

Pushing to GitHub:

Short Version for reference.
Long Version for learning.

Pushing to GitHub:

Short Version(Go to listed pages for more info on each step)

`mkdir Directory_Name` – Step 1

`cd Directory_Name` – Step 2

`git init` – Step 3

Locate the files that you wish to push to GitHub and copy those files into your directory. – Step 4

`git add --all` - Step 5

`git commit -am 'commit message'` – Step 6

Create a GitHub repository in your browser. – Step 7

`git remote add origin https://github.com/User-Name/Repository-Name` - Step 8

`git push -u origin master` – Step 9

Step 1: Making A Directory

Step 1: mkdir Directory_Name

Command: `mkdir Directory_Name`

- What does mkdir mean?:
- mkdir stands for “Make Directory”
- That is why its use is creating directories.
- What does Directory_Name Mean?:
- The Directory_Name line can be replaced with whatever you wish to name your directory.
- For example, if you are making a directory for the snitch-sniffer exercise, Directory_Name would be replaced with snitch-sniffer

```
117567@MS1-2045-15 MINGW64 ~  
$ mkdir Example_Directory
```

Step 2: Connecting To The Directory

Step 2a: cd Directory_Name

Command: `cd Directory_Name`

What does cd mean?

- cd stands for “Change Directory”.
- It is used to connect to a directory.

What does Directory_Name Mean?:

- The Directory_Name line can be replaced with whatever your directory's name is.
- For example, if you have made a directory for the snitch-sniffer exercise, Directory_Name would be replaced with snitch-sniffer

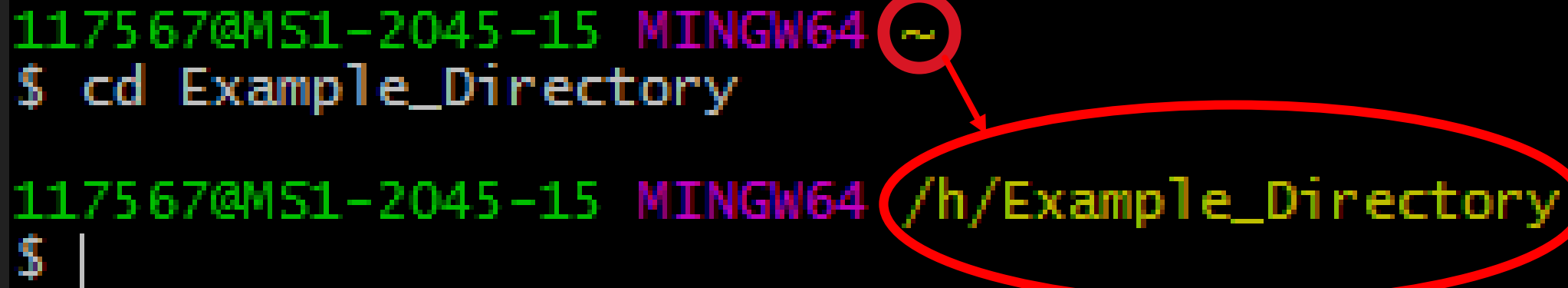
```
117567@MS1-2045-15 MINGW64 ~  
$ mkdir Example_Directory
```

```
117567@MS1-2045-15 MINGW64 ~  
$ cd Example_Directory
```

Step 2b: cd Directory_Name

Once you connect to the directory, you will visually see what directory you have connected to:

```
117567@MS1-2045-15 MINGW64 ~  
$ cd Example_Directory  
117567@MS1-2045-15 MINGW64 /h/Example_Directory  
$ |
```

A terminal window screenshot with a black background and multi-colored text. The first line shows the prompt '117567@MS1-2045-15 MINGW64' followed by a tilde '~' symbol. The second line shows the command '\$ cd Example_Directory'. The third line shows the prompt '117567@MS1-2045-15 MINGW64' followed by the path '/h/Example_Directory'. The fourth line shows '\$ |'. A red circle highlights the tilde symbol in the first line, and a red arrow points from it to a larger red oval that encircles the path '/h/Example_Directory' in the third line.

Step 3: Initializing The Git Repository

Step 3: git init

Command: `git init`

```
117567@MS1-2045-15 MINGW64 /h/Example_Directory
$ git init
Initialized empty Git repository in H:/Example_Directory/.git/
```

What does `git` mean?:

- `git` is the basic Git Bash command. When you want to utilize a command within Git Bash, you will start the command with “`git`” (`git` is going to appear a lot, so I’m only going to explain it once)

