Linux and git

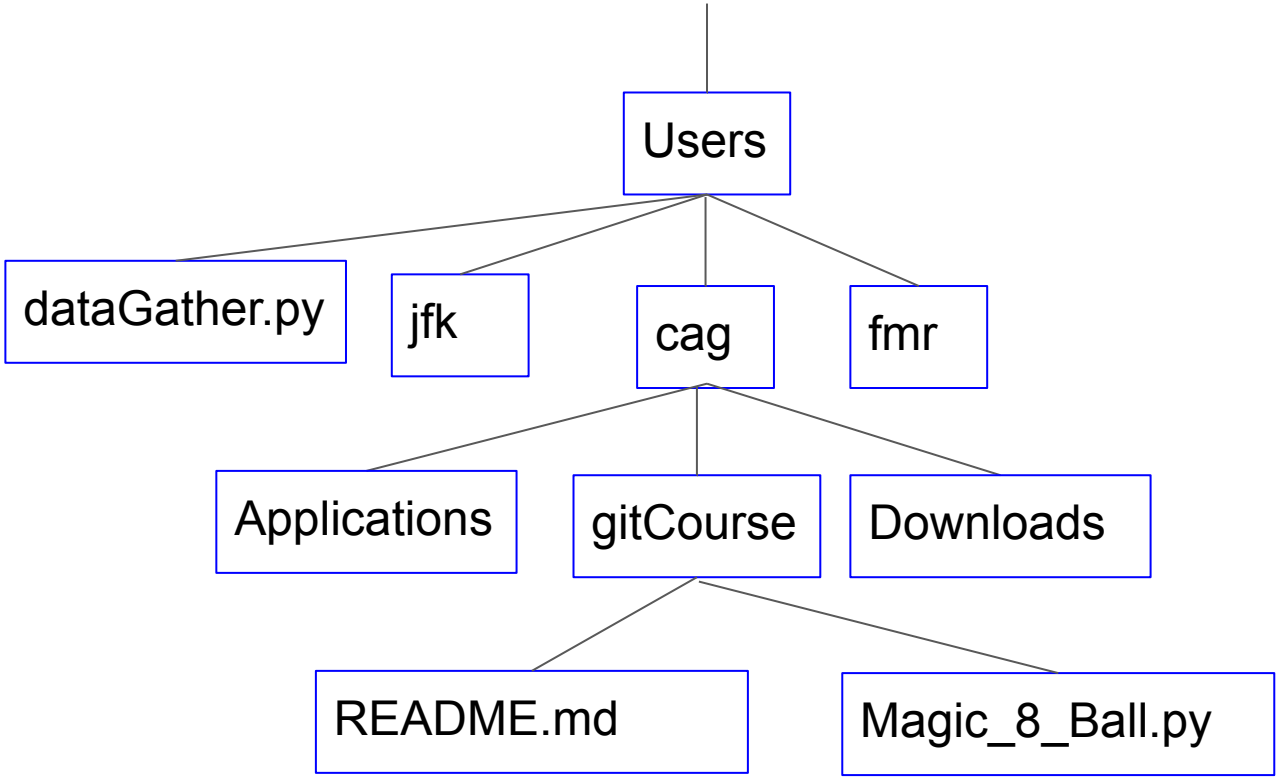
git for Field Technicians

Goals

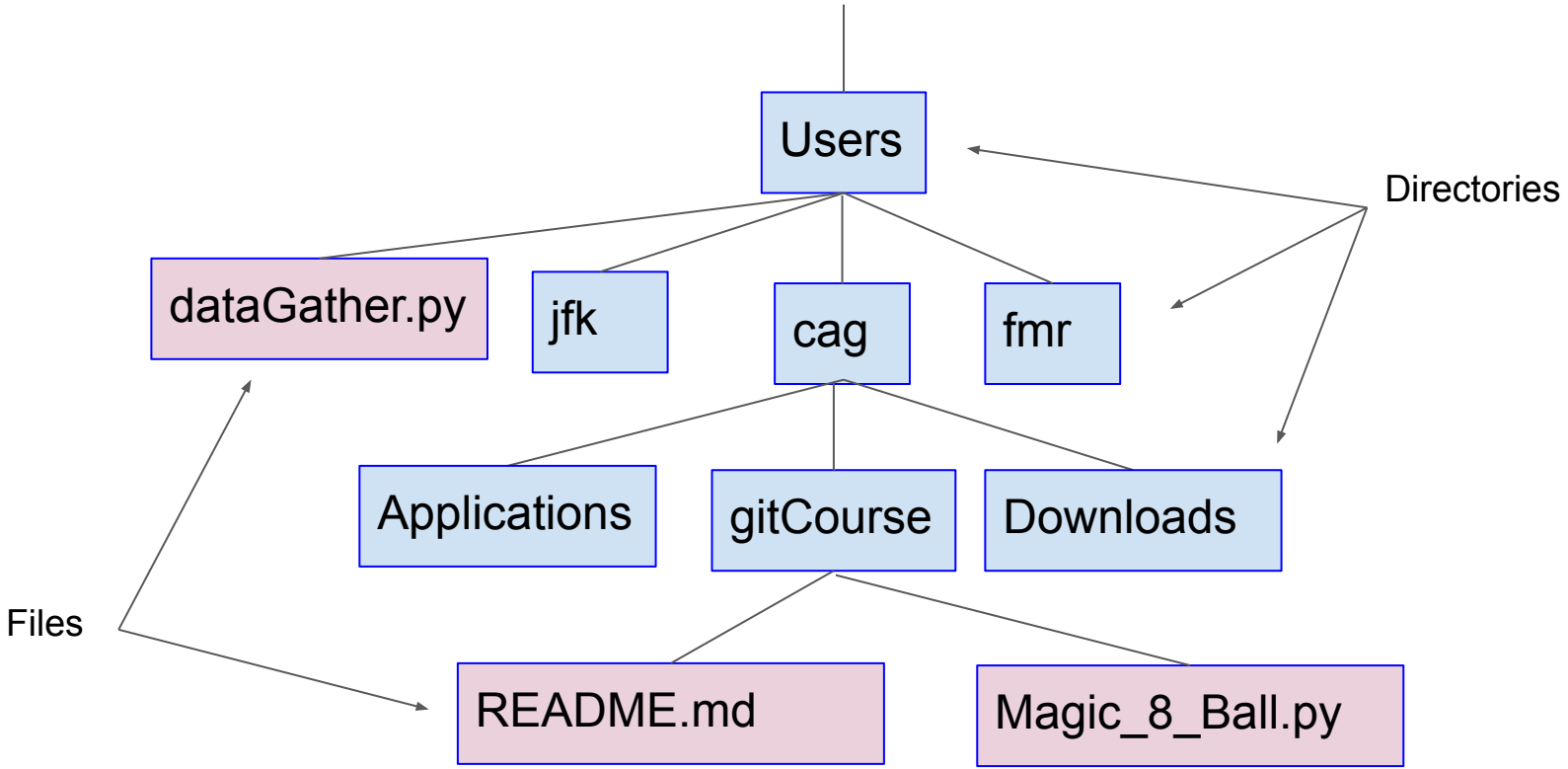
1. Check the status of the code on your computer
2. Check out, fetch, and pull your repository
3. Clone a git repository
4. Create a directory

