

# Linux and git

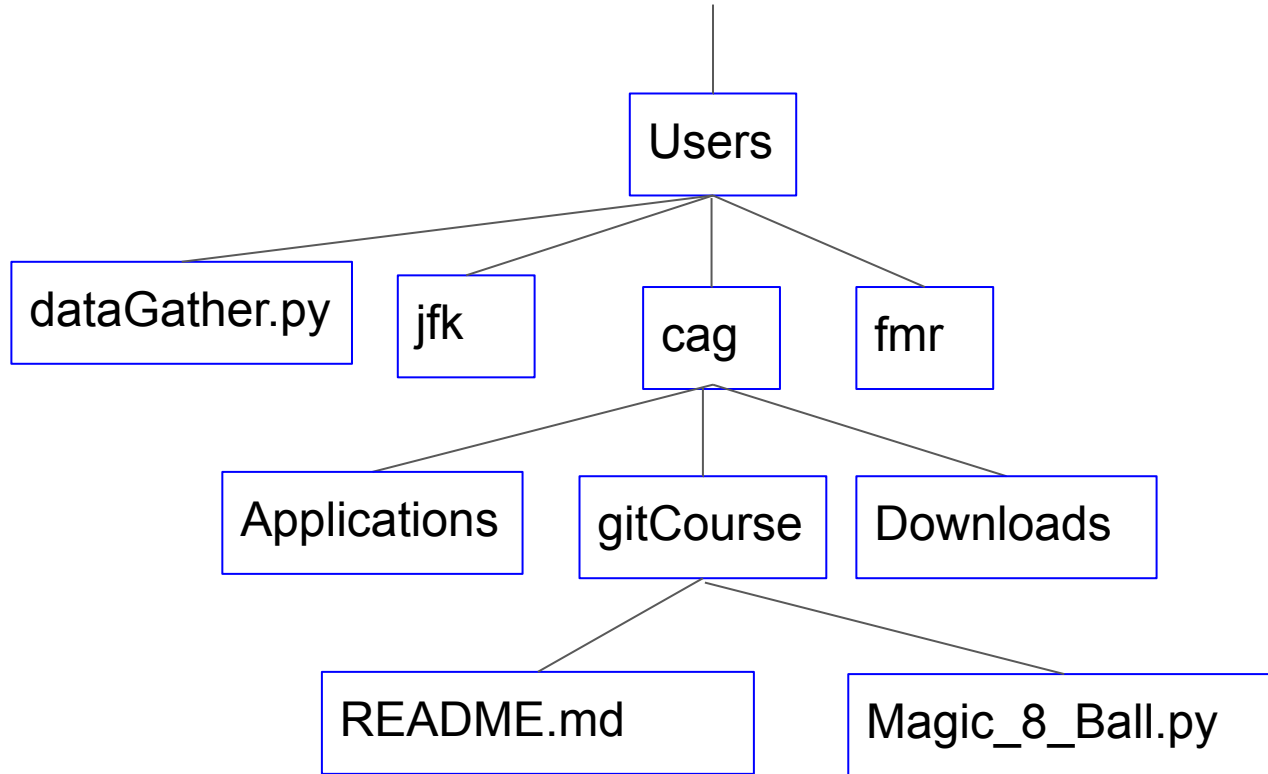
git for Field Technicians

# Goals

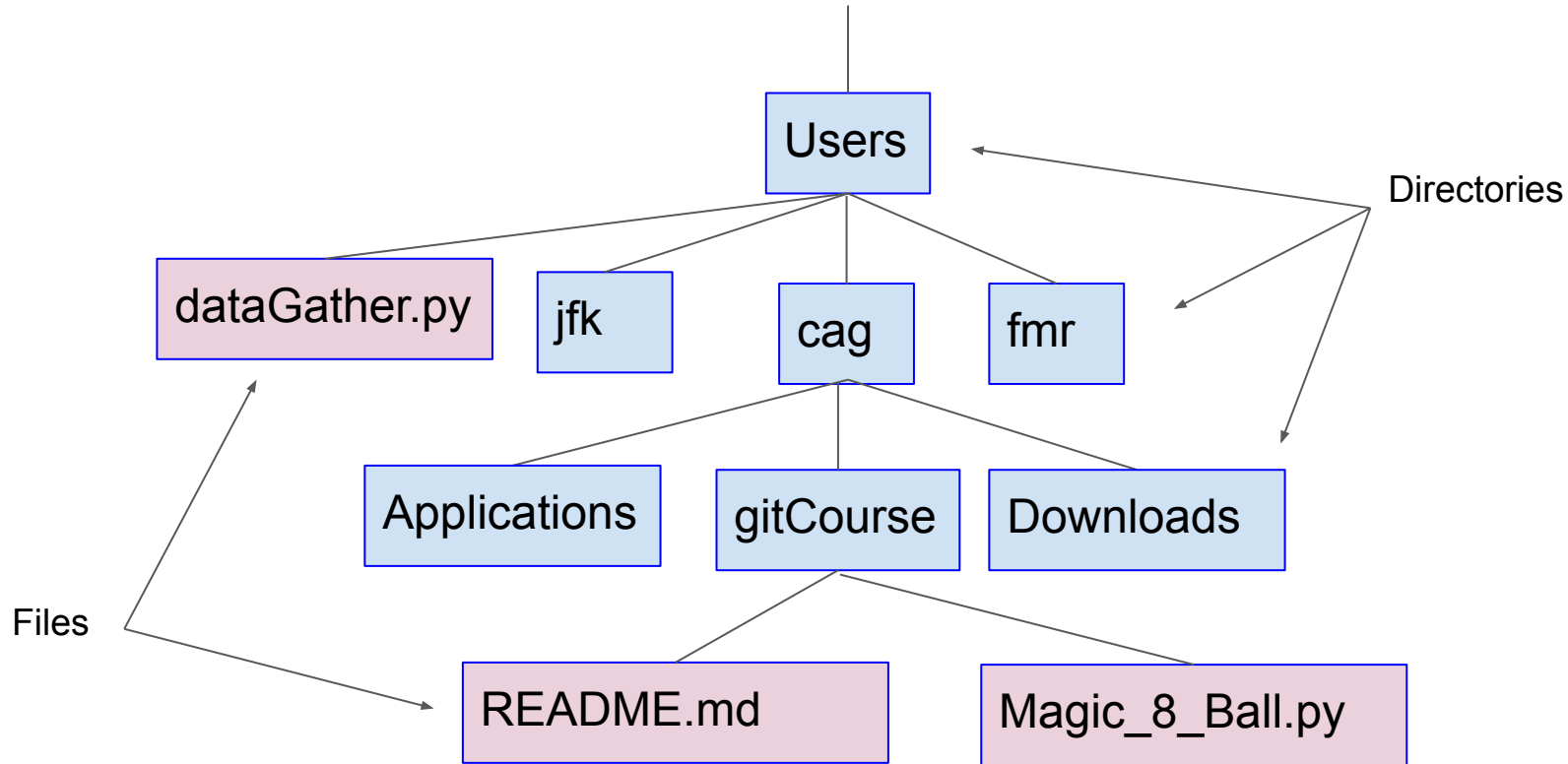
1. Check the status of the code on your computer
2. Check out, fetch, and pull your repository
3. Clone a git repository
4. Create a directory