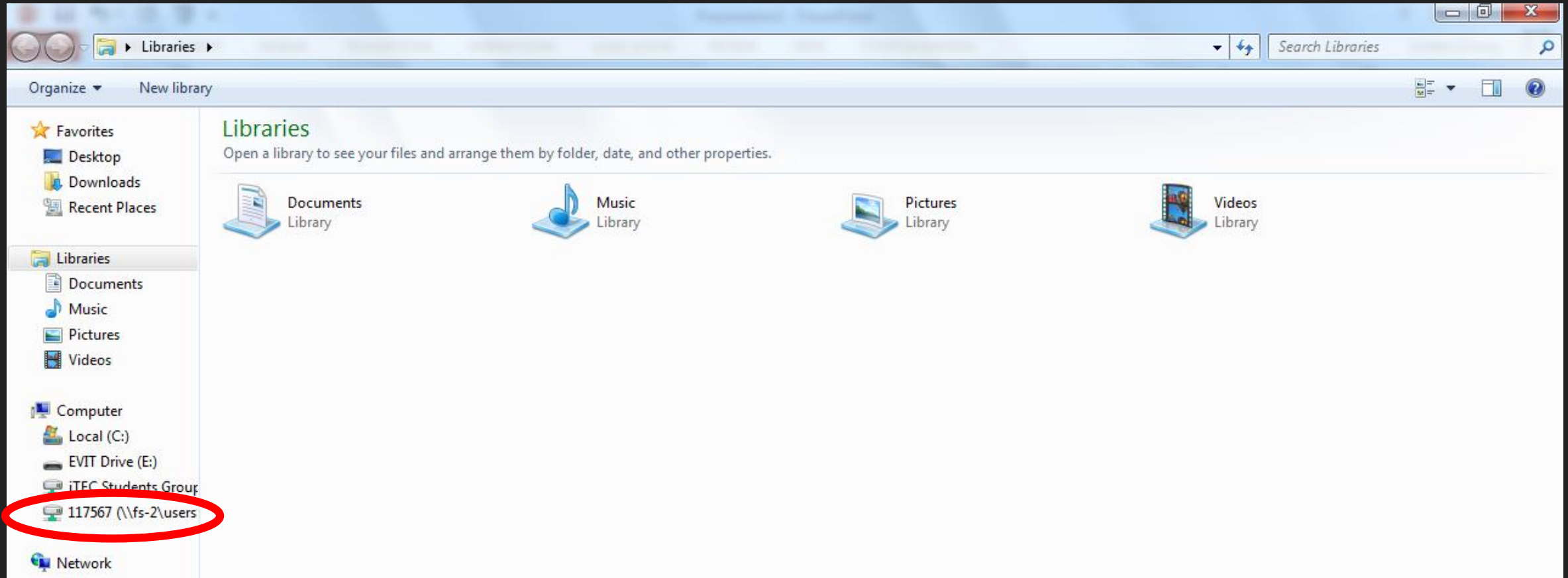
What does `init` mean?:

- `init` stands for “Initialize”.
- It is used to initialize a git repository that monitors changes within the directory.

