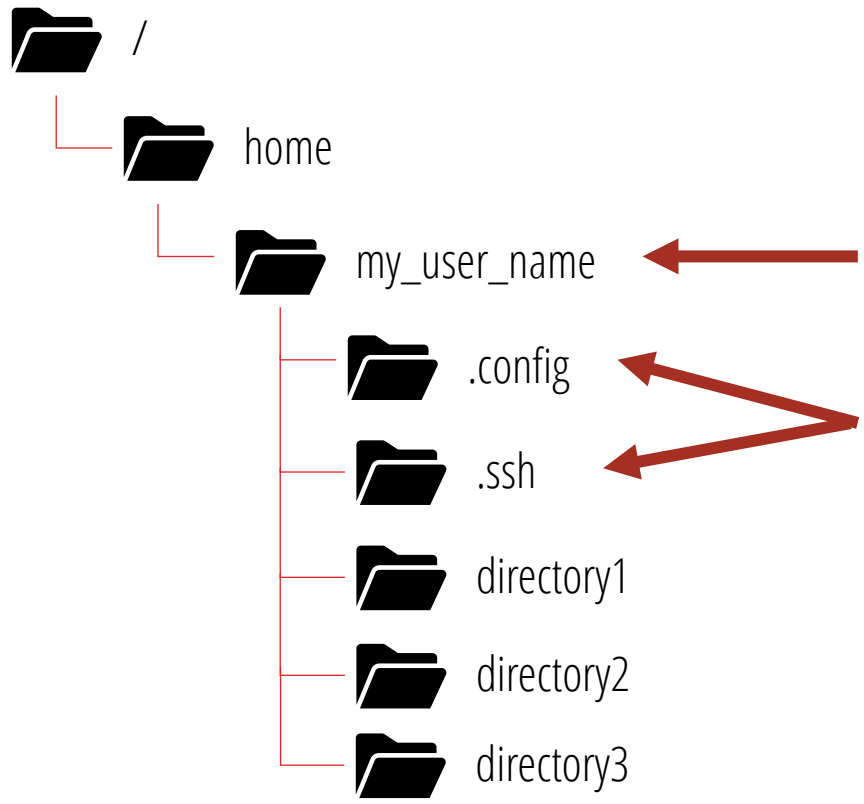


When you are

Hidden directories: These are where your computer stores various configuration files. For instance, for python, or for ssh

Hidden Directories

When you run an “ls”, you don’t see everything. In your home directory, try running “ls –a”