# Introduction to Linux

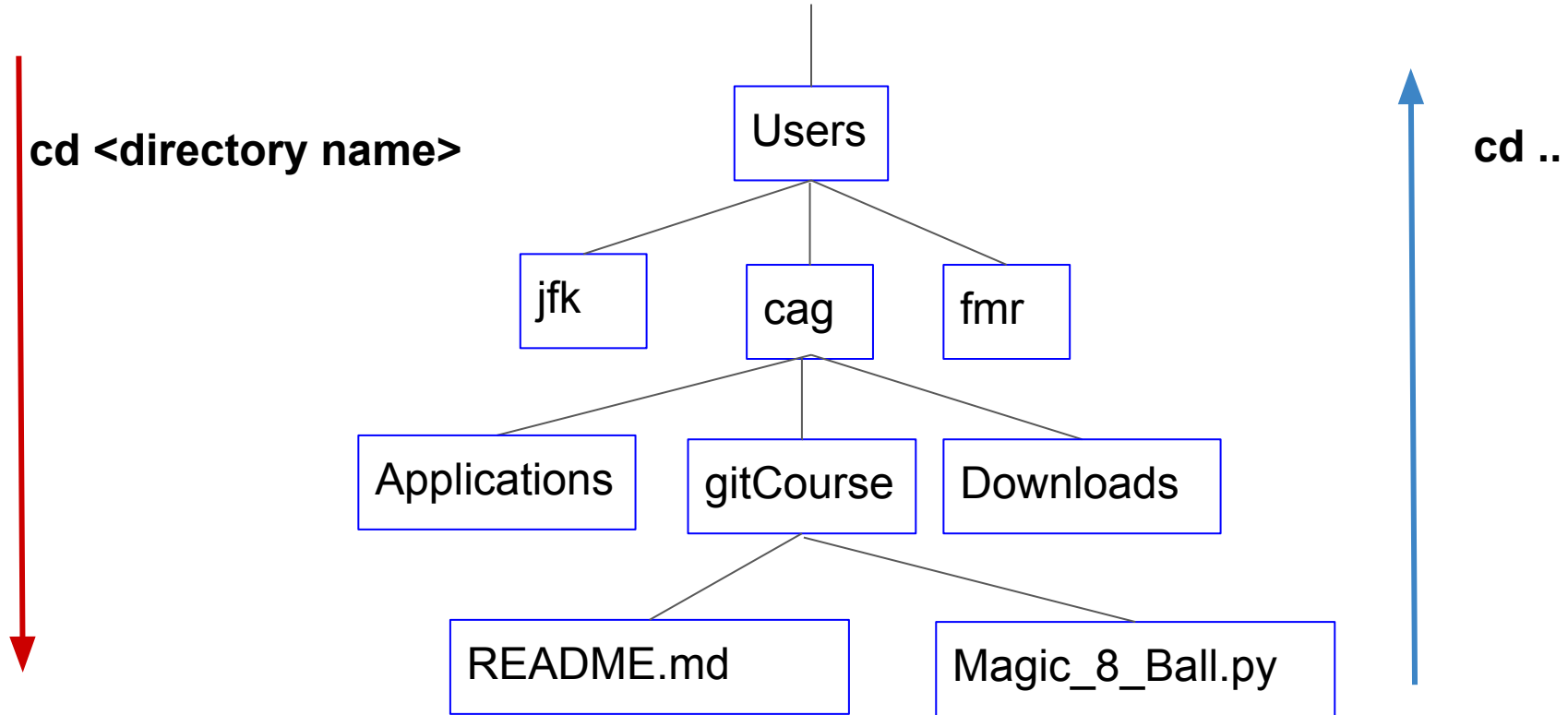
# Linux is structured like a tree



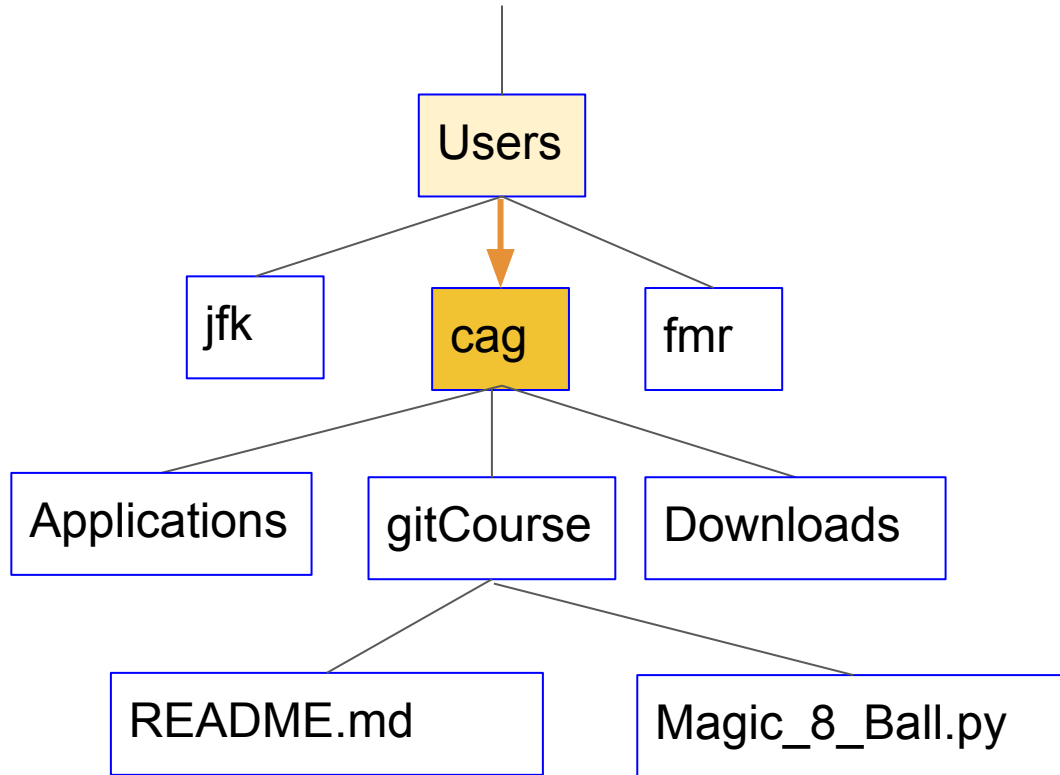
# Directories can contain files and directories



Use the command **cd** to travel up and down the tree



# Traversing down the tree

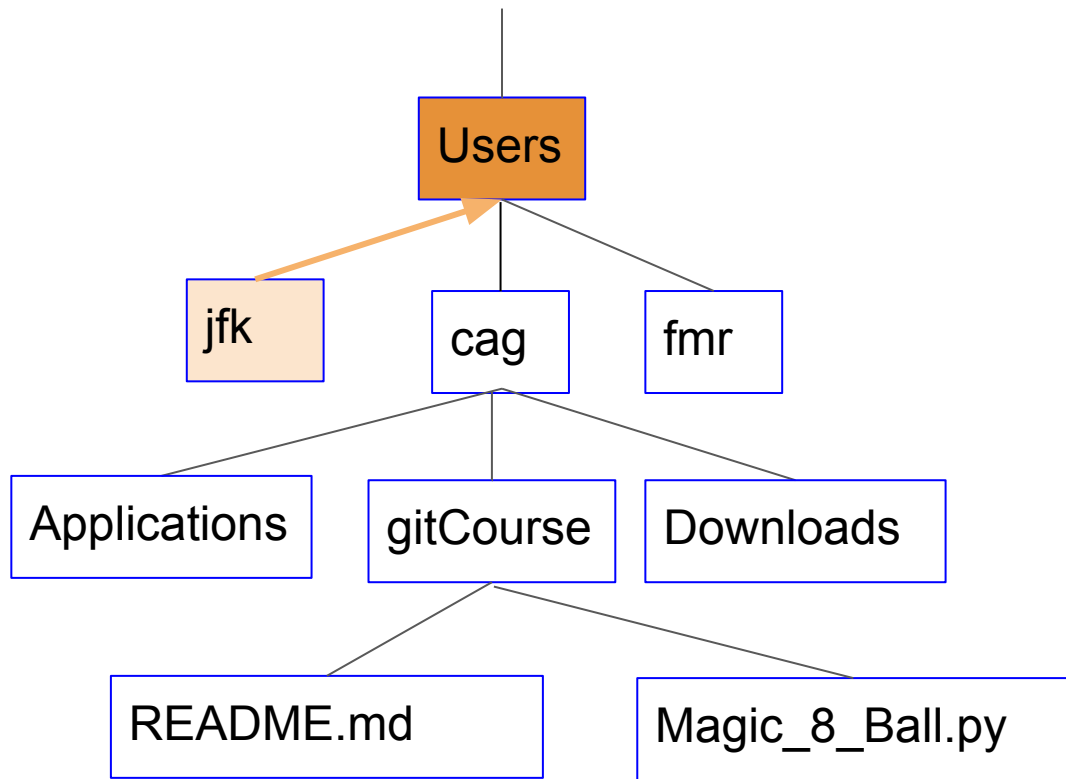


You are in the **Users** directory and you would like to go to the **cag** directory

`cd <directory name>`

**`cd cag`**

# Traversing up the tree

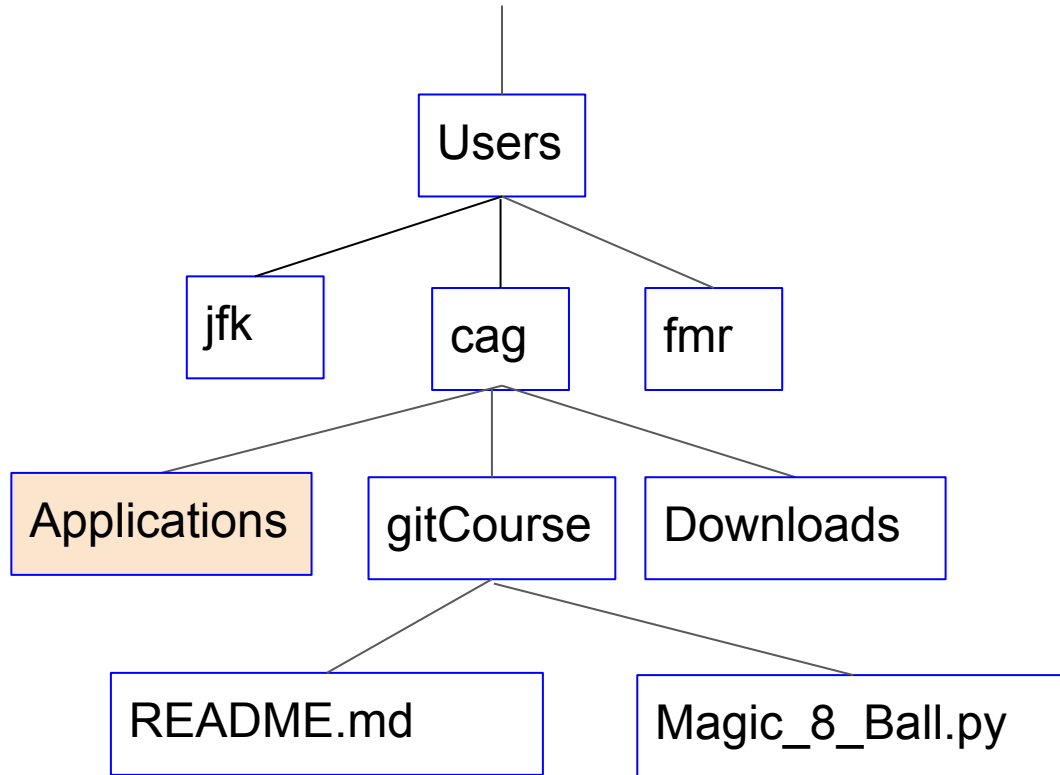


You are in the **jfk** directory and you would like to go to the **Users** directory

