Git & GitHub

Agenda

1. What is Version Control?
2. History of Git.
3. Why VC is Needed.
4. What is GitHub?
5. How to use Git
6. How to connect to GitHub
7. Collaboration.
8. Activity.

1

What is Version Control?

What is Version Control?

- VCS – Version Control System
- Tracks file modifications
- Allows for restore points
- Allows for collaboration features
- Supports branching
- Holds detailed history
- Example: Git, SVN, others...

2

History of Git

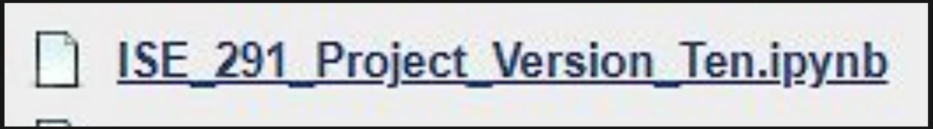
History of Git.

- Git is a distributed VCS
- It was created in 2005 by Linus Torvalds because of a fight with another VCS, BitKeeper
- The distributed nature means every user has full project history
- Was boosted by GitHub’s launch.

3

Why is VC needed?

Why is VC needed?

- It is needed to avoid situations like this: 
ISE_291_Project_Version_Ten.ipynb
- Solve and avoid code conflicts
- Streamline collaboration
- Store backups
- History to track changes

4

What is GitHub?

What is GitHub?

- It is a centralized software hub that hosts Git repositories
- Enables open source collab
- Track issues
- Allows for CI/CD pipelines

5

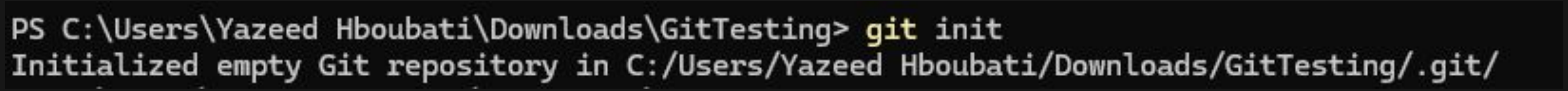
How to use Git

How to use Git

- First, you need to install the Git software
- Navigate to https://git-scm.com/downloads
- Or:

How to use Git

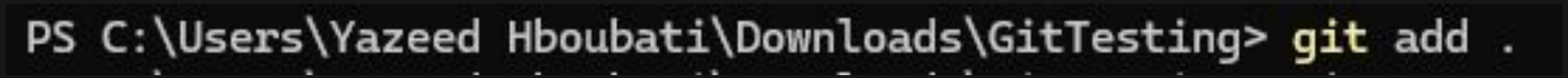
- Open a folder on your desktop
- Right click the empty space
- Click open in terminal
- Run git init



```
PS C:\Users\Yazeed Hboubati\Downloads\GitTesting> git init
Initialized empty Git repository in C:/Users/Yazeed Hboubati/Downloads/GitTesting/.git/
```

How to use Git

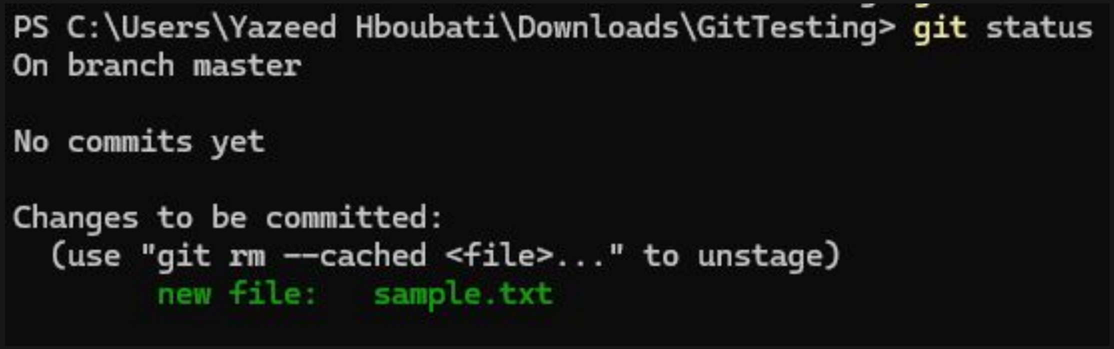
- Now that we have initialized a local repository, it is time to add files to it for them to be tracked
- Create a simple sample.txt file
- Now type git add . into the command prompt



```
PS C:\Users\Yazeed Hboubati\Downloads\GitTesting> git add .
```