Introduction to Linux

Linux is structured like a tree

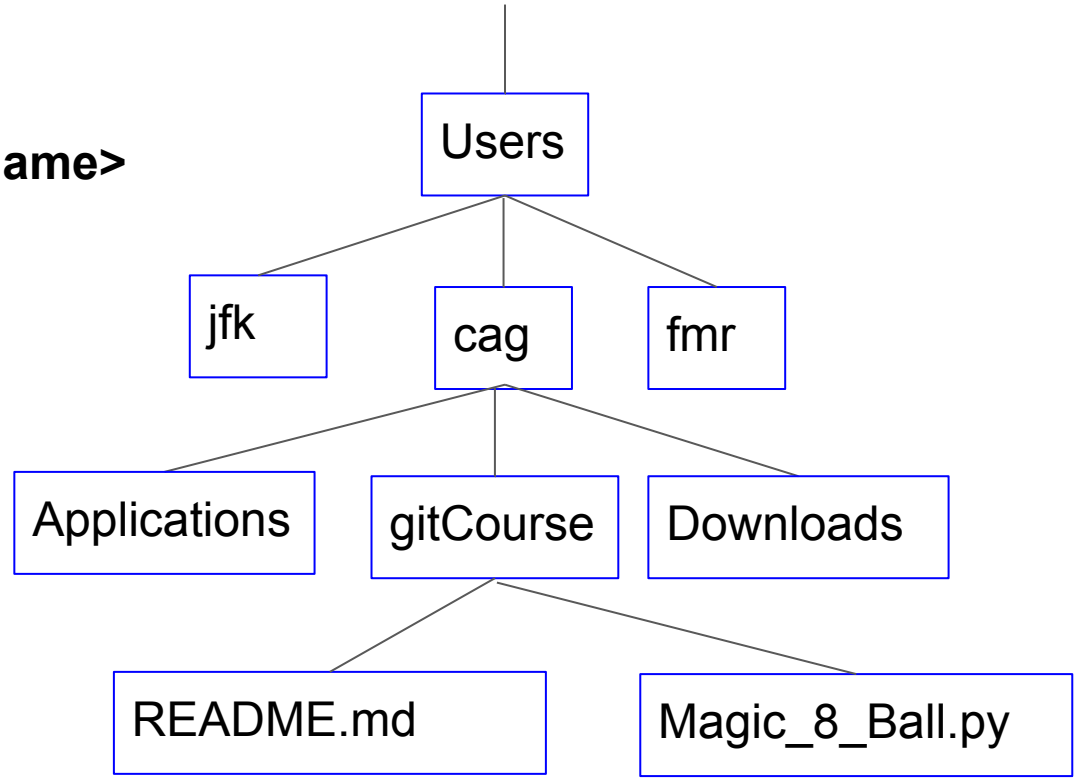


Directories can contain files and directories



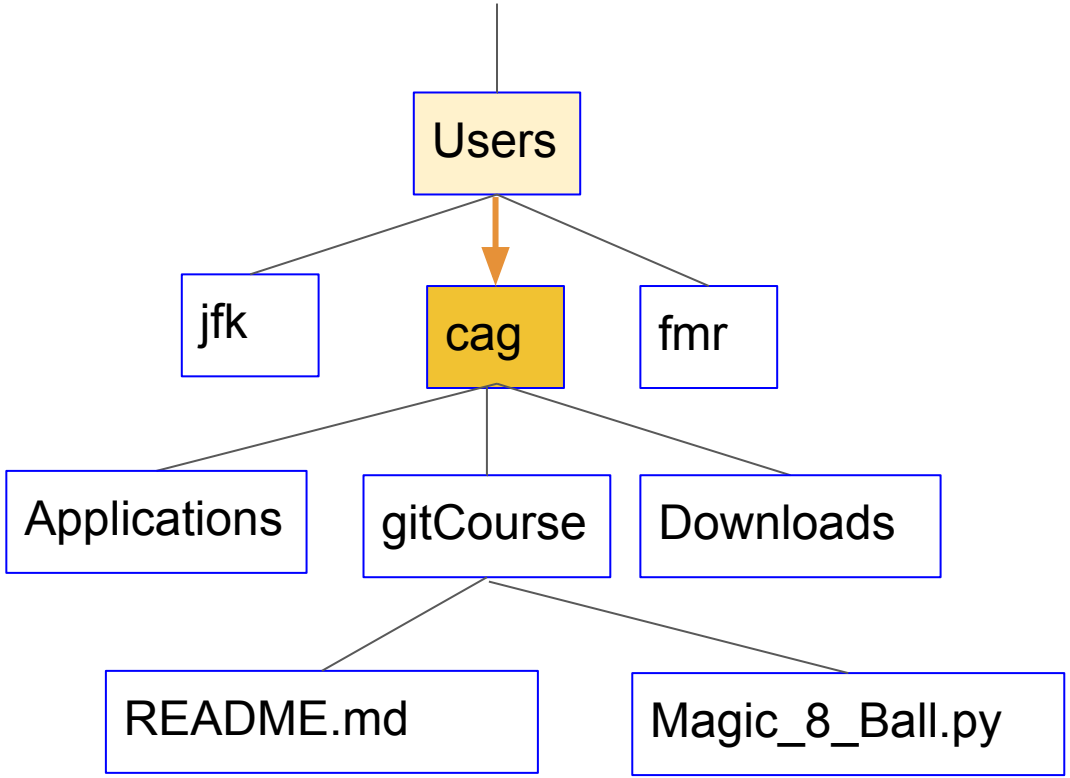
Use the command **cd** to travel up and down the tree

cd <directory name>



cd ..

Traversing down the tree

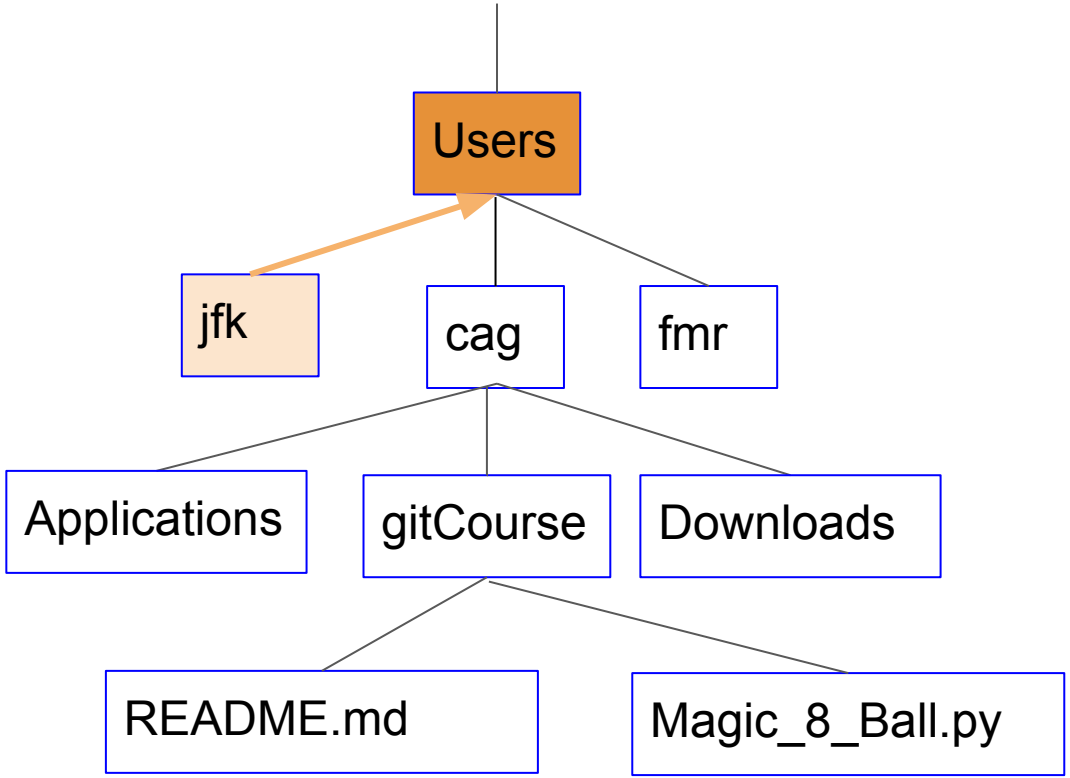


You are in the **Users** directory and you would like to go to the **cag** directory

cd <directory name>

cd cag

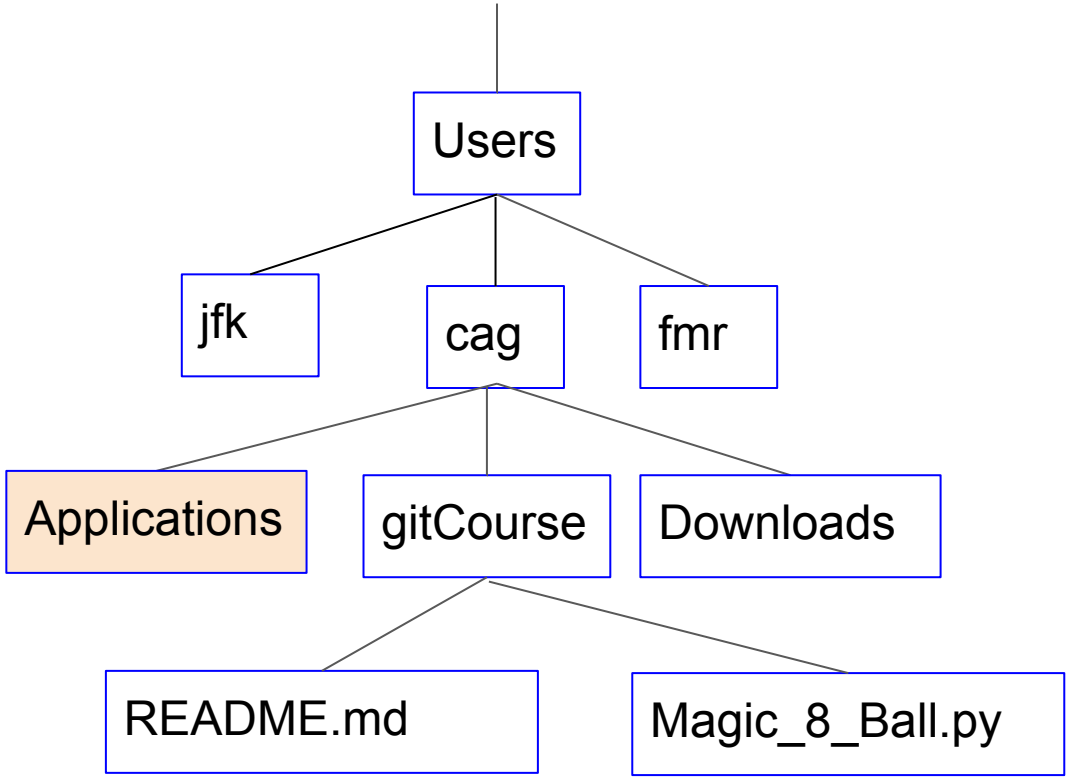
Traversing up the tree



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

cd ..

Finding your location in the tree

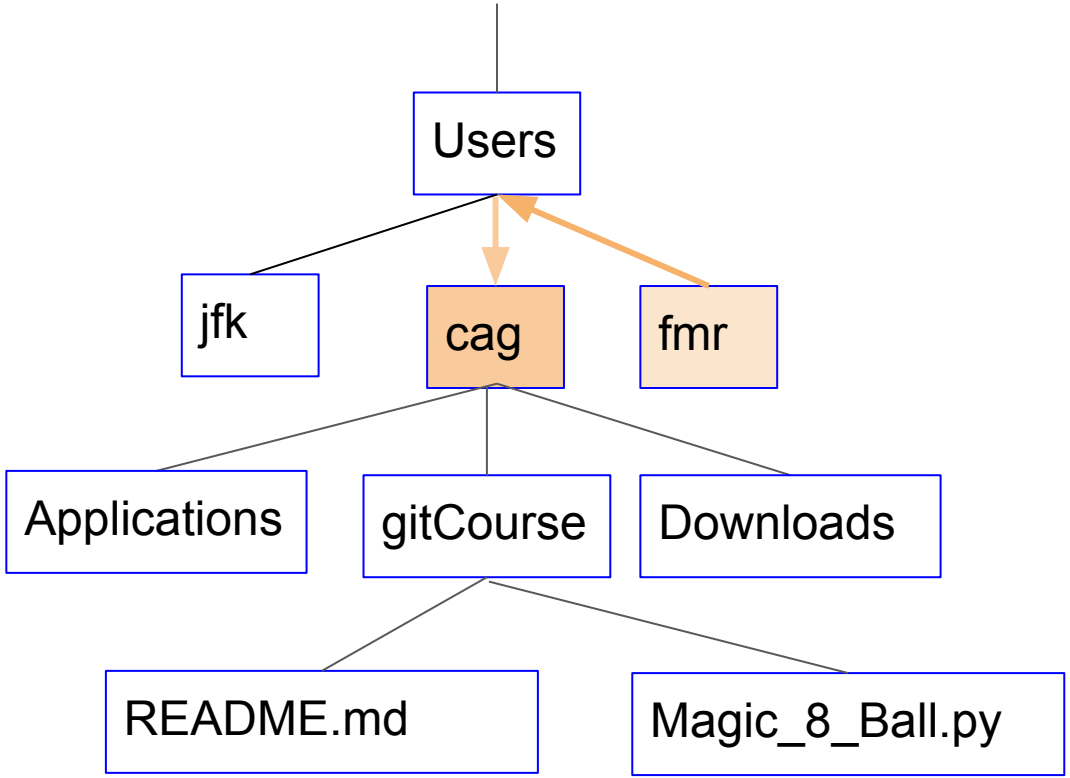


You are working in Linux and you want to know in which directory you are working.