**cd ..**



# Finding your location in the tree

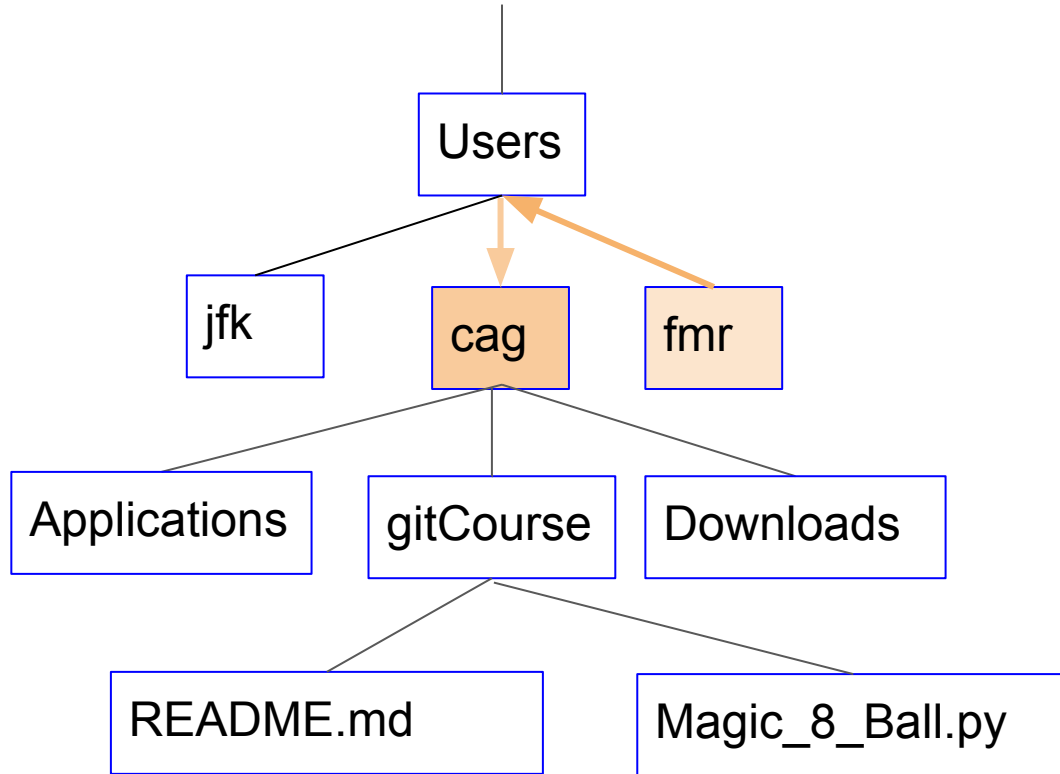


You are working in Linux and you want to know in which directory you are working.

**pwd**

The name of the directory you are in is printed on the terminal.

# Going home

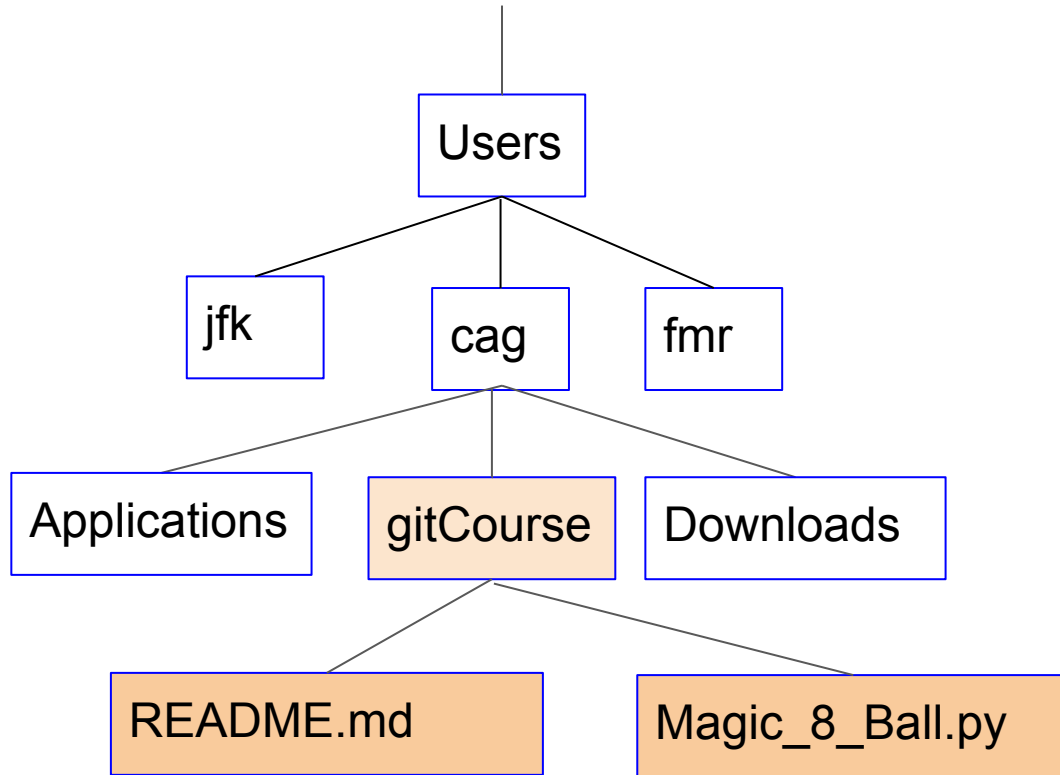


You are working in Linux  
and you want to go to  
your home directory

**cd ~**

You will return to your  
home directory

# Listing directory contents

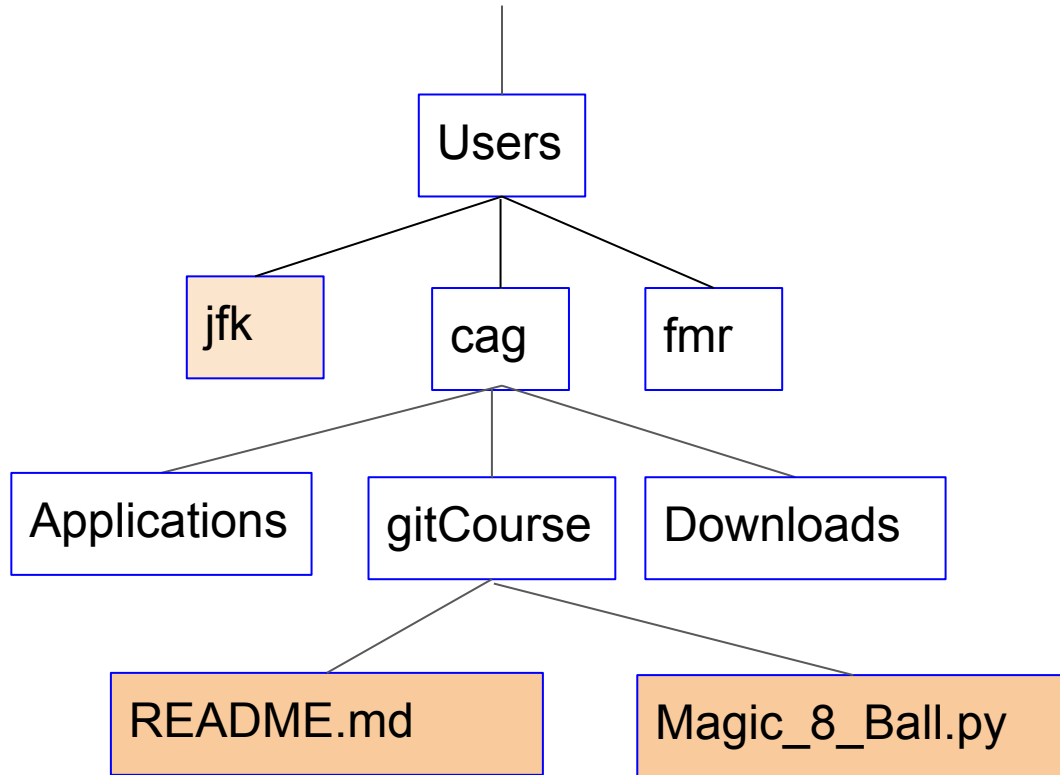


You are working in the gitCourse directory and you want to know what is in the directory.

**ls**

The contents of gitCourse will be printed on the terminal

# Listing another directory's contents



You are working in the **jfk** directory and you want to know what is in the gitCourse directory.