How to use Git

- To check the status of out git repository, we use git status

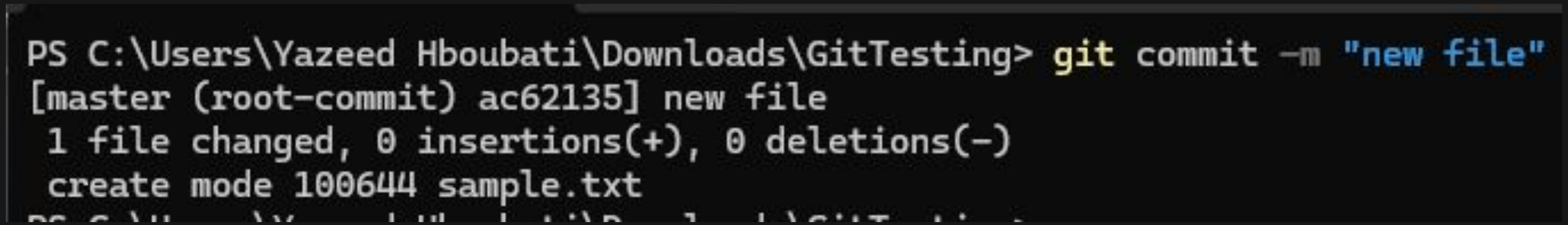


```
PS C:\Users\Yazeed Hboubati\Downloads\GitTesting> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   sample.txt
```

- Now this tells us there is 1 new file added and waiting to be committed

How to use Git

- Committing means to save a snapshot in time of what a file looks like.
- To save our file in the git repo history, we run git commit -m “new file”



```
PS C:\Users\Yazeed Hboubati\Downloads\GitTesting> git commit -m "new file"
[master (root-commit) ac62135] new file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 sample.txt
```

How to use Git

- The previous commands were git basics, we are just getting into the fun stuff!

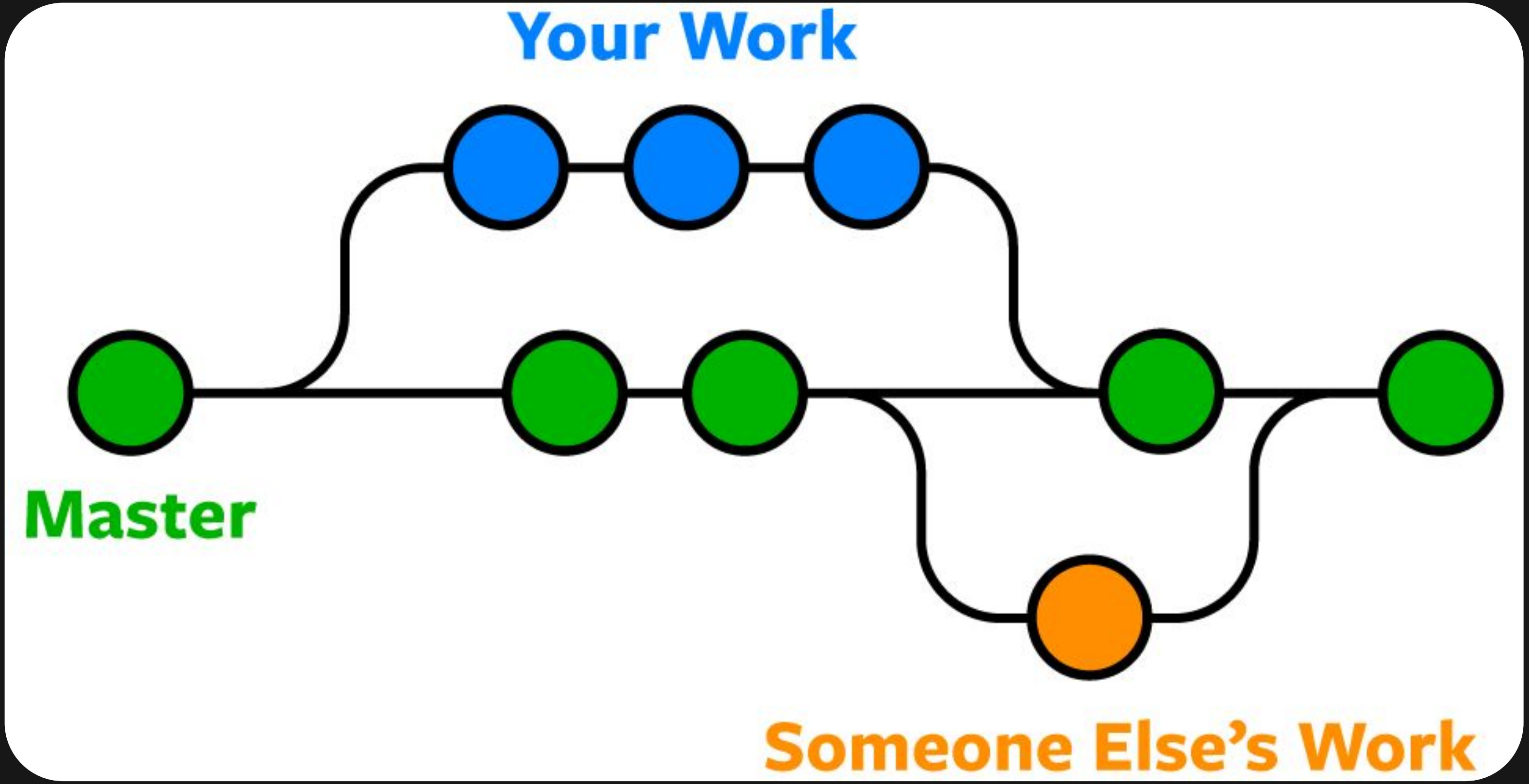
- Let us shift the difficulty up a bit and talk about collaboration, shall we?