Step 4: Working With The Directory

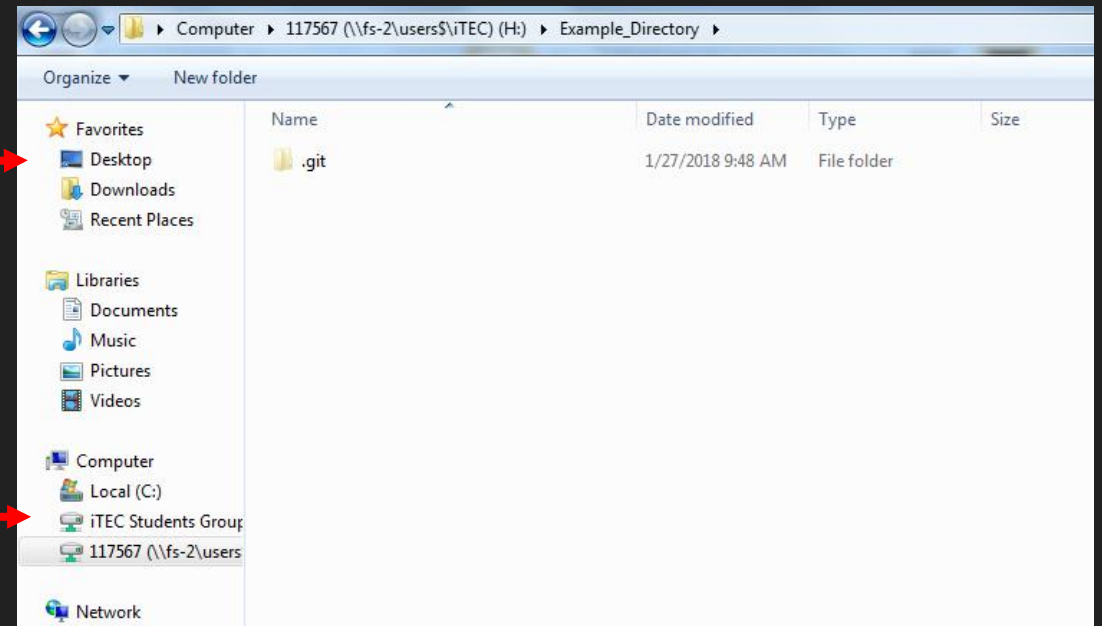
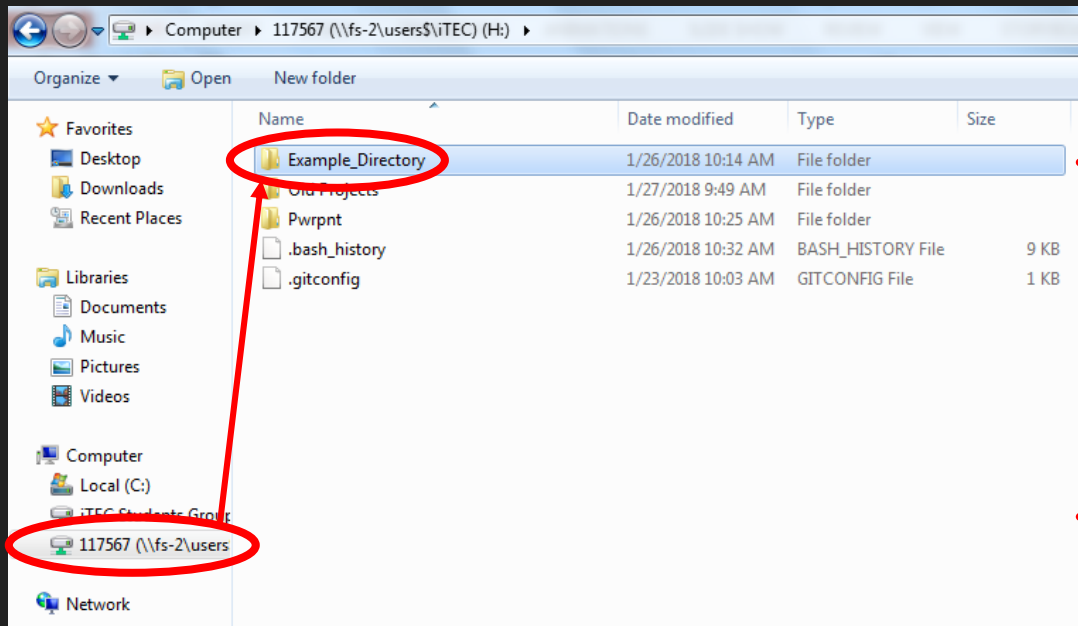
Step 4a: Locating Your Directory

Open your file explorer, and click on your EVIT ID:



Step 4b: Opening Your Directory

Locate and open your repository. (This will have the same name as Directory_Name)