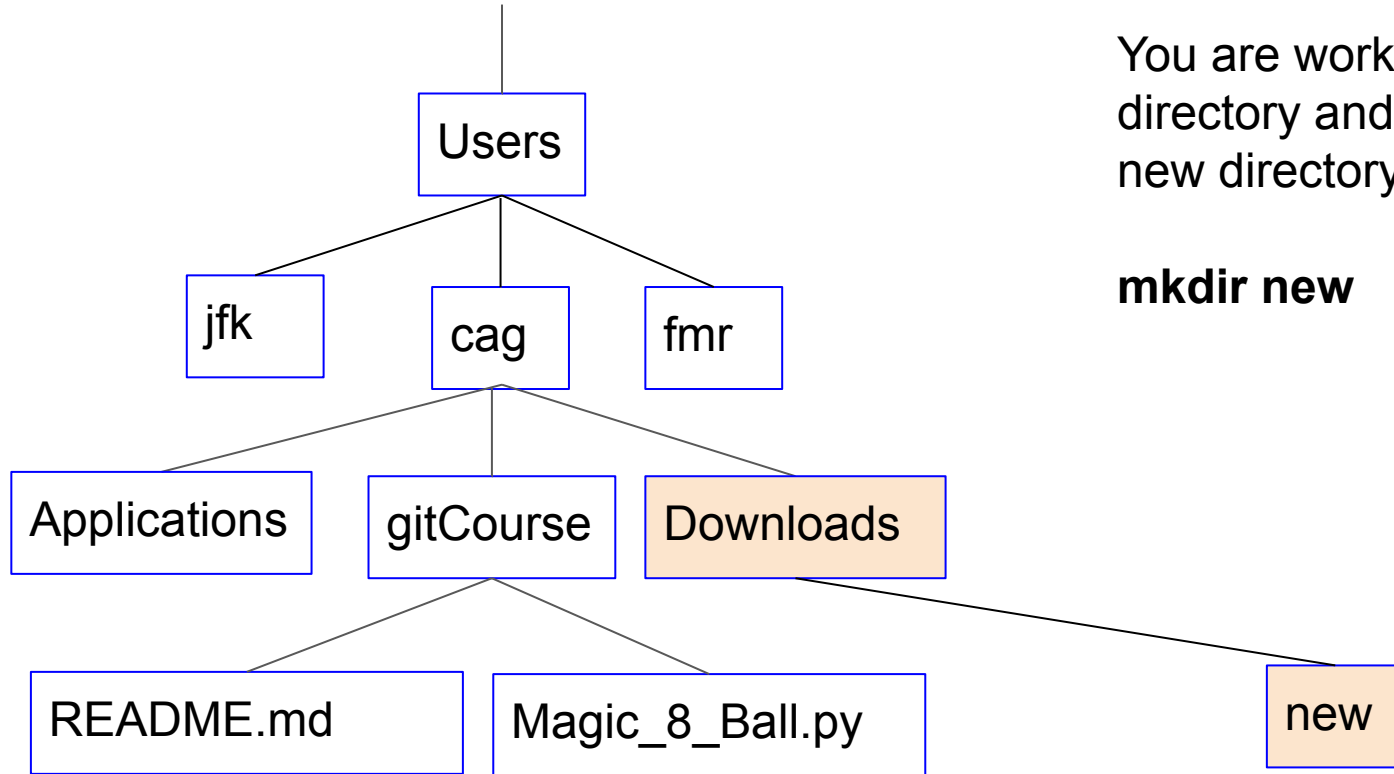
**Is ~cag/gitCourse**

The contents of gitCourse will be printed on the terminal

# Make a new directory

You are working in the **Downloads** directory and you want to create a new directory called new

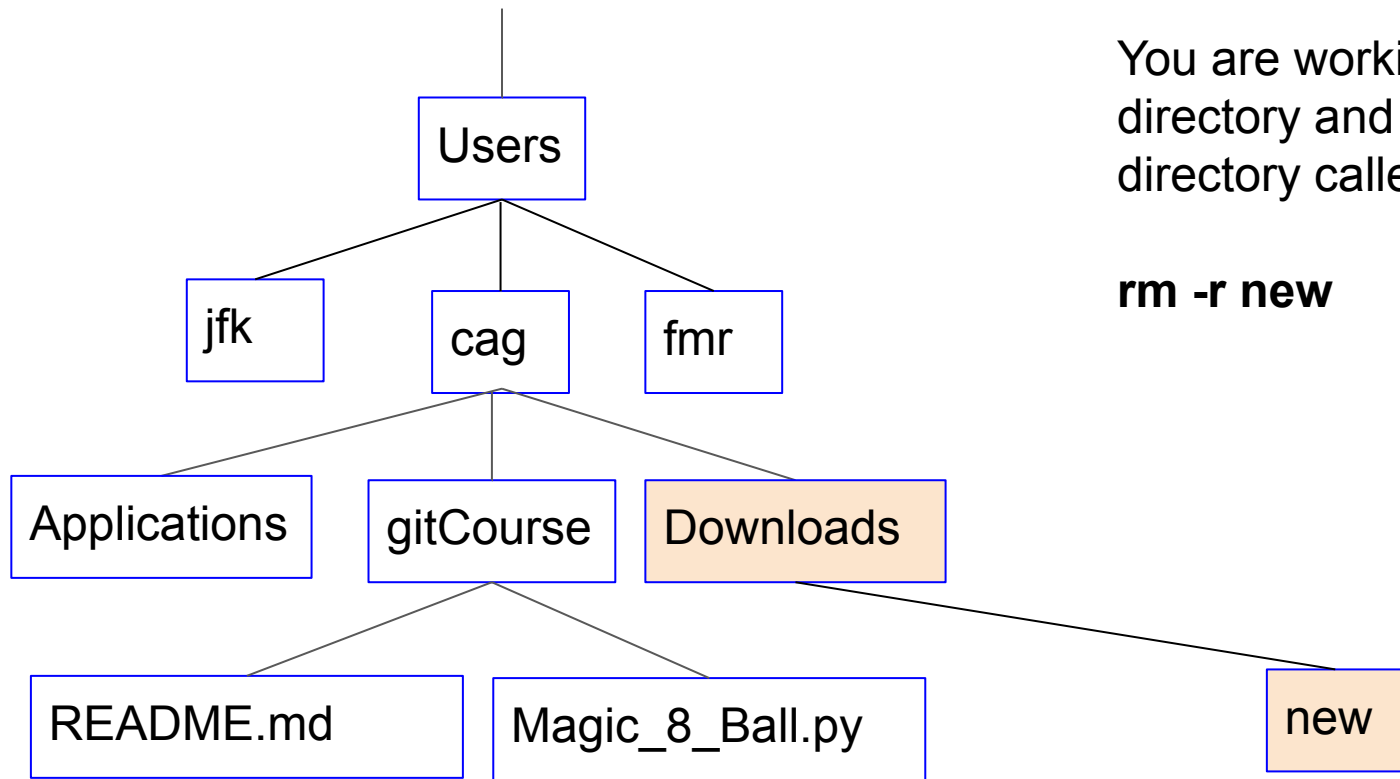
**mkdir new**



# Delete a directory

You are working in the **Downloads** directory and you want to delete a directory called new