How to use Git

- Now collaboration is not as simple as 2 people committing
- This would result in conflicts and headaches
- This is where we introduce Git branches

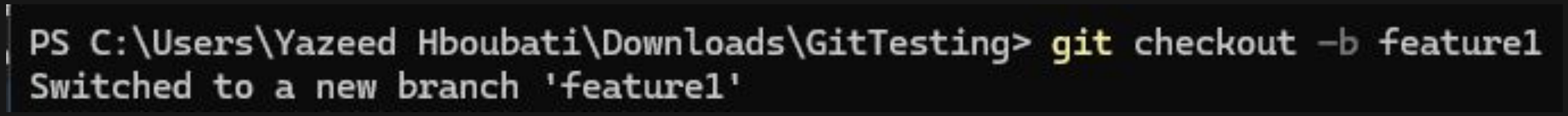
How to use Git

- Branching Diagram:



How to use Git

- Now to create a branch so we can develop peacefully from other developers, we use the following command:
  git checkout -b feature1

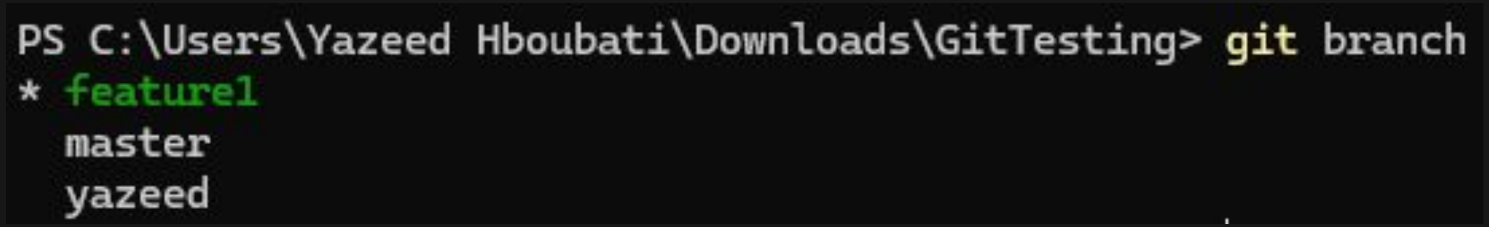


```
PS C:\Users\Yazeed Hboubati\Downloads\GitTesting> git checkout -b feature1
Switched to a new branch 'feature1'
```

How to use Git

- Now to check our branches and check our current working branch, we use the following command:

git branch



```
PS C:\Users\Yazeed Hboubati\Downloads\GitTesting> git branch
* feature1
  master
  yazeed
```

How to use Git

- Let’s try branching, go back and do all the starter steps from creating a file to committing
- After doing so, run:

```
git checkout master
```

- You will notice that the new file you created is not in the master branch!