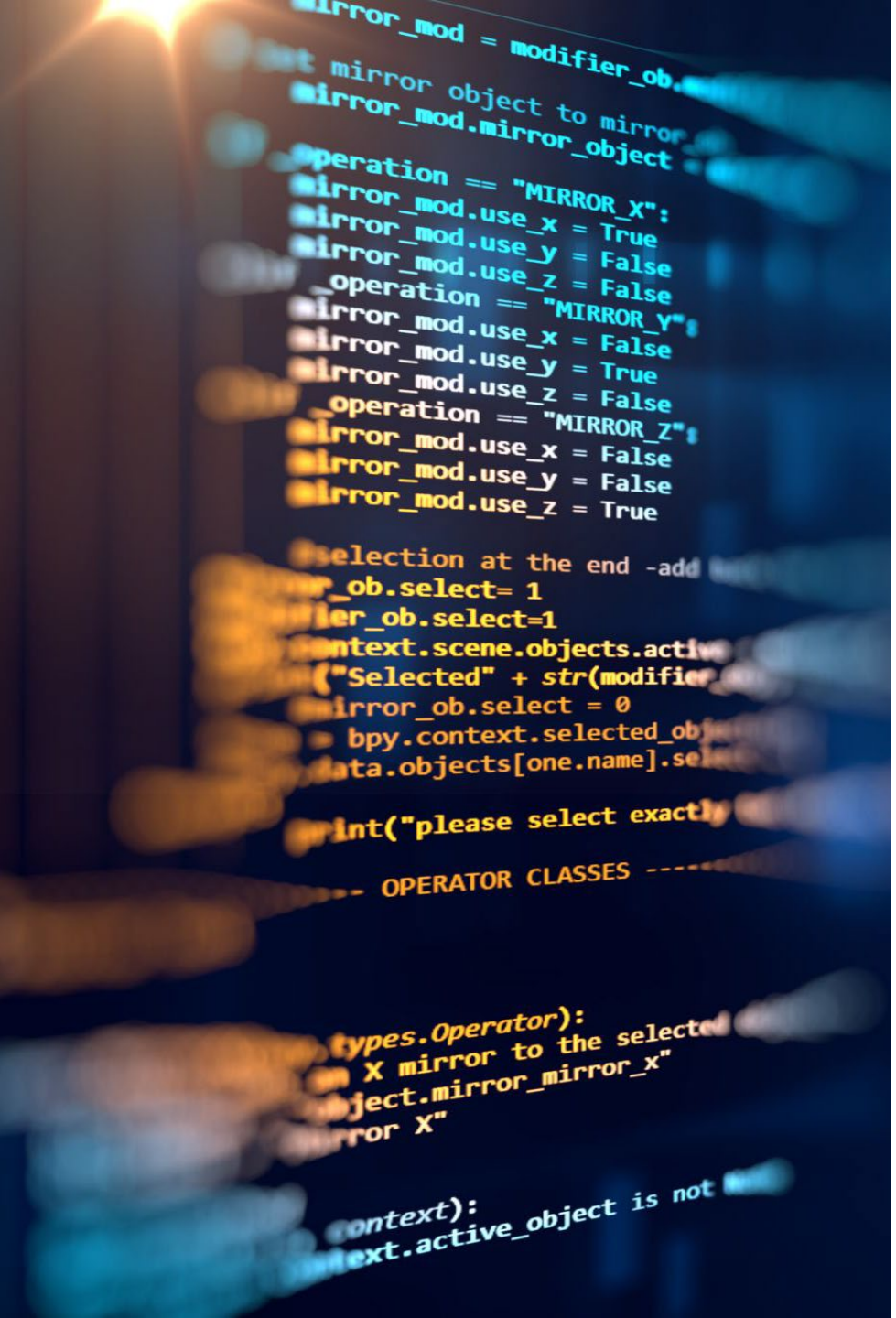


When you are

Hidd
vari
ssh

COOL CATCH

If a directory or file starts with a period (.), it is a hidden file, and won’t show up by default with the “ls” command. We’ll be playing with a bunch of such directories and files in this bootcamp.



SSH

One of the most powerful things about computing is often connecting to other computers! The way most computers that you'll encounter doing this is through the command line through a program called "ssh". That stands for "**s**ecure **sh**ell".

Importantly, anything you communicate across different computers will be encrypted.

If you're going to use a system like Compute Canada/SciNet or CITA, this will be important for you.

Some Basic SSH Commands

```
> ssh <server_name> # ssh into <server_name> with whatever your current
username is

> ssh <username>@<server_name> # ssh into a server with a specific
username

> ssh <server_name> -l <username> # same as above, but longer

> ssh-keygen # generates a security key to allow you to login to places
without a password

> ssh-copy-id <server_name> # copies your generated security key over to
<server_name>
```

Some Basic SSH Commands

```
> ssh <server_name> # ssh into <server_name> with whatever your current
username is

> ssh <username>@<server_name> # ssh into a server with a specific
username

> ssh <server_name> -l <username> # same as above, but longer

> ssh-keygen # generates a security key to
without a password

> ssh-copy-id <server_name> # copies your
<server_name>
```

HOT TIP

You can save settings for different ssh servers in your ~/.ssh/config file, for instance, if you'd like ports forwarded, or if you'd like to use a specific user name

Some Basic SSH Commands

```
> ssh <server_name> # ssh into <server_name> with whatever your current
username is

> ssh <username>@<server_name> # ssh into a server with a specific
username

> ssh <server_name> -l <username> # same as above, but longer

> ssh-keygen # generates a security key to
without a password

> ssh-copy-id <server_name> # copies your
<server_name>
```

HOT TIP

If you need to forward a **port** from another computer to yours, you can use the –L flag. For instance, if you wanted to forward port 8888 to your computer's port 8888, you can use the flag:

```
-L 8888:localhost:8888
```

Moving Files around

More often then not, you'll want to move files around a system, or between systems. Here's how to do it on your own computer:

```
> cp <file_name> <destination_name> # copy a single file from one location
to another

> cp -r <directory_name> <destination_name> # copy a whole directory (and
everything in it) from one location to another

> mv <file_name> <destination_name> # move a file or directory from one
location to another
```

Moving Files around

More often then not, you'll want to move files around a system, or between systems.
uter:

```
me> # copy a single file from one location

nation_name> # copy a whole directory (and
tion to another

me> # move a file or directory from one
```

HOT TIP

Wildcards (*) are your friend! If you want to copy a selection of files that match a certain bit of the filename, you can use a wildcard to represent everything that's *not* supposed to match. For instance:

```
*.txt
```

Will match:

```
a.txt, b.txt, c.txt
```

Moving Files around

How about on another system? You can copy over ssh using **scp:**

```
> scp <file_name> <user_name>@<server_name>:<destination_name> # copy a
single file from your computer to a server.

> scp <user_name>@<server_name>:<file_name> <destination_name> # copy a
                                ion on your computer.

                                :<directory_name> <destination_name> #
                                location on your computer.
```

HOT TIP

For individual files, of if you don't remember exactly what the file was called, don't be a hero – use a GUI. For Windows, I use the WinSCP client. For Mac OSX, Cyberduck is a popular choice.

Moving Files around

How about on another system? You can copy

```
> scp <file_name> <user_name>@<serve
single file from your computer to a

> scp <user_name>@<server_name>:<fil
single file from a server to locatio

> scp -r <user_name>@<server_name>:<
copy a directory from a server to lo
```

#

HOT TIP

Is the file you want openly available on the internet? If so, an easy way to grab the file is using the command **wget**. If you have a particular URL, you can download the file to your current directory by:

```
wget <url_of_file>
```

Moving Files around

How about on another system? You can copy over ssh using **scp:**

```
> scp <file_name> <user_name>@<server_name>:<destination_name> # copy a
single file from your computer to a server.

>                                                  e_name> <destination_name> # copy a
s                                                  n on your computer.

>                                                  directory_name> <destination_name> #
c                                                  cation on your computer.
```

HOT TIP

Do you have a more complicated/larger transfer that you need to complete? Perhaps you need to run it regularly? Take a look at **rsync** which is beyond the scope of this bootcamp, but will help you immensely. There's a lot of options, so best to Google/Search for what you'd like to do for the correct command.