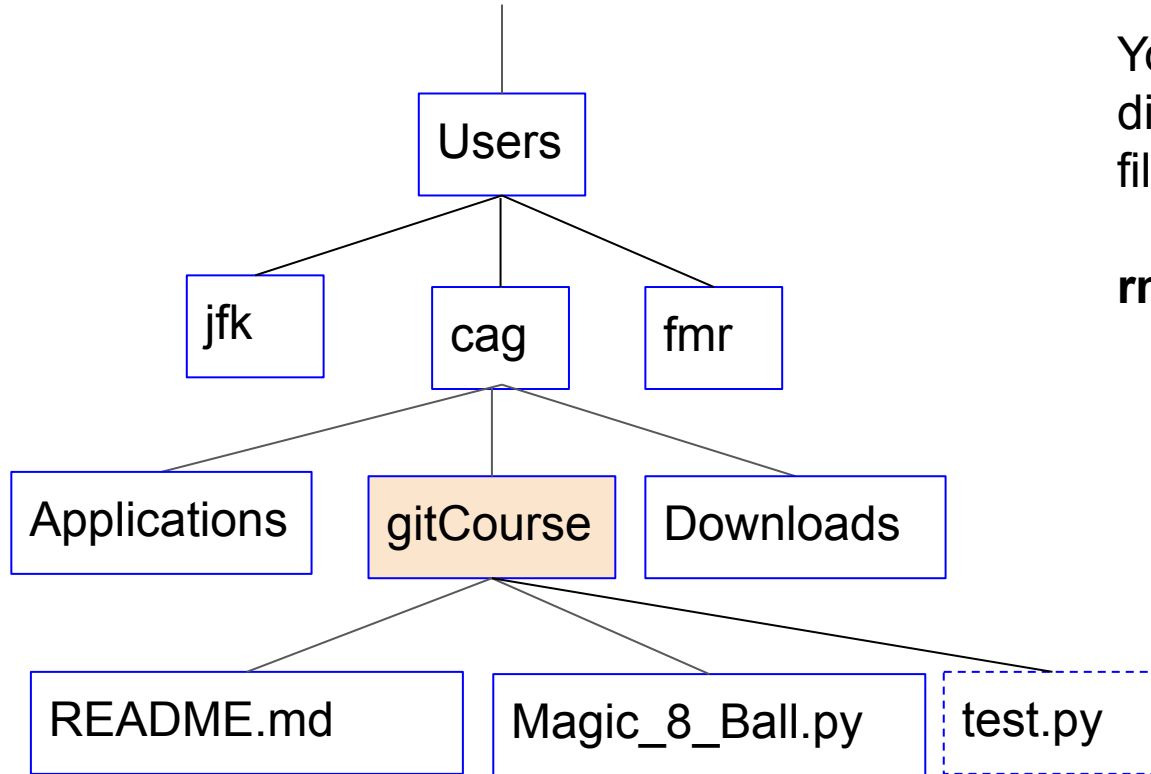
**rm -r new**



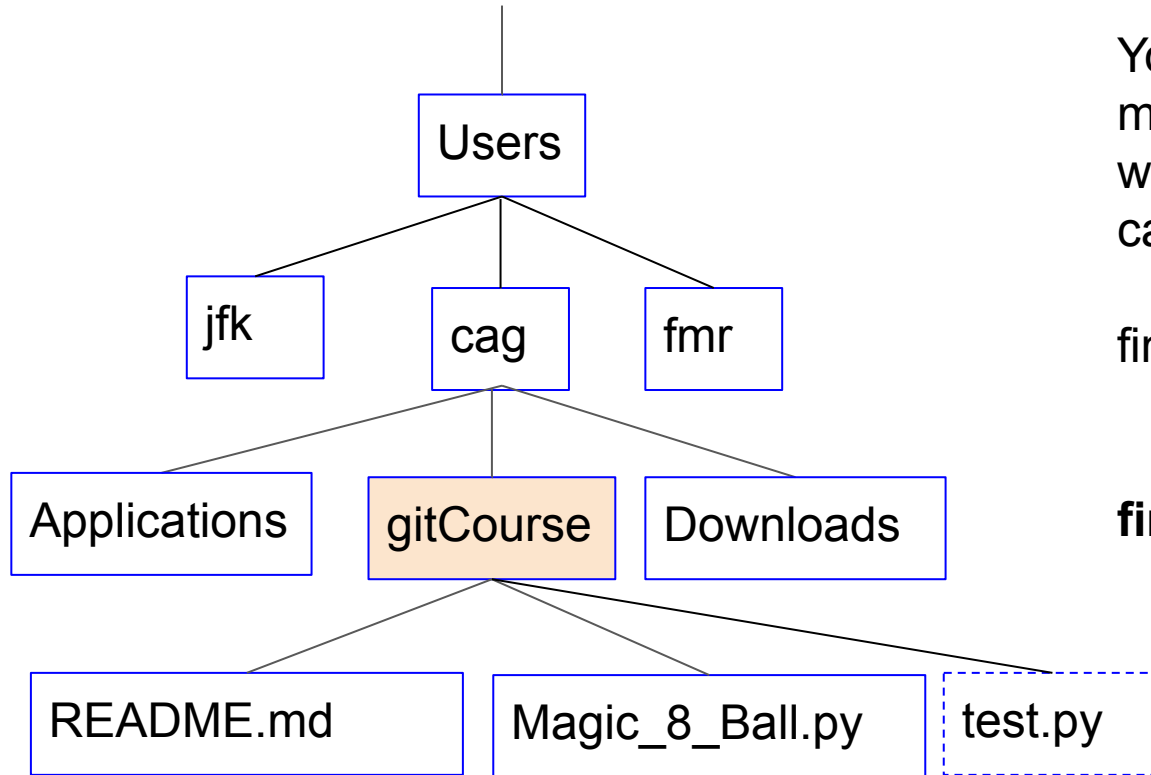
# Delete a file

You are working in the **gitCourse** directory and you want to delete a file called test.py

**rm test.py**



# Find a file or directory



You are working on your Linux machine. You can't remember which directory contains your file called `Magic_8_Ball.py`

`find ~/*/<file or directory name>`

**`find ~/*/Magic_8_Ball.py`**





# Lab

The Linux Environment

# Lab: The Linux environment

In this lab you:

Check for the presence of the correct directory

Make the directory if it is not present

# Open a terminal window

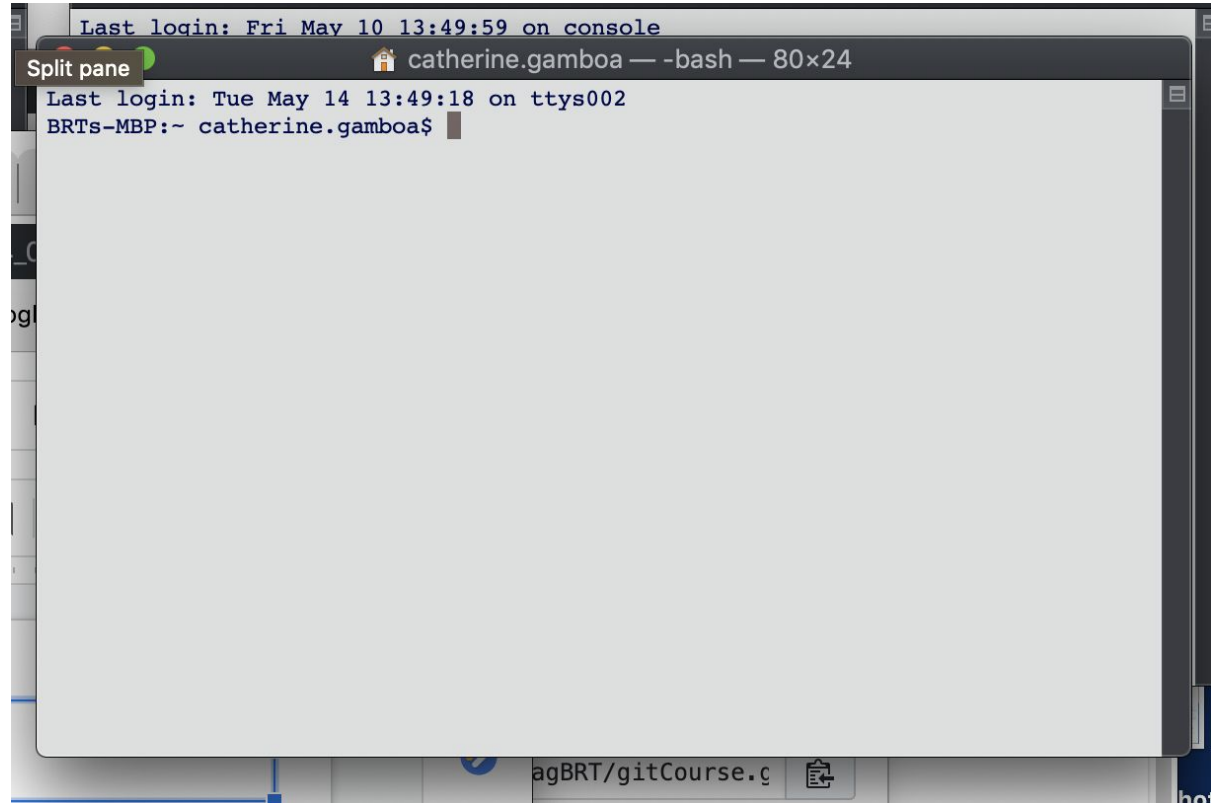
You can open a terminal window using any method you choose.

If you like, use the shortcut keys -  
Ctrl-Cmd-T



1 Press **Ctrl** + **Alt** + **T**. This will launch the Terminal.<sup>[1]</sup>

# Use Linux commands in the terminal window

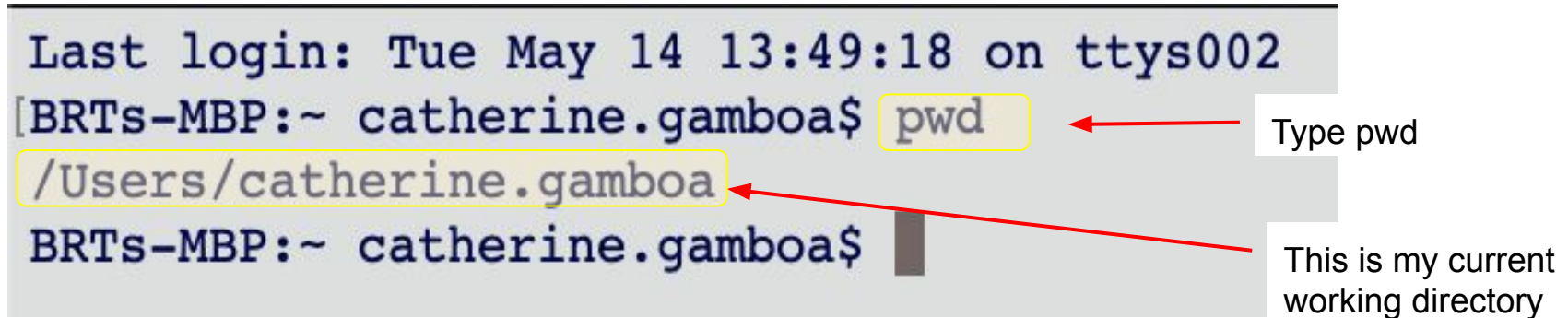


A screenshot of a terminal window on a macOS system. The window has a title bar with a 'Split pane' button and a title 'catherine.gamboa — -bash — 80x24'. The terminal content shows two previous login sessions: 'Last login: Fri May 10 13:49:59 on console' and 'Last login: Tue May 14 13:49:18 on ttys002'. The current prompt is 'BRTs-MBP:~ catherine.gamboa\$' with a cursor. The window is part of a larger application, with a dock at the bottom showing a file named 'agBRT/gitCourse.ç'.

```
Last login: Fri May 10 13:49:59 on console
Split pane
catherine.gamboa — -bash — 80x24
Last login: Tue May 14 13:49:18 on ttys002
BRTs-MBP:~ catherine.gamboa$
```

# Use the command **pwd** to find where you are in the computer

pwd (print working directory): this command prints the current directory.



```
Last login: Tue May 14 13:49:18 on ttys002
[BRTs-MBP:~ catherine.gamboa$ pwd
/Users/catherine.gamboa
BRTs-MBP:~ catherine.gamboa$
```

The image shows a terminal window with a light gray background. The text is in a monospaced font. The first line is the login message. The second line shows the prompt and the command 'pwd' being entered. The third line shows the output of the command. There are two yellow rectangular highlights: one around 'pwd' and one around the output path. Two red arrows point from text boxes on the right to these highlights. The first arrow points to the 'pwd' highlight and is labeled 'Type pwd'. The second arrow points to the output path highlight and is labeled 'This is my current working directory'.