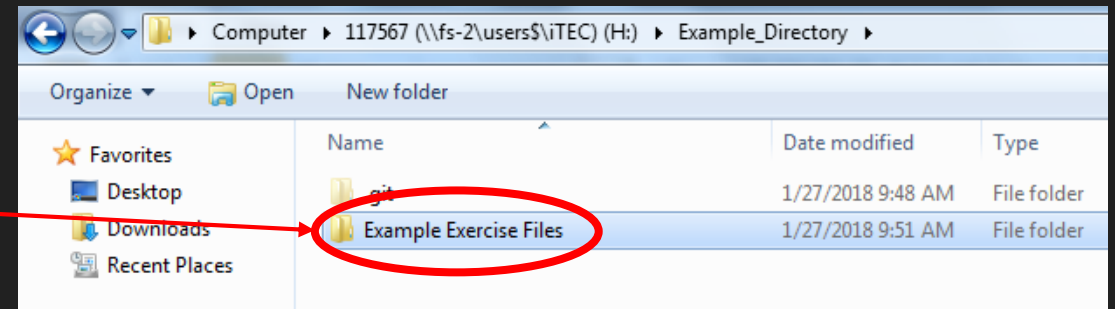


Step 4c: Copy Files Into Directory

1. Find your files that you wish to push to GitHub:

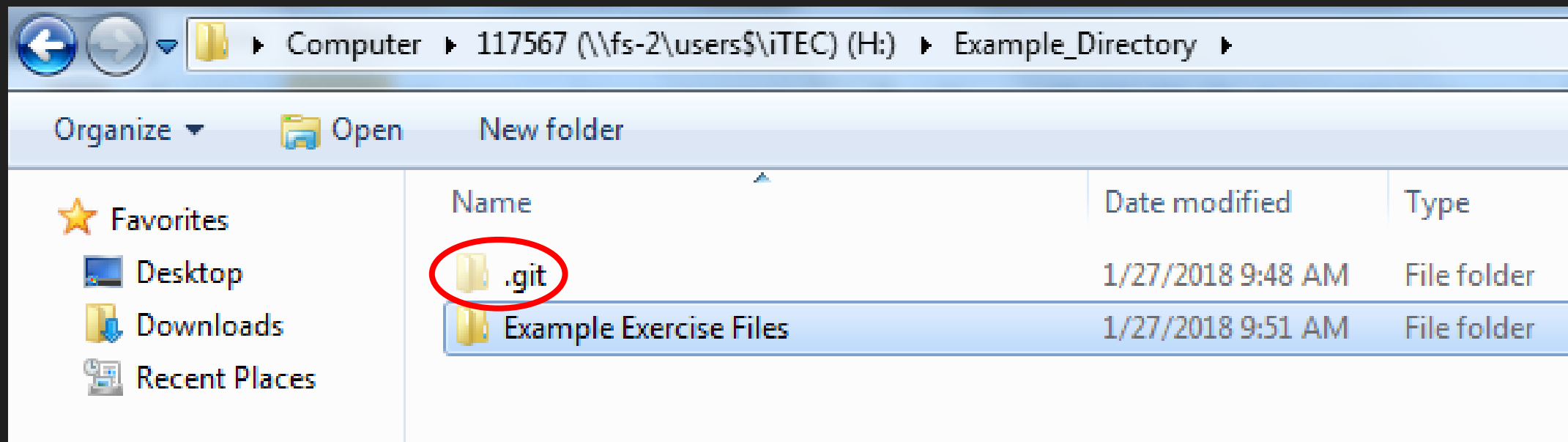


2. Copy them into the repository:



Step 4 Extra Notes:

This faded “.git” file is the file that was added when using the `git init` command



Step 5: Adding Files To The Repository

Step 5: git add --all

Command: `git add --all`

Reference Slide 6 for “git” info

What does add mean?:

- When you type “add” you are telling your .git file that you want to add the files from the Directory, to the repository.

What does --all mean?:

- “--all” specifies which files you want to add.
- Typing “--all” tells Git Bash that you want to add all the files that are currently in your Directory.

```
117567@MS1-2045-15 MINGW64 ~
$ cd Example_Directory

117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ git init
Reinitialized existing Git repository in H:/Example_Directory/.git/

117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ git add --all

117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ |
```


Step 5 Extra Notes:

In the previous slide, I mentioned both **Directories** and **Repositories**.
To prevent further confusion, I am going to clarify which is which.

The **Directory** is the file that you created and connected to with the “**mkdir**” and “**cd**” commands
The **Directory** is where files are stored on your computer.

The **Repository** is the file that you created with the “**git init**” command.
The **Repository** is what makes it possible for files on your computer to be stored in GitHub.

Think of it this way:

A **Directory** is where files are locally stored (on your computer).
While a **Repository** is where they are stored on a cloud (GitHub).

Step 6: Committing Files

Step 6a: git commit -am 'commit message'

Command: `git commit -am 'commit message'`

Reference Slide 6 for “git” info

```
117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ git commit -am 'Added python exercise files'
[master a68274d] Added python exercise files
2 files changed, 1 insertion(+)
delete mode 100644 Example Exercise Files/Another Folder/lkjsdf1jafs.bmp
create mode 100644 Example Exercise Files/Example Python File.py

117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ |
```

What does commit mean?:

- When editing a repository, GitHub wants to keep track of what changes are made. You can tell GitHub your changes by using “commit”

Step 6b: git commit -am 'commit message'

Command: `git commit -am 'commit message'`

```
117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ git commit -am 'Added python exercise files'
[master a68274d] Added python exercise files
 2 files changed, 1 insertion(+)
 delete mode 100644 Example Exercise Files/Another Folder/lkjsdf1jafs.bmp
 create mode 100644 Example Exercise Files/Example Python File.py

117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ |
```

What does -am mean?:

- -am is actually two different commands. '-a' is the first command, with 'm' immediately after.
- The '-a' stands for "all" and essentially serves the same purpose as the '--all' command in `git add --all`
- The 'm' stands for message. This is what allows you to type a commit message when committing a change.

Step 6c: git commit -am 'commit message'

Command: `git commit -am 'commit message'`

What should I put in 'commit message'?:

- Inside the single quotes, you will want to explain what you added or changed. In this case, I added the example exercise files, so in the 'commit message' section, I would want to say that:

```
117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ git commit -am 'Added python exercise files'
[master a68274d] Added python exercise files
2 files changed, 1 insertion(+)
delete mode 100644 Example Exercise Files/Another Folder/lkjsdf1jafs.bmp
create mode 100644 Example Exercise Files/Example Python File.py

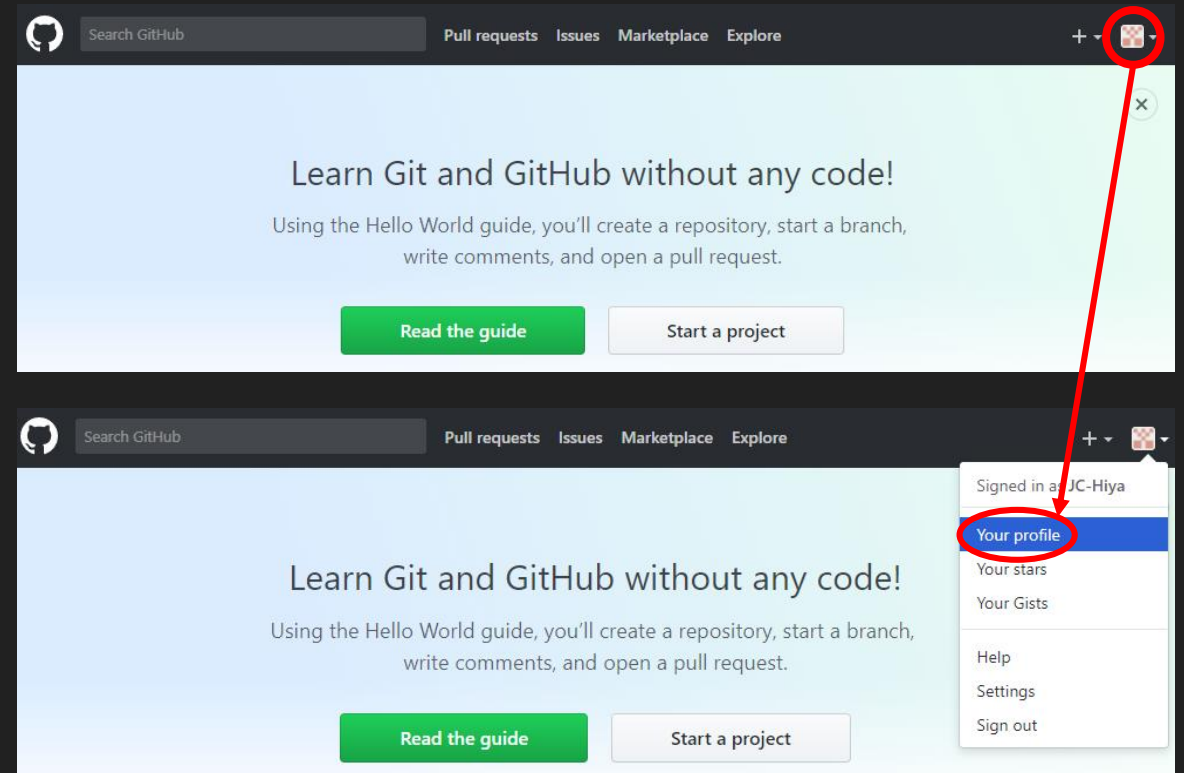
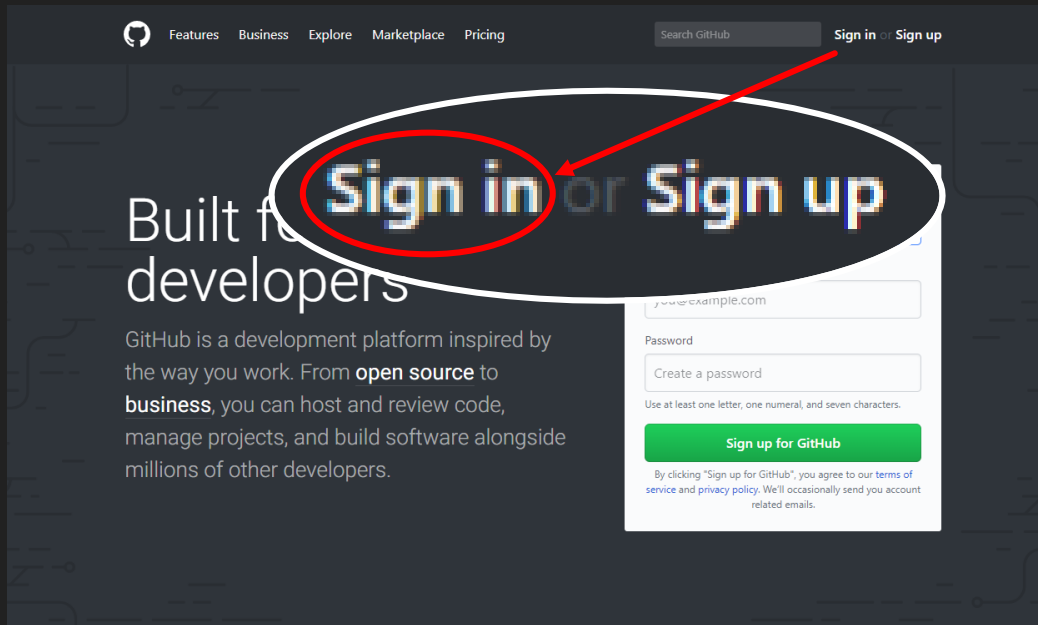
117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ |
```