pwd

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

Going home

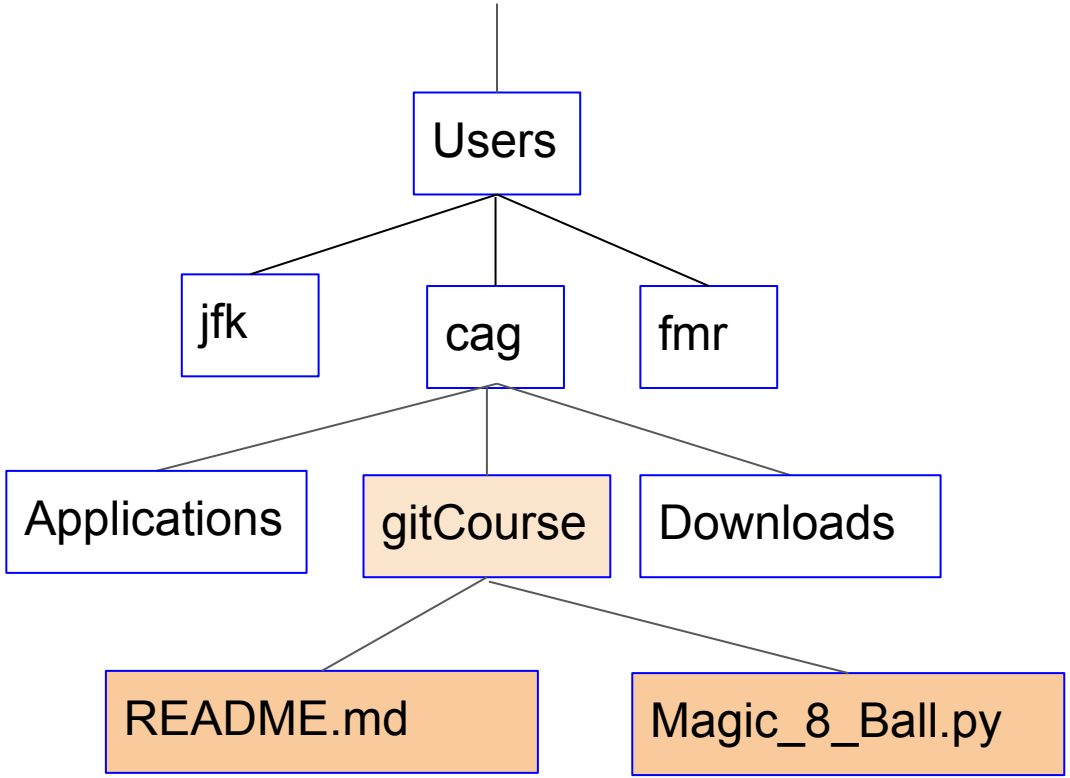


You are working in Linux and you want to go to your home directory

cd ~

You will return to your home directory

Listing directory contents

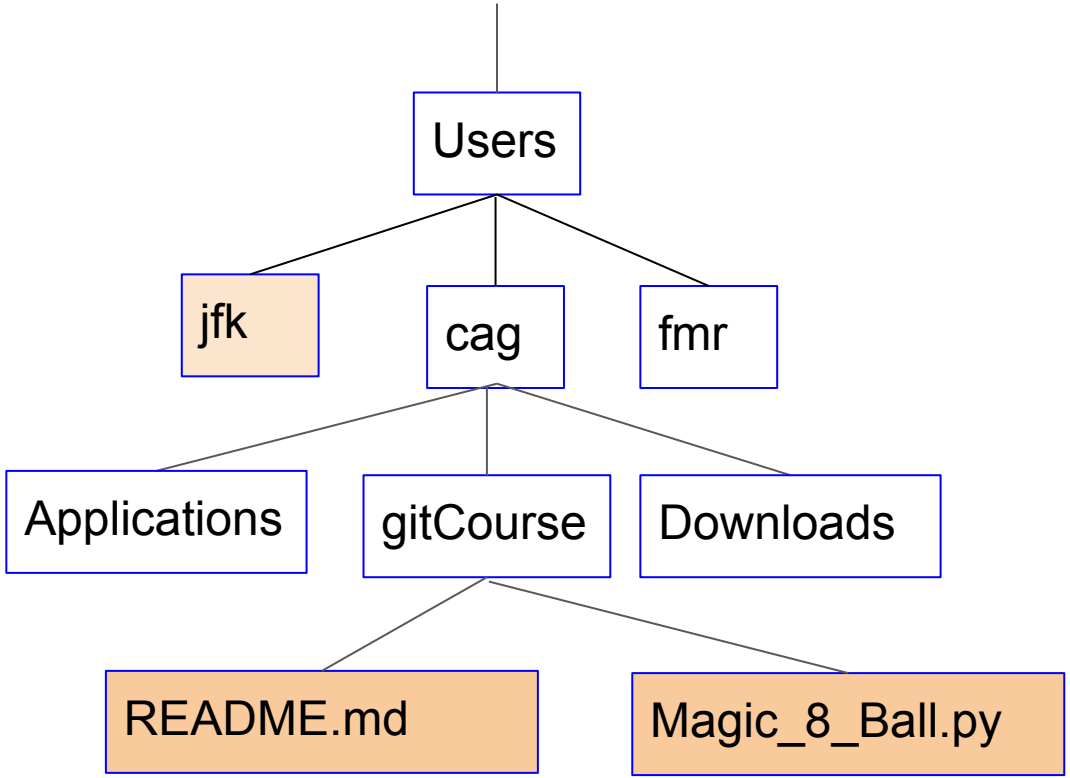


You are working in the gitCourse directory and you want to know what is in the directory.

ls

The contents of gitCourse will be printed on the terminal

Listing another directory's contents

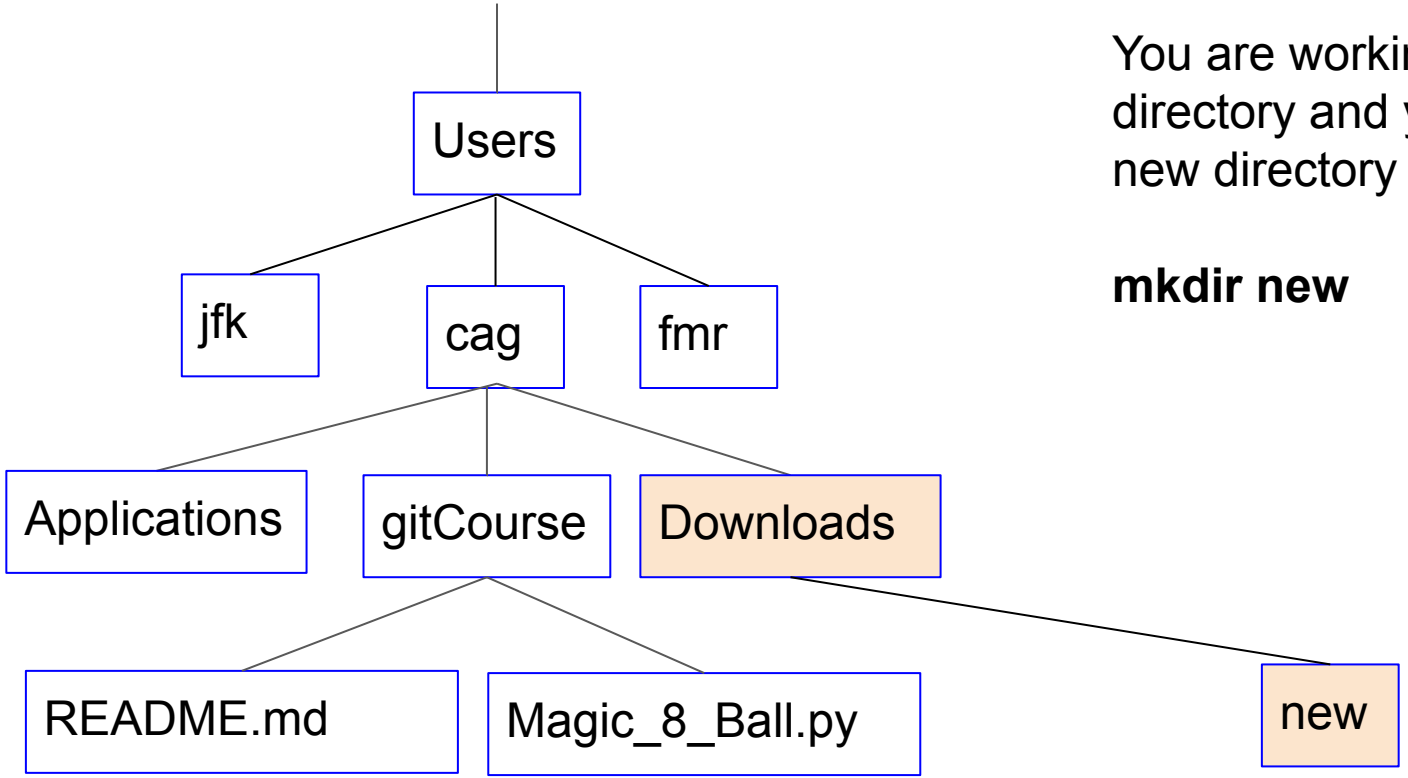


You are working in the **jfk** directory and you want to know what is in the gitCourse directory.