Type pwd

This is my current working directory

# The directory name is part of the Linux prompt

```
Last login: Tue May 14 13:49:18 on ttys002
[BRTs-MBP:~ catherine.gamboa$ pwd
/Users/catherine.gamboa
BRTs-MBP:~ catherine.gamboa$
```

Notice the current directory name is included as part of the Linux prompt

# The ~ (tilda) command signifies your home directory

```
Last login: Tue May 14 13:49:18 on ttys002
[BRTs-MBP:~ catherine.gamboa$ pwd
/Users/catherine.gamboa
BRTs-MBP:~ catherine.gamboa$
```



This means my home directory

# Use the command **cd** to change directories

cd (change directory): is used to change current working directory

For example `cd <directory name>`

Type 'cd directory name'

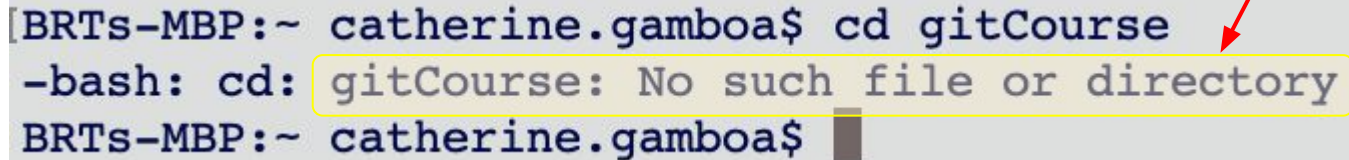


```
[BRTs-MBP:~ catherine.gamboa$ cd gitCourse  
-bash: cd: gitCourse: No such file or directory  
BRTs-MBP:~ catherine.gamboa$
```



# The directory may not exist

The directory does not exist in my home directory



```
[BRTs-MBP:~ catherine.gamboa$ cd gitCourse  
-bash: cd: gitCourse: No such file or directory  
BRTs-MBP:~ catherine.gamboa$
```

A terminal window showing a user attempting to change the directory to 'gitCourse'. The command 'cd gitCourse' is entered, but the system responds with an error: '-bash: cd: gitCourse: No such file or directory'. A red arrow points from the text 'The directory does not exist in my home directory' to the error message. The error message is highlighted with a yellow box.

# The directory may not exist



**Caution:** it is a very common mistake to forget where you created your directory.

If you use 'cd <directory name>' in the wrong directory, you will get this message, even though the directory exists somewhere on your computer.

(use the find command to find the file or directory)

The directory does not exist in my home directory

```
[BRTs-MBP:~ catherine.gamboa$ cd gitCourse  
-bash: cd: gitCourse: No such file or directory  
BRTs-MBP:~ catherine.gamboa$
```

# Create a directory


Step 1: make sure you are in the right directory

**pwd**

Step 2: make the directory using the 'mkdir' command

**mkdir gitCourse**

Make the directory called gitCourse



```
[BRTs-MBP:~ catherine.gamboa$ mkdir gitCourse
```

# List the contents of the current directory to check that gitCourse was created

**ls:** list the contents of the directory

List the contents of the current directory

The directory gitCourse was created

```
[BRTs-MBP:~ catherine.gamboa$ mkdir gitCourse
[BRTs-MBP:~ catherine.gamboa$ ls
Applications      Library           Public
Desktop           Movies            code
Documents         Music             gitCourse
Downloads         Pictures          iCloud Drive (Archive)
BRTs-MBP:~ catherine.gamboa$
```

# Introduction to git

# What is Git?



**Git** is a version control system to keep track of changes to files and projects over time.

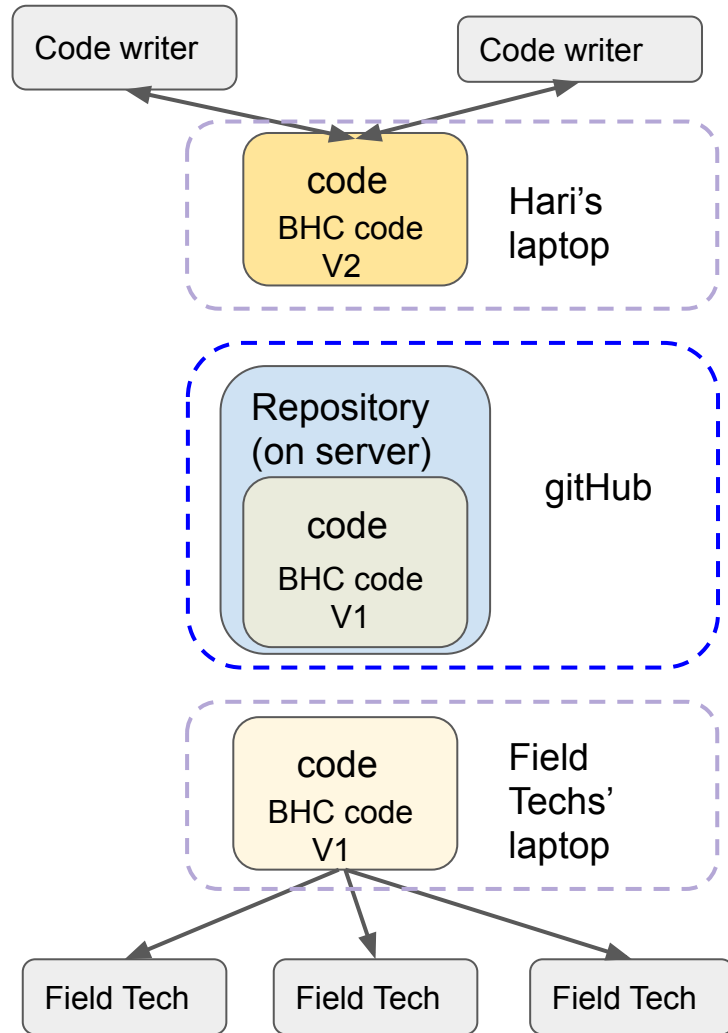


**GitHub** is a website that hosts Git repositories online, making it easier for developers to share code.

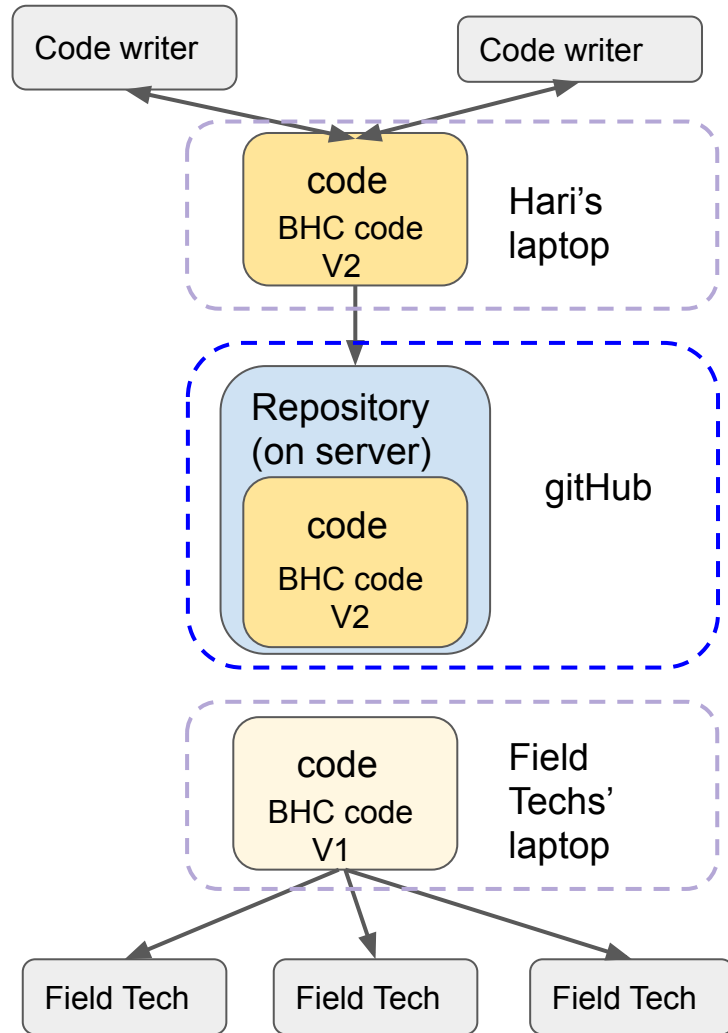


**Repositories** (or "repos") are folders which contain intentional snapshots of progress called commits.

With git you can use a version of the software while Hari and Ramitha modify a copy of the code.