6

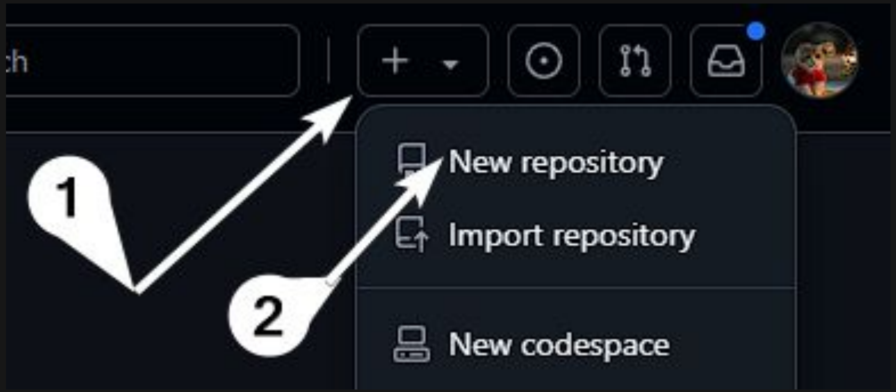
How to connect to GitHub

How to connect to GitHub

- First off, an account must be created on GitHub.
- To create an account, scan the QR code and create the account.

How to connect to GitHub

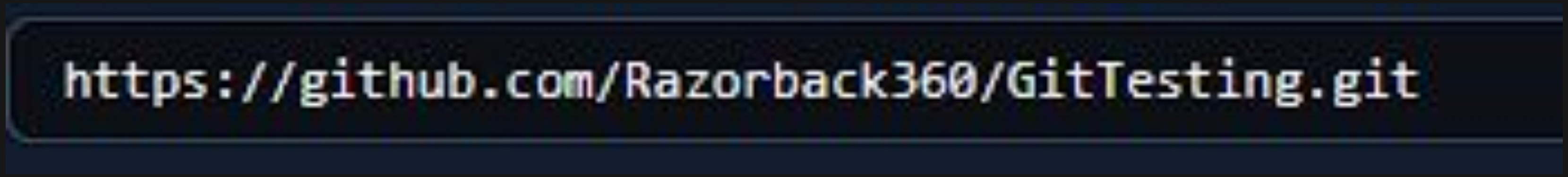
- Now, create a new repository on GitHub by following these steps:



- Write a repository title (use the same as your local repository).
- Set privacy to public.

How to connect to GitHub

- After creating a new repository, you will need to copy the repository Git URL.
- It will be the same URL + .git in the end.
- In my case it is:



```
https://github.com/Razorback360/GitTesting.git
```

How to connect to GitHub

- After copying the Git URL, we will link our remote repository (GitHub repository) to our local repository using this command:

```
git remote add origin GIT_URL
```

- You will be asked for your login information. Enter it and continue.

How to connect to GitHub

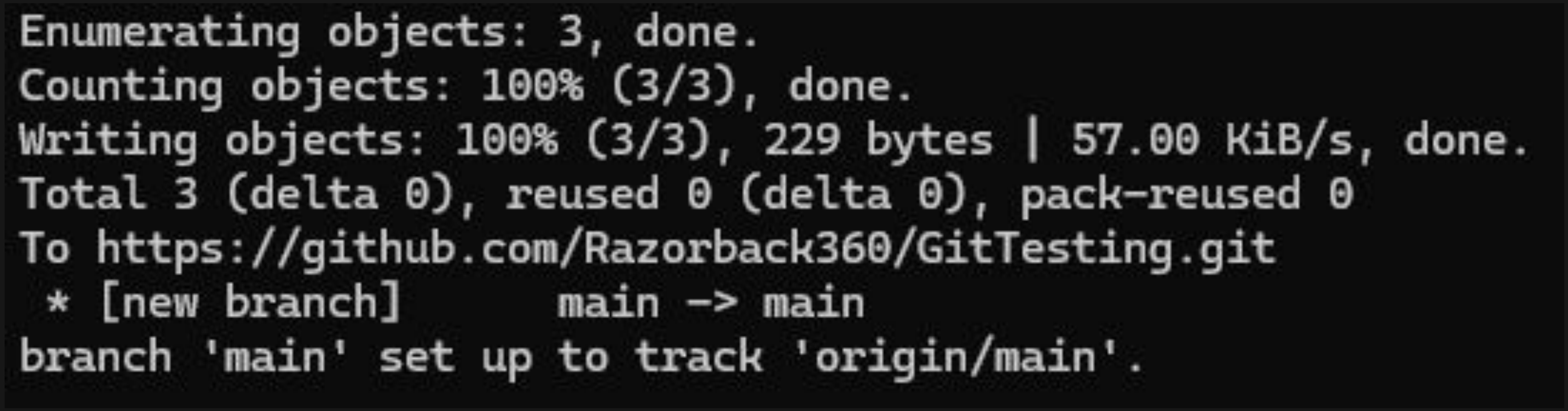
- After connecting, we set the main branch to be the main branch on GitHub using:

```
git branch -M main
```