ls ~cag/gitCourse

The contents of gitCourse will be printed on the terminal

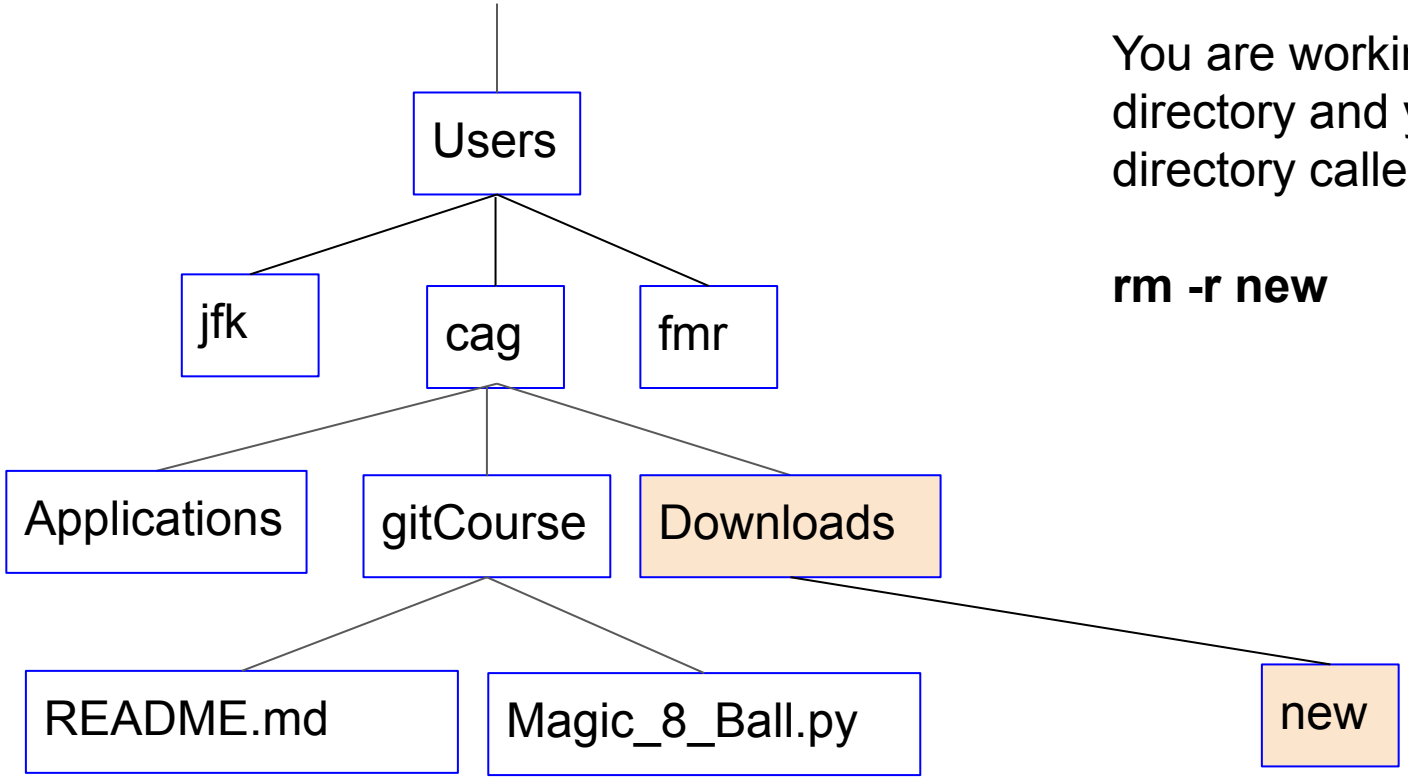
Make a new directory



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

mkdir new

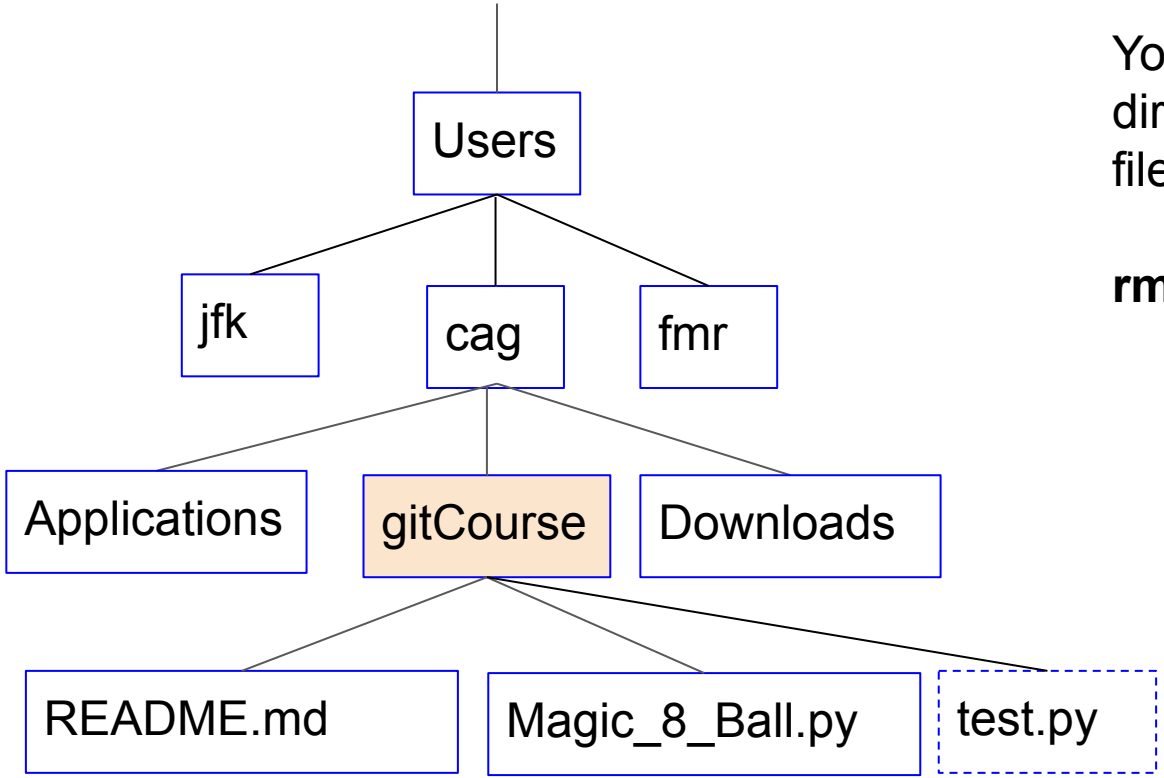
Delete a directory



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

rm -r new

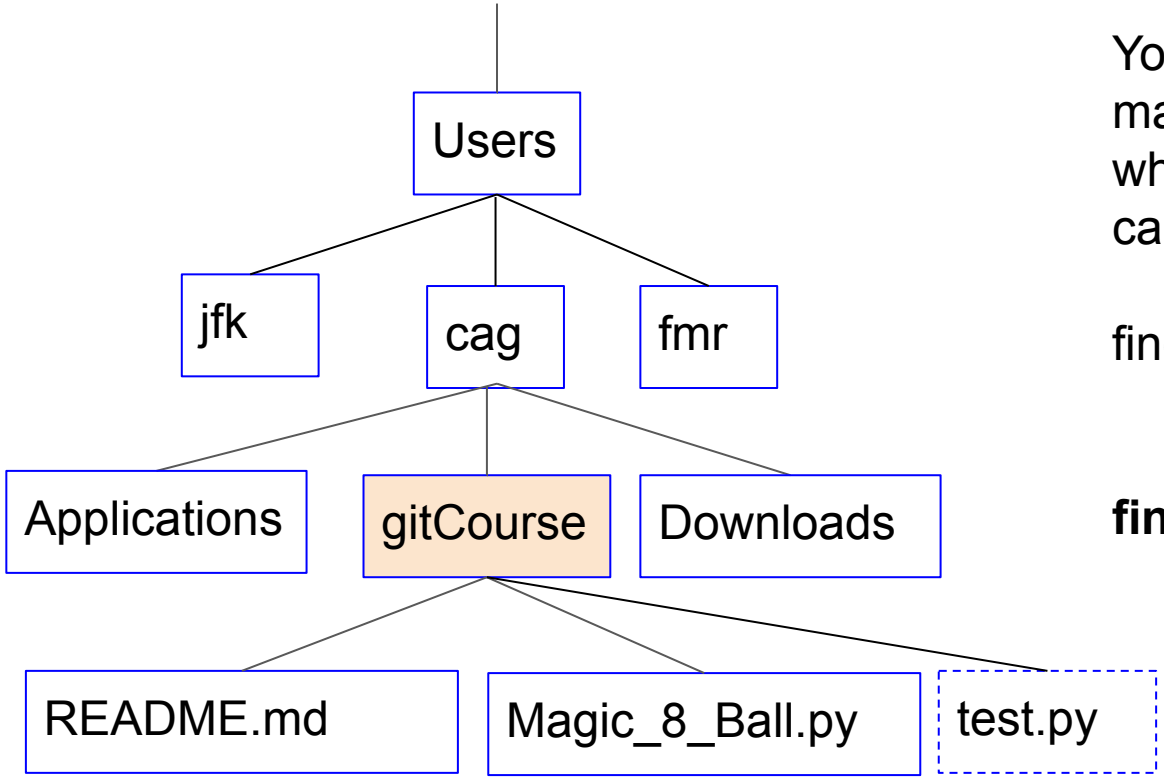
Delete a file



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

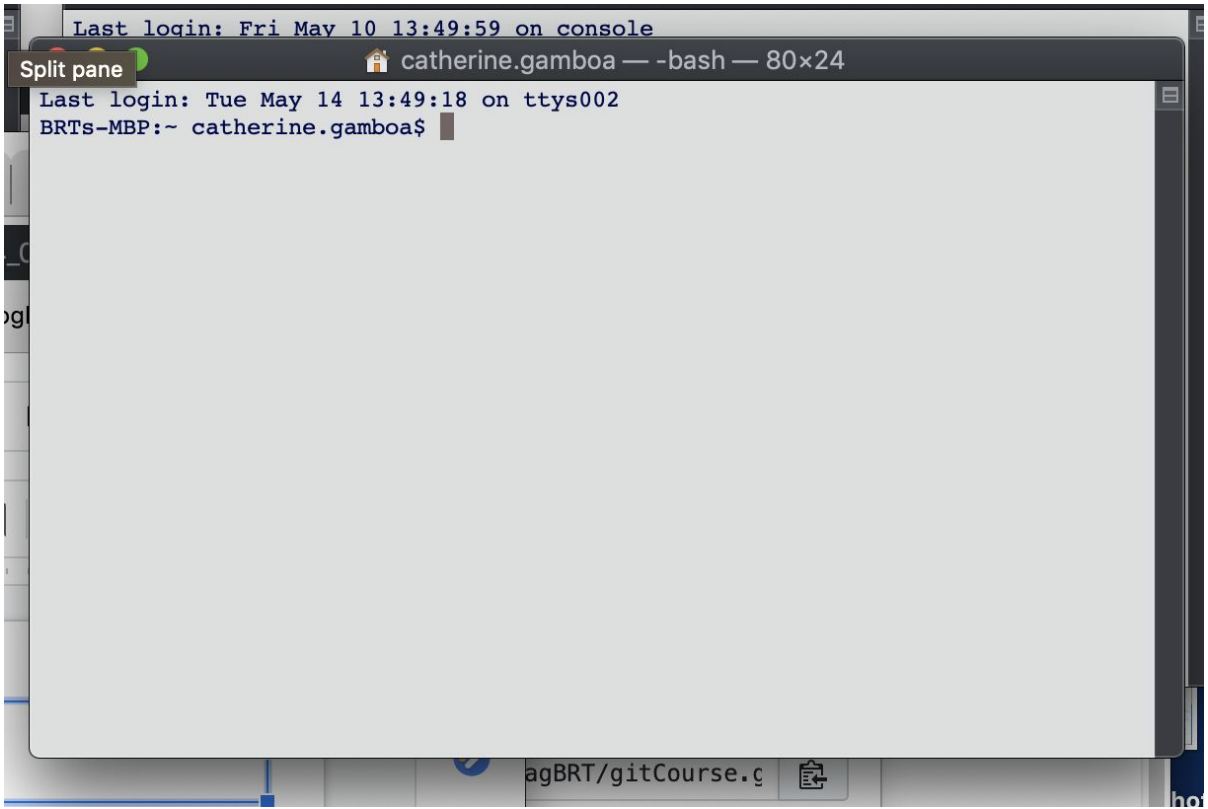
Make the directory if it is not present

Open a terminal window