Step 7: Making A GitHub.com Repository

Step 7a: Getting to “Your Profile”

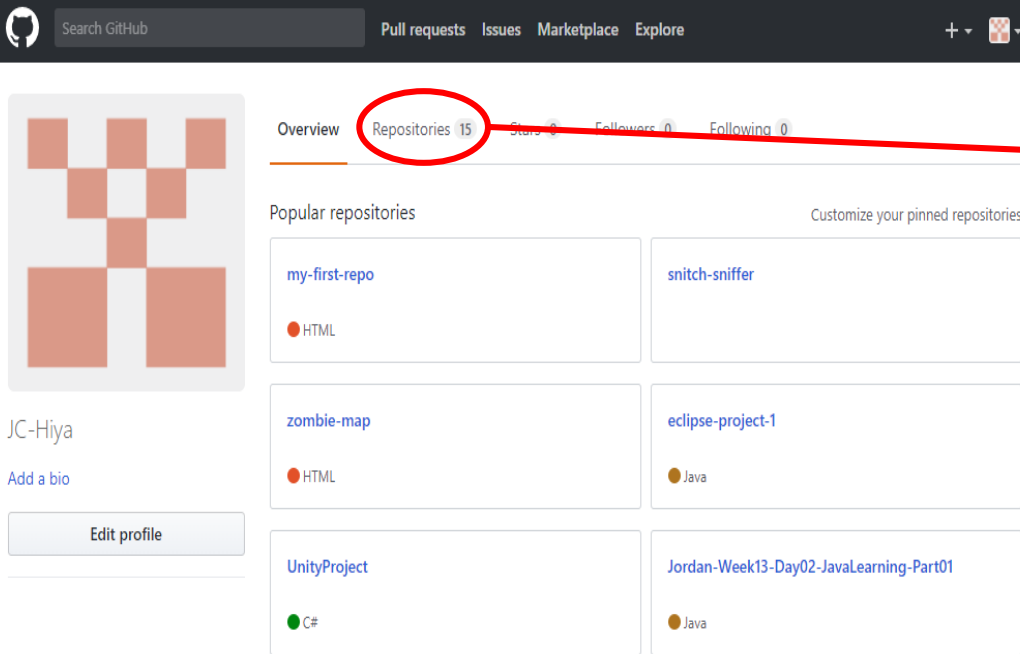
1. Login into your GitHub Account:

2. Go to “Your Profile”:



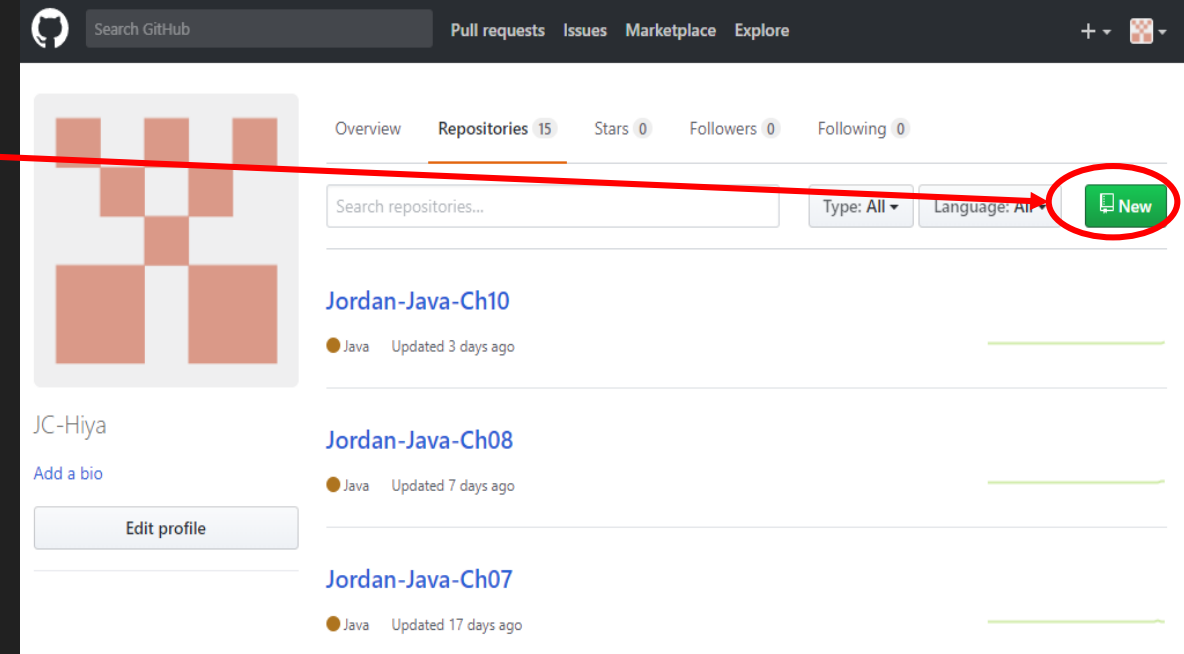
Step 7b: Making A New Repository

1. Open your “Repositories” tab:



A screenshot of a GitHub profile page. The user's name is JC-Hiya. The 'Repositories' tab is selected and circled in red. Below the profile information, there are several repository cards, including 'my-first-repo' (HTML), 'snitch-sniffer', 'zombie-map' (HTML), 'eclipse-project-1' (Java), 'UnityProject' (C#), and 'Jordan-Week13-Day02-JavaLearning-Part01' (Java).