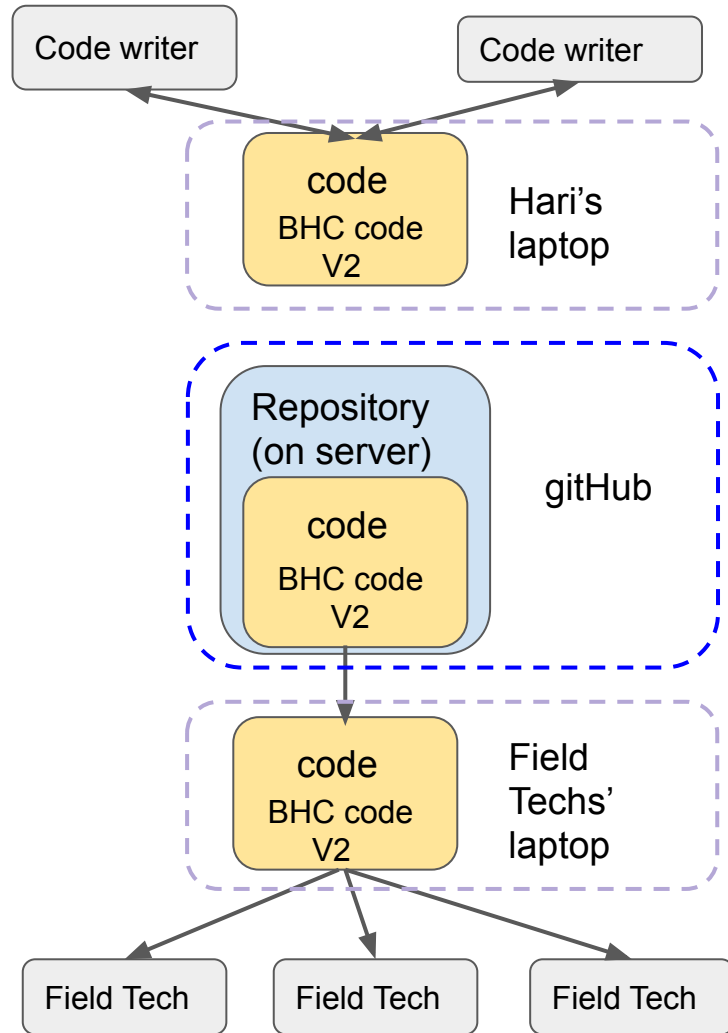


When Ramitha and Hari are ready, they can push their code to the server



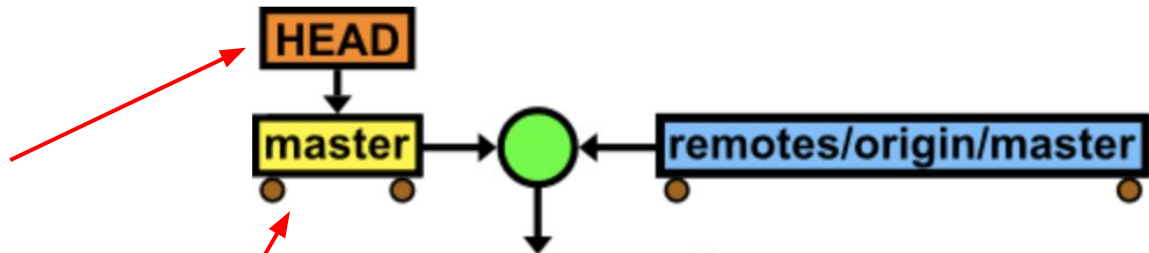


Then you can pull  
the code and use it



# Origin and head can be considered to be the same for this gitHub

`HEAD` is actually a special type of reference that points to another reference. It may point to `master` or it **may not** (it will point to whichever branch is currently checked out).



`master` is the default branch, it's different than head or origin. It's a special branch the rest of the branches are by default merged with it.



# Lab

Using git

# Lab: Cloning a git Repository

In this lab you:

Go to the gitHub for the course

# Go to the GitHub for this course

Go to <https://github.com/cagBRT>

# Select the gitCourse repository

Overview

Repositories 1

Projects 0

Stars 0

Followers 0

Following 0

## Popular repositories


**gitCourse**


A short course to teach how to use Git.


Click here


# The repository will have code called Magic 8 Ball

A short course to teach how to use Git.

 2 commits

 2 branches

 0 releases

 1 contributor

Branch: master ▾

New pull request

Find File

Clone or download ▾

 **cagBRT** Create Magic 8 Ball ...

Latest commit a7bf201 19 minutes ago

 **README.md**

## gitCourse

---

A short course to teach how to use Git.

# Download the repository

Click here



Find File

Clone or download ▼

## Clone with HTTPS ⓘ

Use Git or checkout with SVN using the web URL.

`https://github.com/cagBRT/gitCourse.g`



Open in Desktop

Download ZIP



# Sign in if you are prompted

Find File

Clone or download ▾

## Downloading.

Want to be notified of new releases in  
**cagBRT/gitCourse?**

[Sign in](#)

[Sign up](#)



## Sign in to GitHub

Username or email address

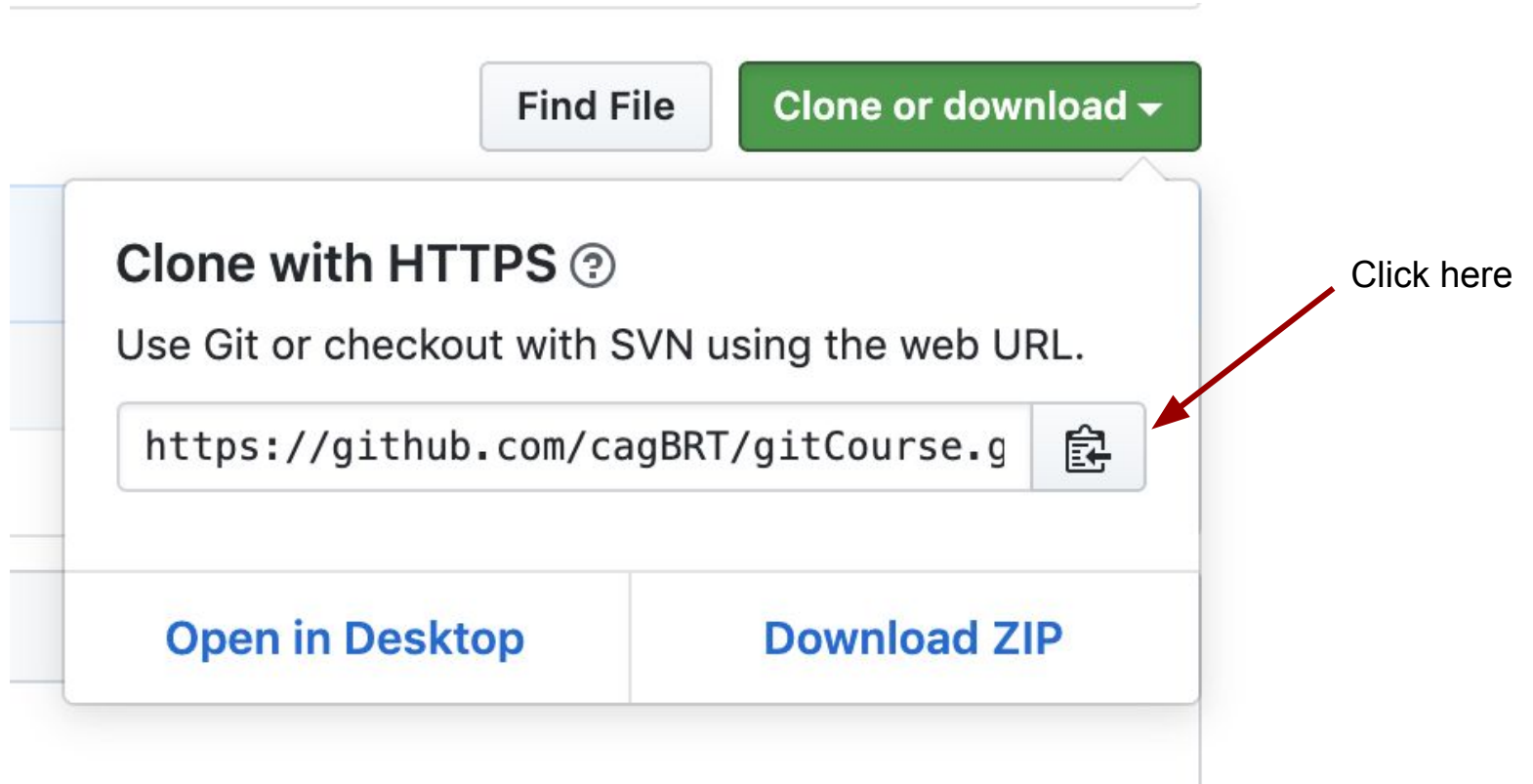
Password

[Forgot password?](#)

[Sign in](#)

New to GitHub? [Create an account.](#)

# Clone with HTTPS




The image shows a GitHub interface with a 'Clone or download' button. A dropdown menu is open, displaying the 'Clone with HTTPS' option. A red arrow points to the copy icon next to the URL, with the text 'Click here' next to it.

Find File Clone or download ▼

**Clone with HTTPS** ⓘ

Use Git or checkout with SVN using the web URL.

`https://github.com/cagBRT/gitCourse.g` 

Open in Desktop Download ZIP

Click here

# Save the URL

The URL is now copied into your clipboard.

It should look similar to this:

<https://github.com/cagBRT/gitCourse.git>



# Lab

Clone the repository to your directory

# Check the contents of gitCourse

Use **ls** to check that gitCourse is an empty directory.

```
BRTs-MBP:gitCourse catherine.gamboa$ ls gitCourse  
BRTs-MBP:gitCourse catherine.gamboa$
```

If gitCourse is not empty, the next step will give you a fatal error message. You learn how to handle this error soon.

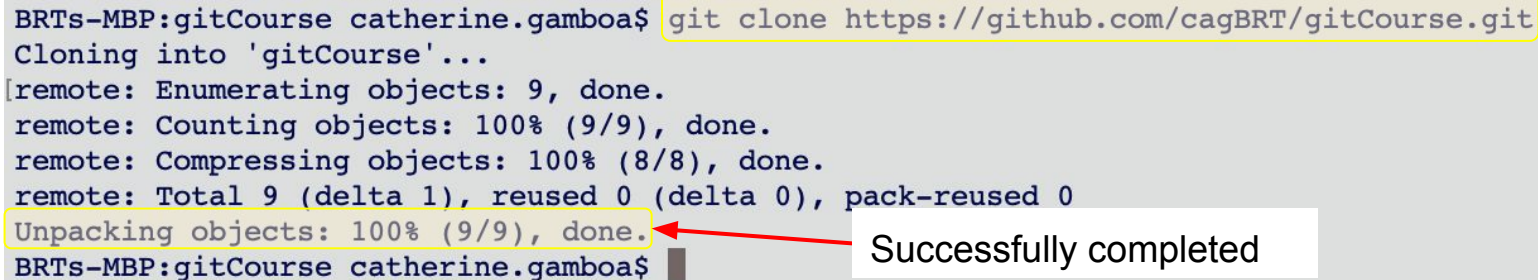
# Use the 'git clone' command to copy the repository to your directory

1. In your terminal window type **git clone <repository URL>**

Paste the repository URL into the terminal after the **git clone** command

2. Then **<return>**

git clone git-repository



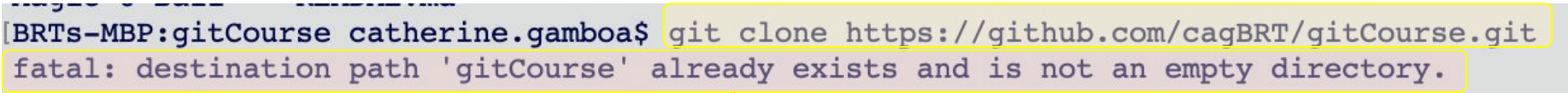
The screenshot shows a terminal window with the following text:   
BRTs-MBP:gitCourse catherine.gamboa\$ git clone https://github.com/cagBRT/gitCourse.git   
Cloning into 'gitCourse'...   
[remote: Enumerating objects: 9, done.   
remote: Counting objects: 100% (9/9), done.   
remote: Compressing objects: 100% (8/8), done.   
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0   
Unpacking objects: 100% (9/9), done.   
BRTs-MBP:gitCourse catherine.gamboa\$   
A red arrow points from the text 'git clone git-repository' to the URL 'https://github.com/cagBRT/gitCourse.git'. Another red arrow points from the text 'Successfully completed' to the line 'Unpacking objects: 100% (9/9), done.'.

```
BRTs-MBP:gitCourse catherine.gamboa$ git clone https://github.com/cagBRT/gitCourse.git
Cloning into 'gitCourse'...
[remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.
BRTs-MBP:gitCourse catherine.gamboa$
```

Successfully completed

If your directory is not be empty, you will get this error message

git clone git-repository



```
[BRTs-MBP:gitCourse catherine.gamboa$ git clone https://github.com/cagBRT/gitCourse.git  
fatal: destination path 'gitCourse' already exists and is not an empty directory.
```

Error! You will need to do a few more steps to fix this.  
You learn to fix this later in the course.