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

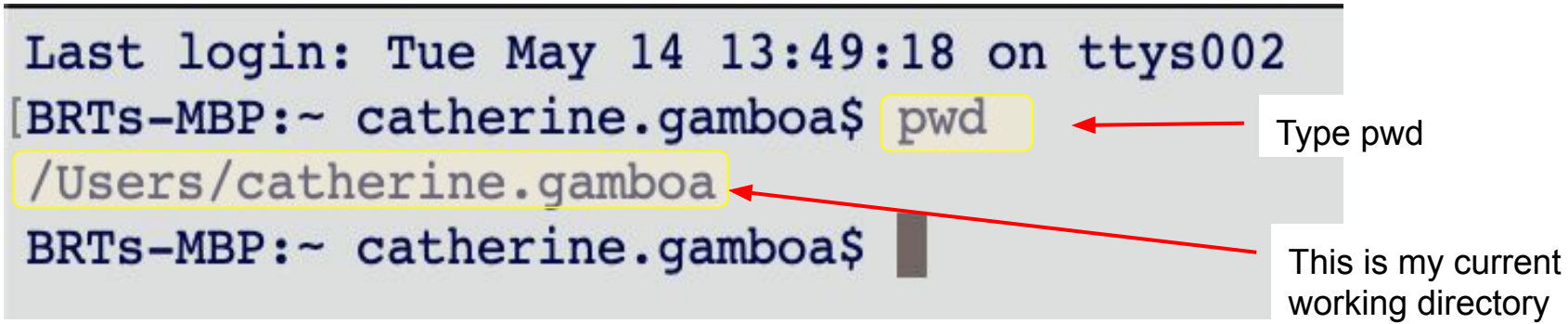
If you like, use the shortcut keys - Ctrl-Cmd-T

Use Linux commands in the terminal window

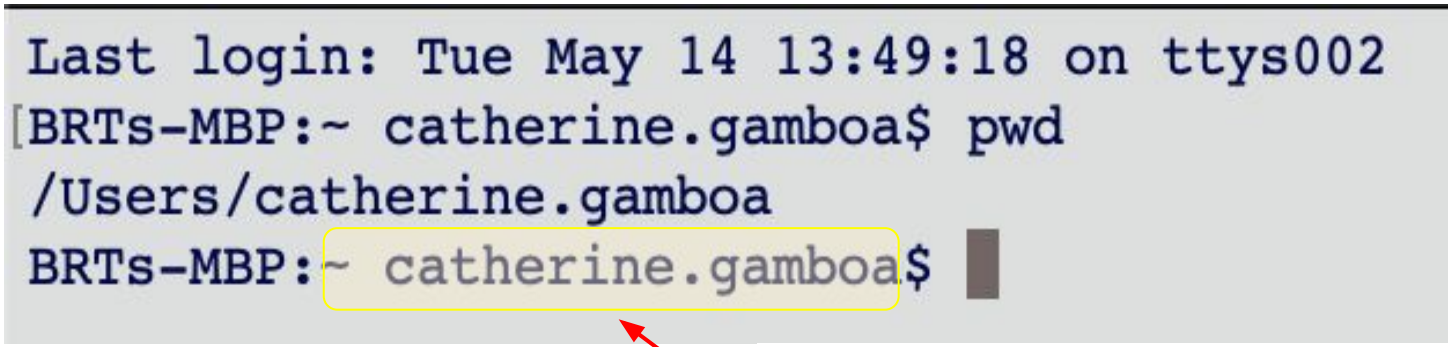


Use the command **pwd** to find where you are in the computer

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

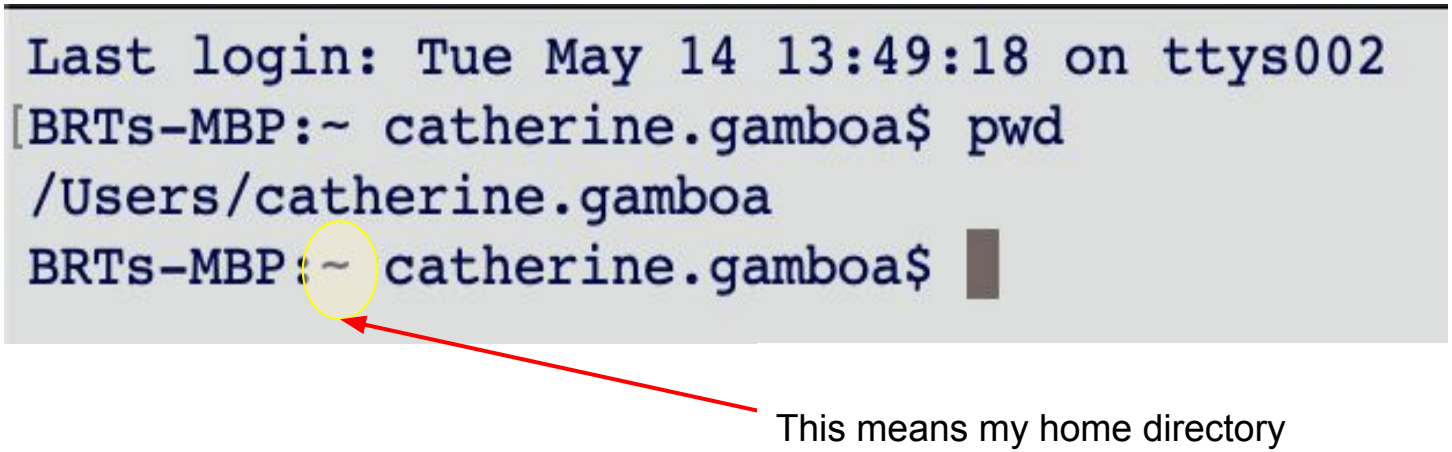


The directory name is part of the Linux prompt



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

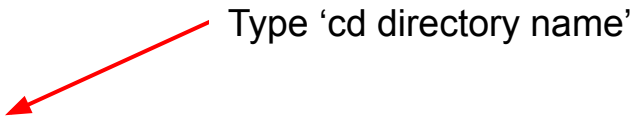
The ~ (tilda) command signifies your home directory