How to connect to GitHub

- The final step is publishing our repository. We do that using:

git push -u origin main



```
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 229 bytes | 57.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Razorback360/GitTesting.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

7

Collaboration

Collaboration

- Now that the repository is linked, we will talk about collaboration.
- Main Git commands:
  - clone
  - push
  - pull
  - fetch

Collaboration

- The clone command downloads a repository given a URL.
- Usually used at the start of a project to get the files and start work.
- To clone:

```
git clone <repository url>
```

Collaboration

- The push command takes your local changes and sends them to the remote repository.
- Prior to pushing, you must have committed changes.
- To push:

```
git push
```

Collaboration

- Full workflow example:
  - Change a file's contents
  - Run `git add .`
  - Run `git commit -m "Changed XX to do YY"`
  - Run `git push`

Collaboration

- The pull command get changes from remote and merges them into your local version.
- You do not get to view what the changes are before the merge.
- To pull:

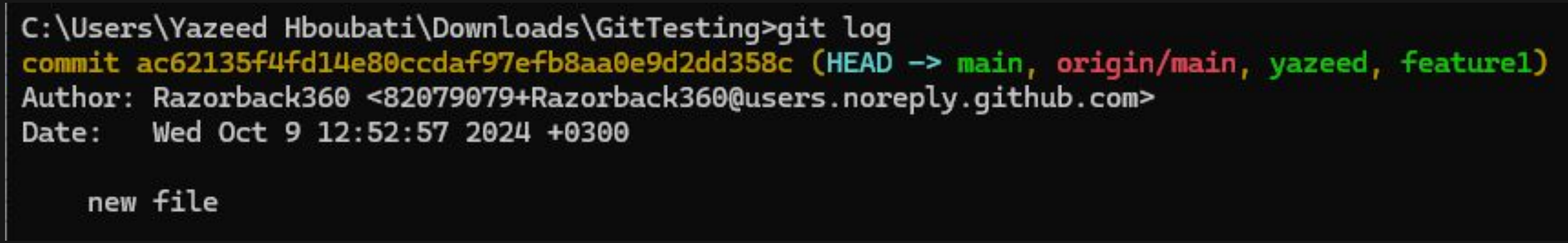
```
git pull
```

Collaboration

- The fetch command downloads the changes other people made **without** merging with your changes.
- This is useful to check changes without doing any destructive actions.
- To fetch: `git fetch`

Collaboration

- Full workflow example:
  - Run git fetch
  - Run git log to view changes



```
C:\Users\Yazeed Hboubati\Downloads\GitTesting>git log
commit ac62135f4fd14e80ccdaf97efb8aa0e9d2dd358c (HEAD -> main, origin/main, yazeed, feature1)
Author: Razorback360 <82079079+Razorback360@users.noreply.github.com>
Date:   Wed Oct 9 12:52:57 2024 +0300

    new file
```

  - Optional: Run git merge to get changes locally

Collaboration

- Collaboration in a private or public repository as a member is simple.
- However, what if you need to collaborate on a public repository without a membership?

Collaboration

- To collaborate on a public repository, we will need to fork it.
- A fork is essentially a “branch” in a way, but in a form of an entire repository.
- The repository is under your account.

Collaboration

- After forking the repository, you can clone your fork locally and work on it.
- After finishing work, push your changes to the fork.
- After that, you create a Pull Request.

Collaboration

- Try forking the following repository: https://github.com/Razorback360/CCGitWorkshop

Or:

Collaboration

- Add any file to the repository
- Now try to create a Pull Request to the original repository with your name and ID.

8

Activity

Activity

- This is a solo activity. The activity files are in the repository cloned earlier along with instructions on what to do.