2. Make a new Repository:



A screenshot of the same GitHub profile page, but with the 'New' button highlighted in red. The 'New' button is located in the top right corner of the repository list area, next to the search and filter options. The repository list shows 'Jordan-Java-Ch10' (Java, updated 3 days ago), 'Jordan-Java-Ch08' (Java, updated 7 days ago), and 'Jordan-Java-Ch07' (Java, updated 17 days ago).

Step 7c: Creating the Repository

Name the Repository appropriately, and click “Create repository”:

Search GitHub Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: JC-Hiya

Repository name: Jordan-Example-Repository ✓

Great repository names are short and memorable. Need inspiration? How about [reimagined-spork](#).

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

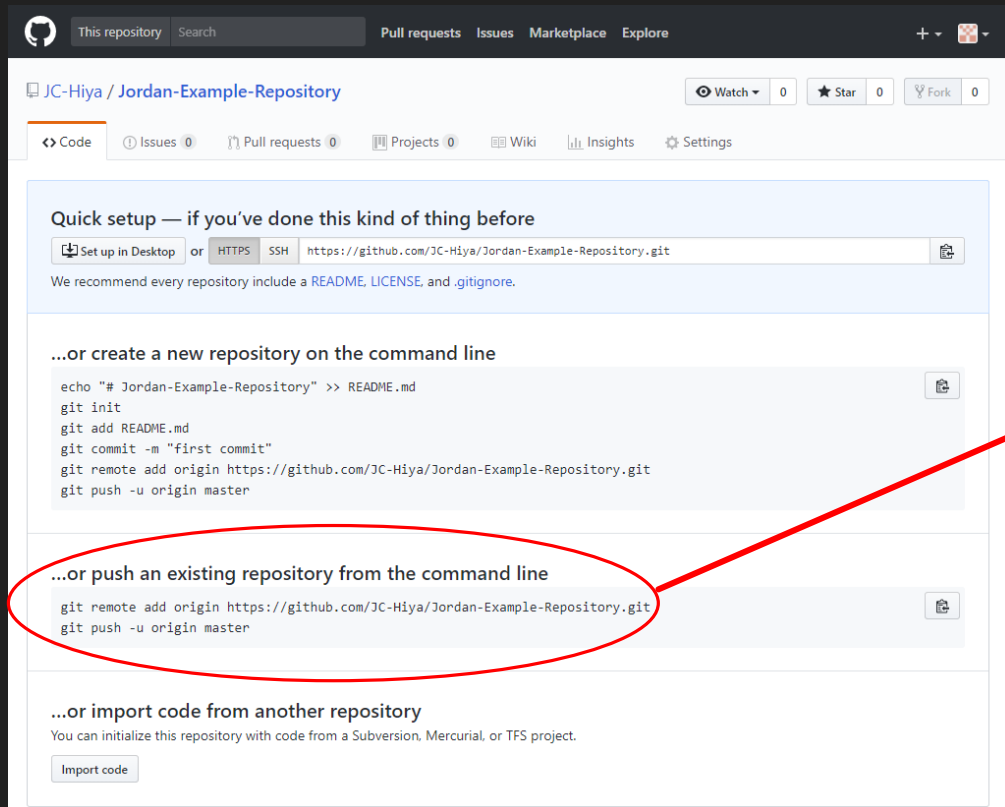
Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None ⓘ

Create repository

Step 7d: Finding Your Next Command

Once you create the Repository, find this line of code.
Copy it for the next step:



The screenshot shows the GitHub interface for a repository named 'Jordan-Example-Repository'. The 'Code' tab is selected, displaying a 'Quick setup' section. Under the heading '...or push an existing repository from the command line', the following terminal commands are listed:

```
git remote add origin https://github.com/JC-Hiya/Jordan-Example-Repository.git
git push -u origin master
```

This section is circled in red, and a red arrow points from it to a callout box on the right.

...or push an existing repository from the command line

```
git remote add origin https://github.com/JC-Hiya/Jordan-Example-Repository.git
git push -u origin master
```

Step 7 Extra Notes:

You might be wondering: “Why did I have to create another repository?
Didn't I already make one with the `git init` command?”

The answer is: Yes. You did already make a repository with the `git init` command. But, there is a difference between the two repositories.