Use the command **cd** to change directories

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

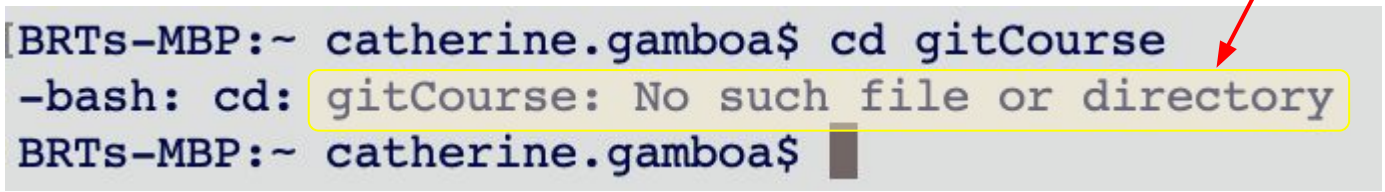
For example cd <directory name>




```
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



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

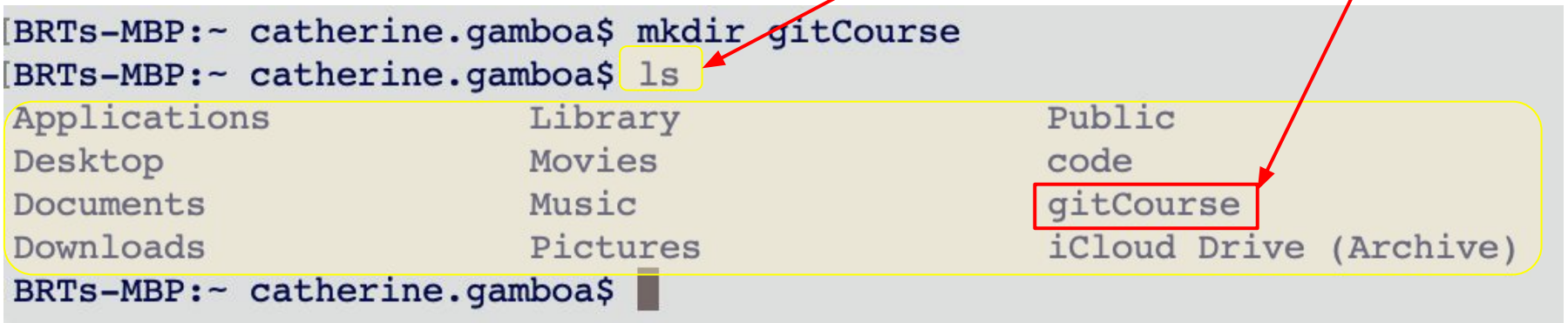
Make the directory called gitCourse

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



Introduction to git

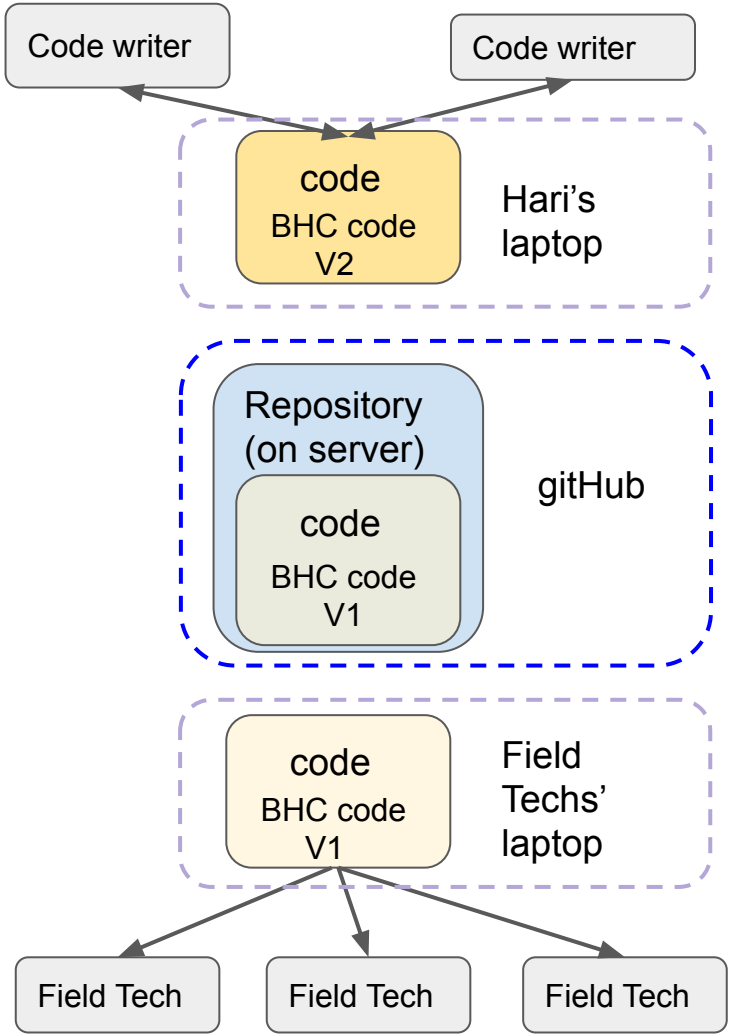
What is Git?

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

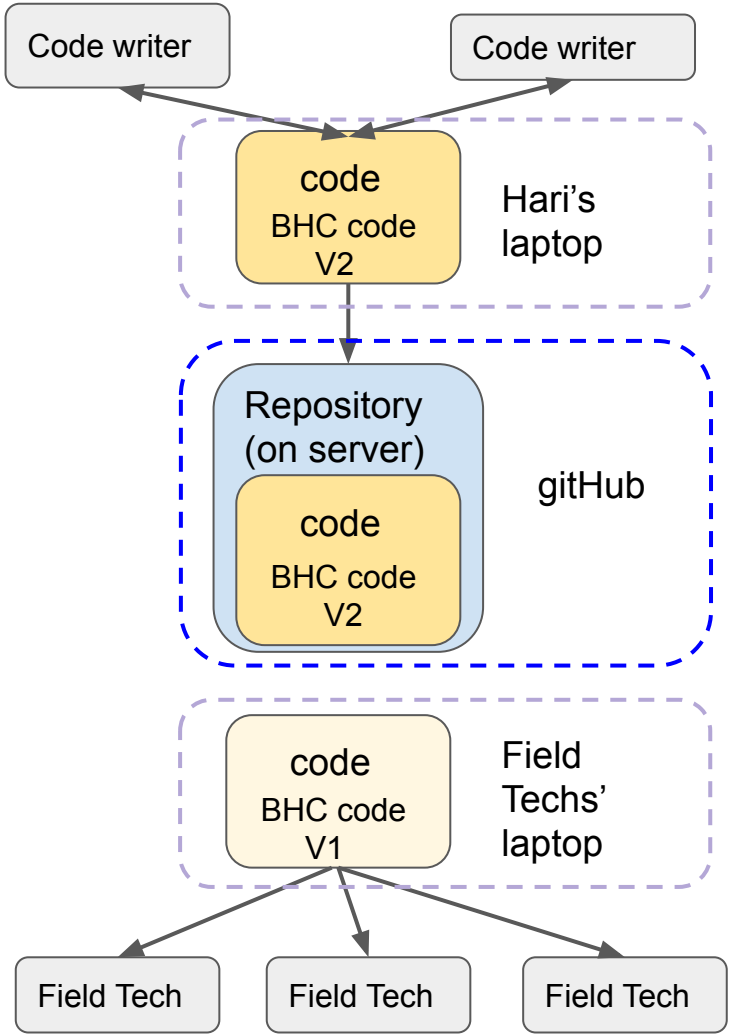
GitHub is a website that hosts Git repositories online, making it easier for developers to share code.

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

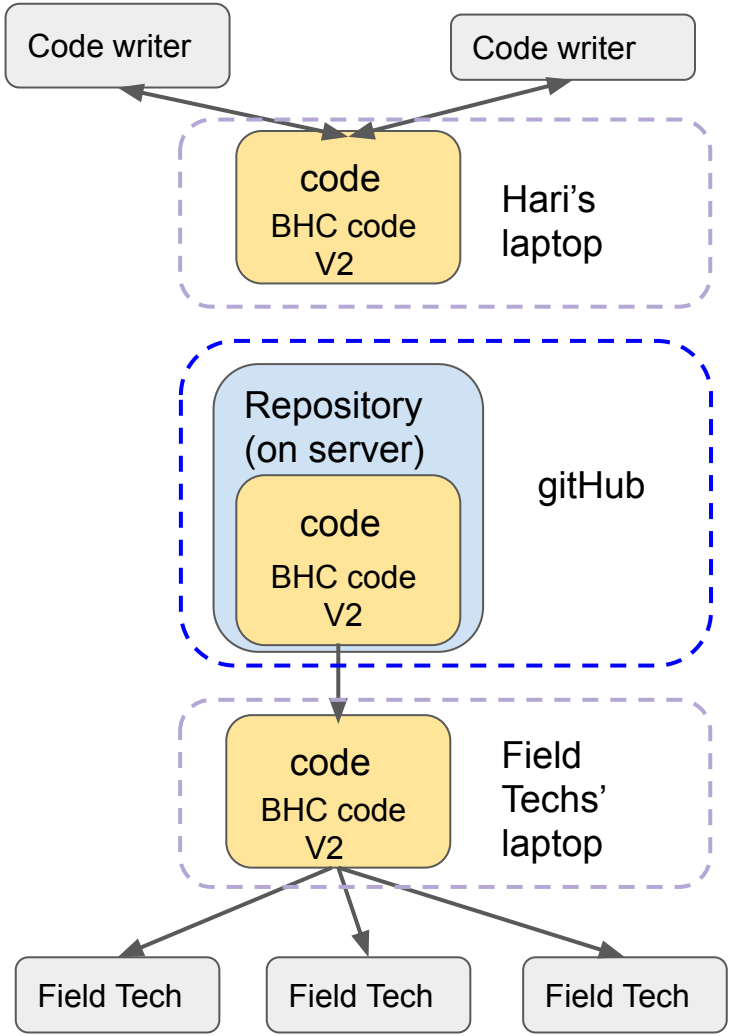