# Check that the repository was correctly copied

Use **ls gitCourse** to check the contents of gitCourse

There should be a file called **Magic\_8\_Ball.py** and one called README.md

```
BRTs-MBP:~ catherine.gamboa$ ls gitCourse  
Magic_8_Ball.py README.md
```



# If your repository copied correctly, you are ready to use the code

1. Go to the gitCourse directory (`cd ~/gitCourse`)
2. **`python Magic_8_Ball.py`**
3. Ask the magic 8 ball a question and it will give you an answer.

```
BRTs-MBP:gitCourse catherine.gamboa$ python Magic_8_Ball.py
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
My sources say no
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
My sources say no
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
Reply hazy, try again
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
Concentrate and ask again
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
You may rely on it
Ask the magic 8 ball a question: (press enter to quit)
BRTs-MBP:gitCourse catherine.gamboa$
```

Wow! Everyone will love this course!

\*ignore the other answers

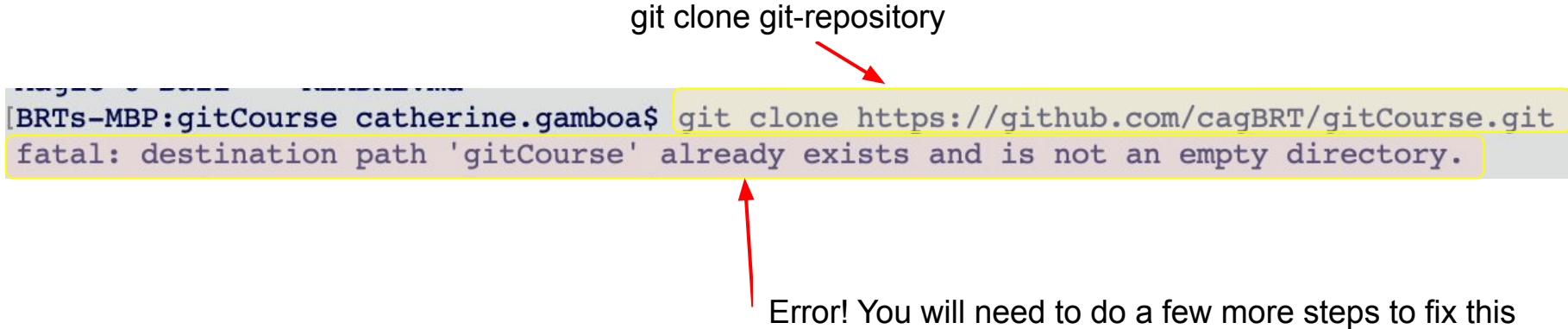


# Lab

Fatal Error: Directory already exists

If this happens to you, don't panic, it can be fixed

git clone git-repository



```
[BRTs-MBP:gitCourse catherine.gamboa$ git clone https://github.com/cagBRT/gitCourse.git  
fatal: destination path 'gitCourse' already exists and is not an empty directory.]
```

Error! You will need to do a few more steps to fix this

# You will need to do a pull request and get the code again

1. Check which branch you are on: `git branch -a`

```
BRTs-MBP:gitCourse catherine.gamboa$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/cagBRT-patch-1
remotes/origin/master
```

# You will need to do a pull request and get the code

1. Checkout the branch : **git checkout origin/master**
2. Check that you are on the right branch: **git branch -a**
3. Check the **git remote -v**

```
BRTs-MBP:gitCourse catherine.gamboa$ git remote -v  
origin  https://github.com/cagBRT/gitCourse.git (fetch)  
origin  https://github.com/cagBRT/gitCourse.git (push)
```

4. Get the new copy of the code: **git pull origin master**



# Lab

Check the status of your code

# Check that your code has not been altered

Check that the code on your computer has not been altered: **git status**

```
[BRTs-MBP:gitCourse catherine.gamboa$ git status  
On branch origin/master  
nothing to commit, working tree clean
```

If you see this, you know the code you are using matches the source code on the repository.

# If the code on your local machine has been altered

Check that your code has not been altered: **git status**

```
BRTs-MBP:gitCourse catherine.gamboa$ git status
On branch origin/master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   Magic_8_Ball.py

no changes added to commit (use "git add" and/or "git commit -a")
```

If you see this, you know the code you are using does not match the source code. You will need to get a fresh copy of the code.



# Use **git reset --hard** to get a fresh copy of the code

```
[BRTs-MBP:gitCourse catherine.gamboa$ git status
On branch origin/master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes)
```

You ran **git status**

```
modified:   Magic_8_Ball.py
```

You found out your code is out of sync with the source

```
no changes added to commit (use "git add" and/or "git commit -a")
```

Run **git reset --hard**

```
[BRTs-MBP:gitCourse catherine.gamboa$ git reset --hard
```

```
HEAD is now at e2c64bc Update Magic_8_Ball.py
```

```
[BRTs-MBP:gitCourse catherine.gamboa$ git status
```

Run **git status**

```
On branch origin/master
```

```
nothing to commit, working tree clean
```

Your code is back in sync!

\*Yes that is two dashes in **git reset --hard**



# Lab

Check that you have the latest version of the code

# Lab: Before using the code, check that you have the latest version of it

Check that code is the latest version: **git remote show origin**

```
BRTs-MBP:gitCourse catherine.gamboa$ git remote show origin
* remote origin
  Fetch URL: https://github.com/cagBRT/gitCourse.git
  Push  URL: https://github.com/cagBRT/gitCourse.git
  HEAD branch: master
  Remote branches:
    cagBRT-patch-1 tracked
    master          tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (local out of date)
```

Do you not have the latest version of the code

# Lab: Get the newest version of the code

1. Check the status

```
[BRTs-MBP:gitCourse catherine.gamboa$ git status  
On branch master  
Your branch is up to date with 'remotes/origin/master'.
```

```
nothing to commit, working tree clean
```


2. Reset the local branch

```
[BRTs-MBP:gitCourse catherine.gamboa$ git reset --hard  
HEAD is now at f97cf93 Update Magic_8_Ball.py
```

Follow steps 1 - 6 to get the latest code


# Lab: Get the newest version of the code

## 3. Fetch the new code



```
[BRTs-MBP:gitCourse catherine.gamboa$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

## 4. Check the status



```
From https://github.com/cagBRT/gitCourse
   f97cf93..25269bf  master    -> origin/master
[BRTs-MBP:gitCourse catherine.gamboa$ git status
On branch master
Your branch is behind 'remotes/origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working tree clean
```

# Lab: Get the newest version of the code

5. Pull the new code into you branch

```
BRTs-MBP:gitCourse catherine.gamboa$ git pull origin
```

```
Updating 3e06cbb..55b3035
```

```
Fast-forward
```

```
 Magic_8_Ball.py | 1 +
```

```
 1 file changed, 1 insertion(+)
```

```
BRTs-MBP:gitCourse catherine.gamboa$ git remote show origin
```

```
* remote origin
```

```
Fetch URL: https://github.com/cagBRT/gitCourse.git
```

```
Push URL: https://github.com/cagBRT/gitCourse.git
```

```
HEAD branch: master
```

```
Remote branches:
```

```
  cagBRT-patch-1 tracked
```

```
  master         tracked
```

```
Local branch configured for 'git pull':
```

```
  master merges with remote master
```

```
Local ref configured for 'git push':
```

```
  master pushes to master (up to date)
```

6. Compare the local with the master

You are ready to run the code



# You are ready to use the code

1. Go to the gitCourse directory (`cd ~/gitCourse`)
2. **`python Magic_8_Ball.py`**
3. Ask the magic 8 ball a question and it will give you an answer.

```
BRTs-MBP:gitCourse catherine.gamboa$ python Magic_8_Ball.py
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
My sources say no
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
My sources say no
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
Reply hazy, try again
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
Concentrate and ask again
Ask the magic 8 ball a question: (press enter to quit) Will everyone love my course?
You may rely on it
Ask the magic 8 ball a question: (press enter to quit)
BRTs-MBP:gitCourse catherine.gamboa$
```

Wow! Everyone will love this course!

\*ignore the other answers

# List of commands

[List of Linux and git commands](#)

[URL for Boom Control gitHub](#)

File name for boom control code: **execute\_controllers.py**

Execute code using: **python execute\_controllers.py**



# Questions?

[catherine.gamboa@bluerivert.com](mailto:catherine.gamboa@bluerivert.com)