The repository you made with the `git init` command is a repository on your computer, while the repository you made on GitHub is a repository in the cloud.

The `.git` repository on your computer functions as a bridge. It takes files on your computer, puts them into a local repository, and pushes them to the cloud on GitHub.com

Step 8: Adding A Remote

Step 8: git remote add origin https://GitHub.com/User-Name/Repository-Name

Command: `git remote add origin https://GitHub.com/User-Name/Repository-Name`

This should be the command that you copied earlier.

Because you have this command copied already, I am not going to explain it in detail.

Long story short, this command sets the .git Repository as a remote, and makes the remote's origin equal to the GitHub.com repository.

```
117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ git remote add origin https://github.com/JC-Hiya/Jordan-Example-Repository.git

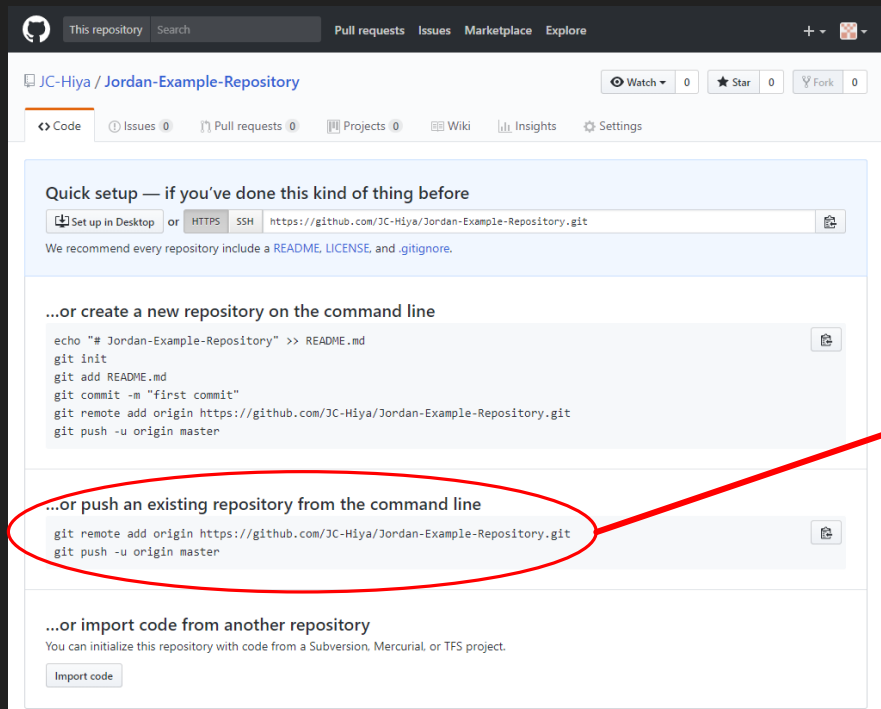
117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$
```

Step 9: Pushing To GitHub.com

Step 9a: git push -u origin master

Command: `git push -u origin master`

This is also a command that can be copied and pasted.
It's found here:



The screenshot shows the GitHub repository page for 'JC-Hiya / Jordan-Example-Repository'. The 'Code' tab is selected. Under the 'Quick setup' section, there are three options: 'Set up in Desktop', 'HTTPS', and 'SSH'. Below this, there are three sections of instructions:

- ...or create a new repository on the command line**:

```
echo "# Jordan-Example-Repository" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/JC-Hiya/Jordan-Example-Repository.git
git push -u origin master
```
- ...or push an existing repository from the command line**:

```
git remote add origin https://github.com/JC-Hiya/Jordan-Example-Repository.git
git push -u origin master
```
- ...or import code from another repository**:

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

A red circle highlights the second section, and a red arrow points from it to a callout box on the right.

...or push an existing repository from the command line

```
git remote add origin https://github.com/JC-Hiya/Jordan-Example-Repository.git
git push -u origin master
```

Step 9b: git push -u origin master

Command: `git push -u origin master`

Again, because you already have this command at your disposal, I am not going to go in depth on what each part does.

The reason I explained the other commands is so you have a better understanding of why you type what you type, but because the last two commands can be copied and pasted, you don't need a big understanding on what this means.

Long story short, this command pushes the .git repository to the specified origin from the previous command.

Step 9c: git push -u origin master

Command: `git push -u origin master`

The command might take a bit to finish, so be patient. You will know that it's pushed once you see something similar to this:

```
117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$ git push -u origin master
Counting objects: 9, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (9/9), 651 bytes | 0 bytes/s, done.
Total 9 (delta 0), reused 0 (delta 0)
To https://github.com/JC-Hiya/Jordan-Example-Repository.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.

117567@MS1-2045-15 MINGW64 /h/Example_Directory (master)
$
```

And There You Go!

Once you refresh the page, you should have all your files in your GitHub.com repository.

For more info on each command, and Git itself, head to [Git's documentation page](#)