With git you can use a version of the software while Hari and Ramitha modify a copy of the code.



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

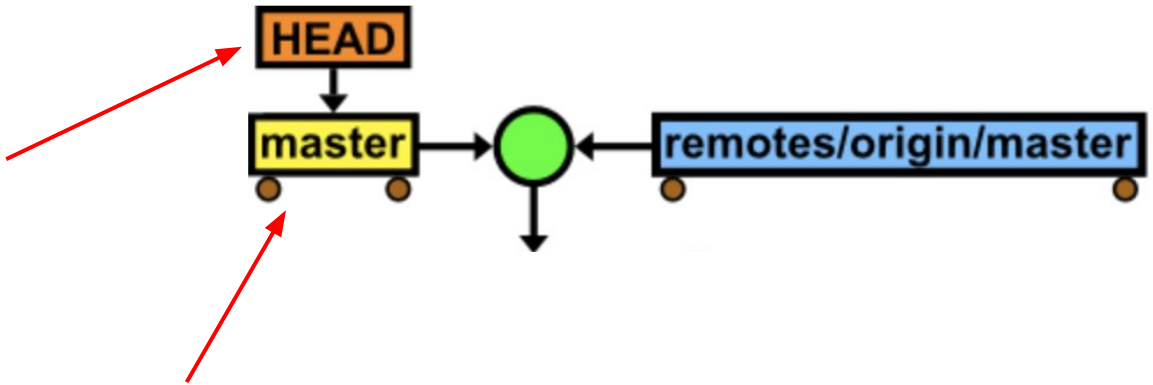


Then you can pull the code and use it



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

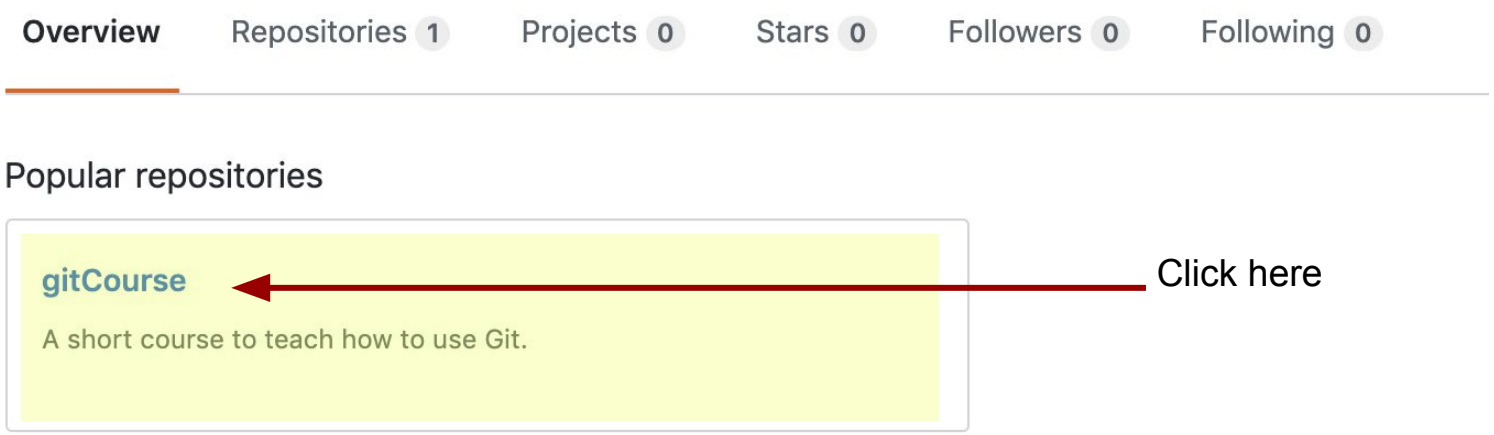
In this lab you:

Go to the gitHub for the course

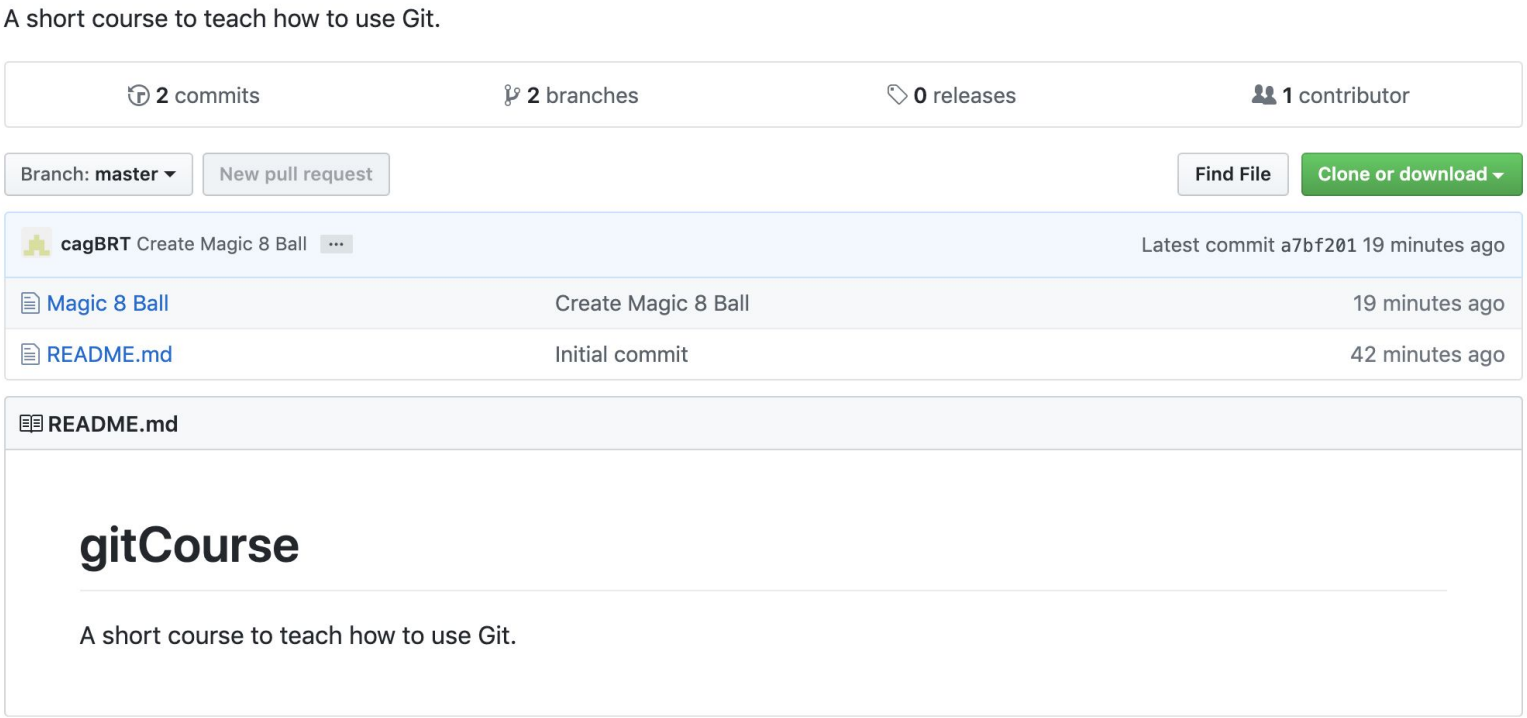
Go to the GitHub for this course

Go to https://github.com/cagBRT

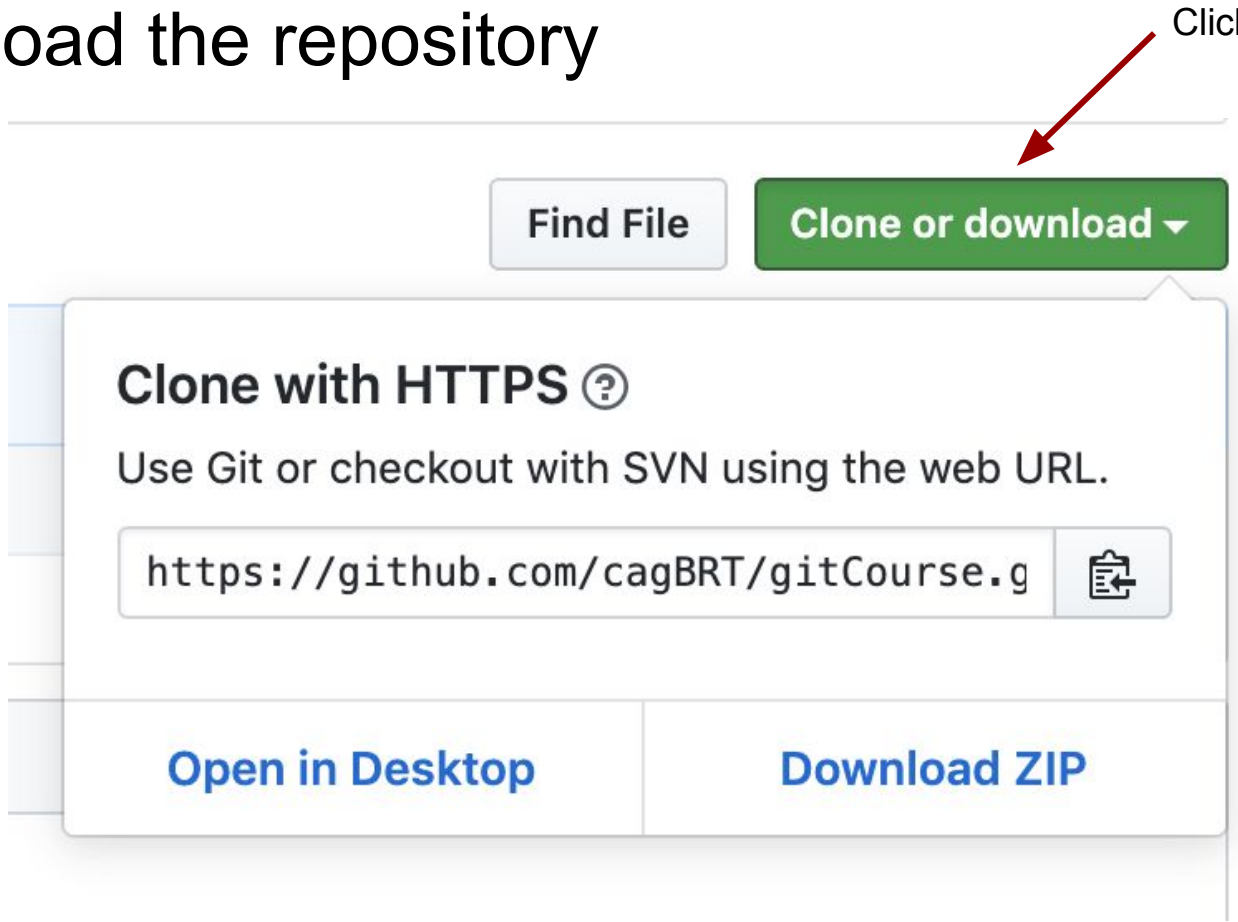
Select the gitCourse repository



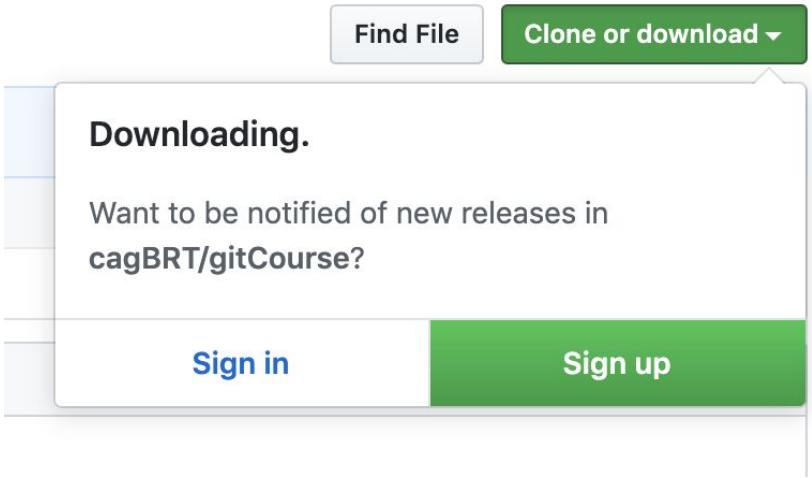
The repository will have code called Magic 8 Ball

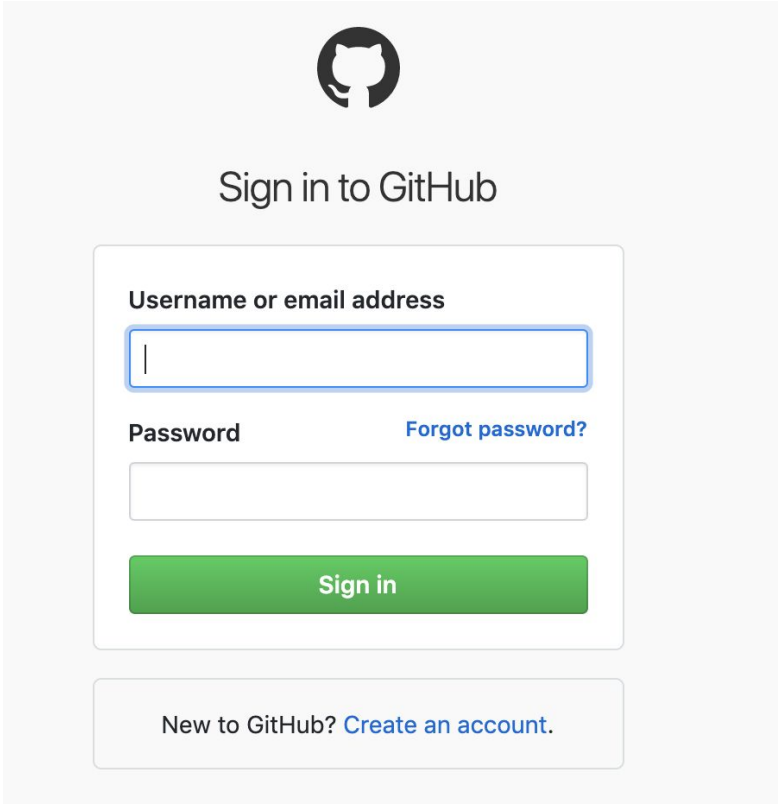


Download the repository

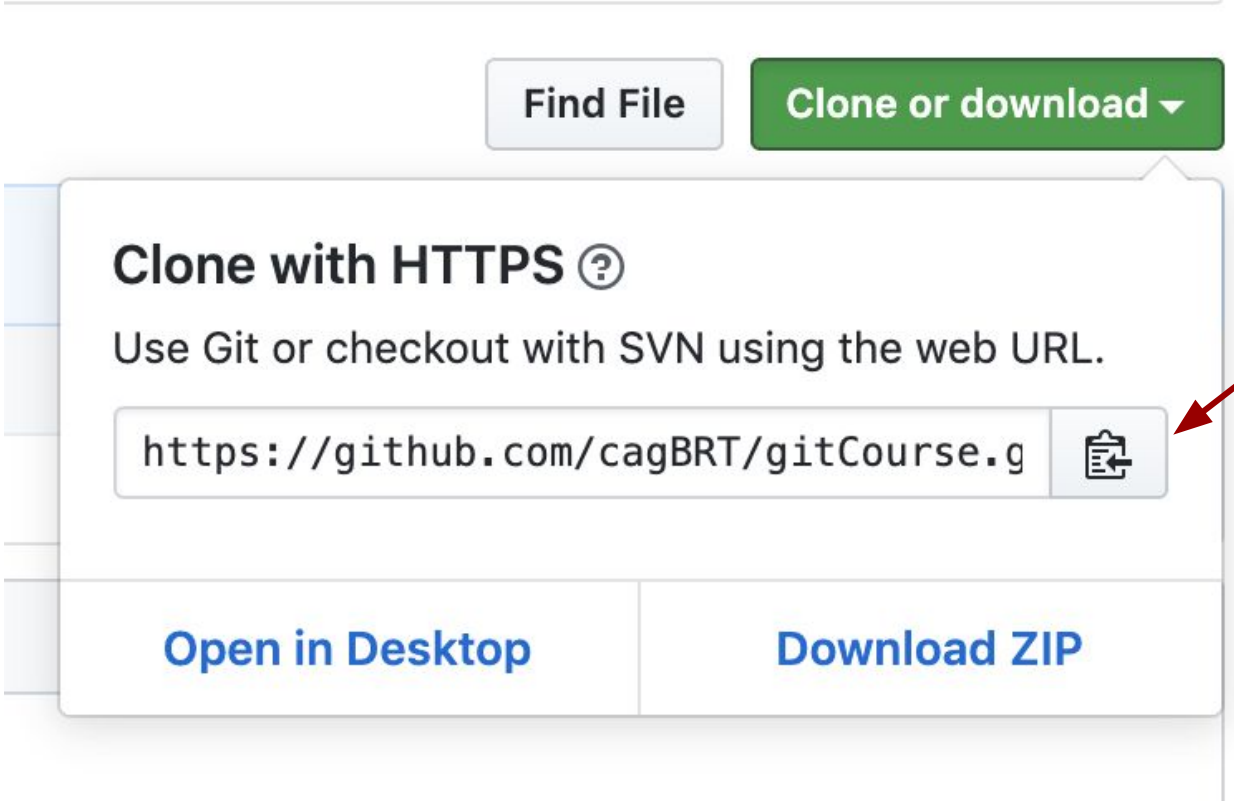


Sign in if you are prompted





Clone with HTTPS



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$
```

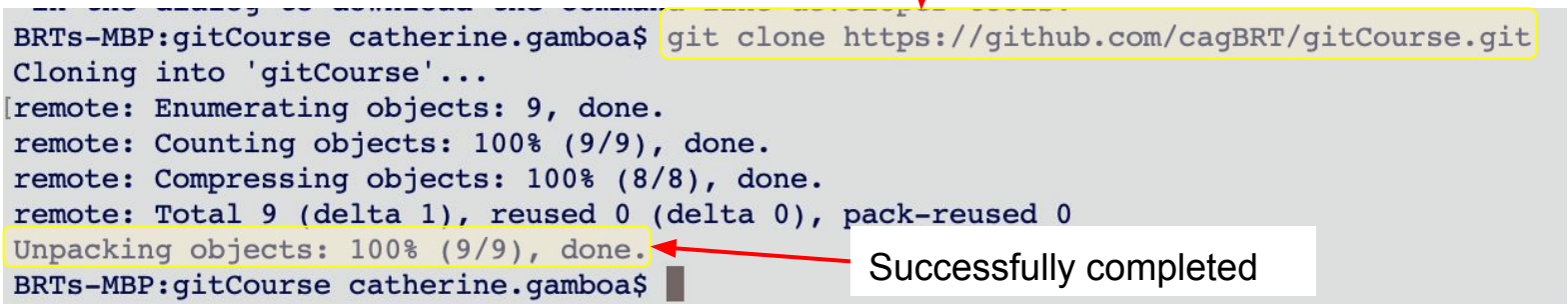
If gitCourse is not empty, the next step will give you a fatal error message. You learn how to handle this error soon.

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**>



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**

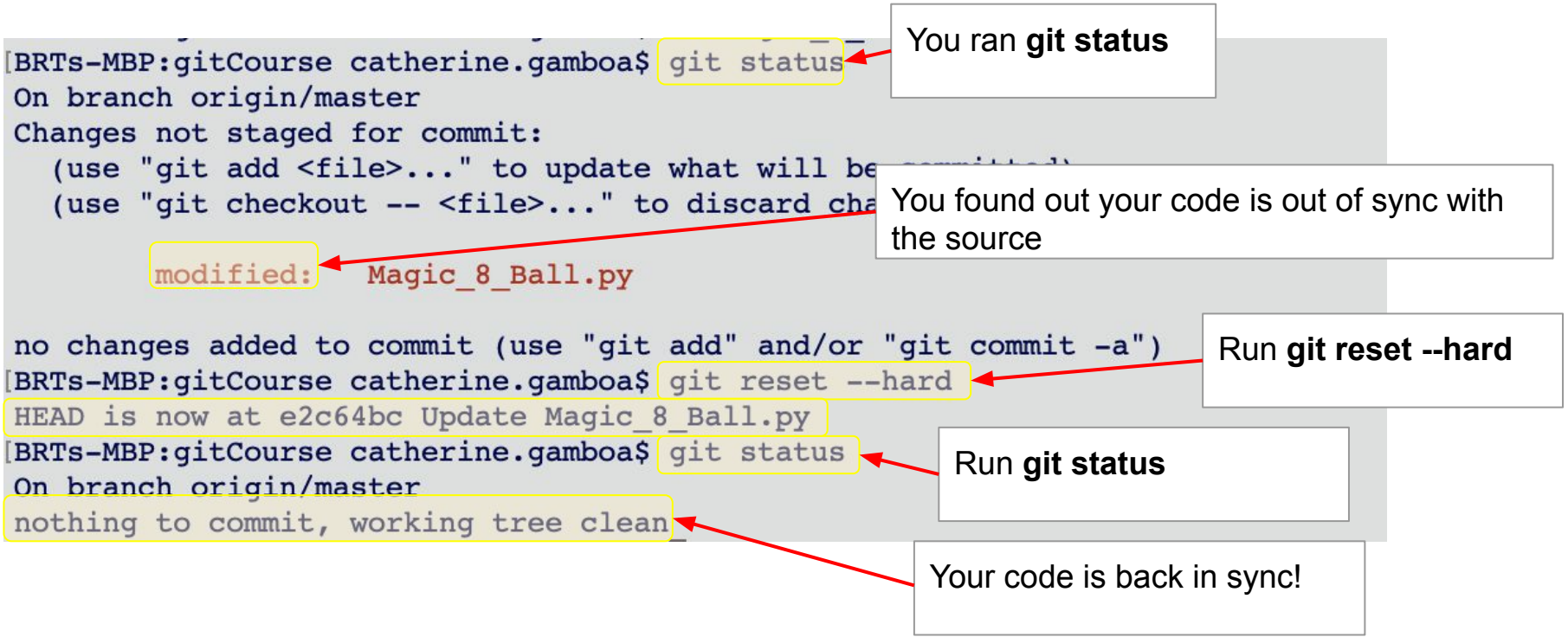
```
BRTs-MBP:gitCourse catherine.gamboa$ git status
On branch origin/master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        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



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

2. Reset the local branch

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.
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
```

4. Check the status

Lab: Get the newest version of the code

```
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)
```

5. Pull the new code into you branch

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