

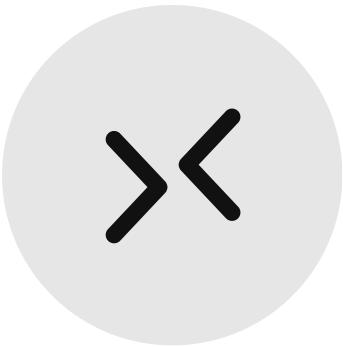
# Module 3



Configuring GitHub



Remote Repository



Setting Remote



Push



Git Cloning



Pull

# GitHub

# GitHub

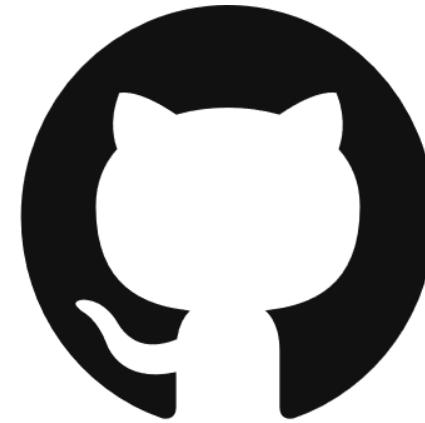
- In 2005, Linus Torvalds, the creator of Linux, built Git, a powerful version control system. However, Git itself was just a tool, not a platform.
- In 2008, Tom Preston-Werner, Chris Wanstrath, and PJ Hyett launched GitHub as a user-friendly platform for hosting Git repositories.
- Think of GitHub like Google Drive, but for code.
- It helps us to store, manage, and share our code files or projects online.



GitLab



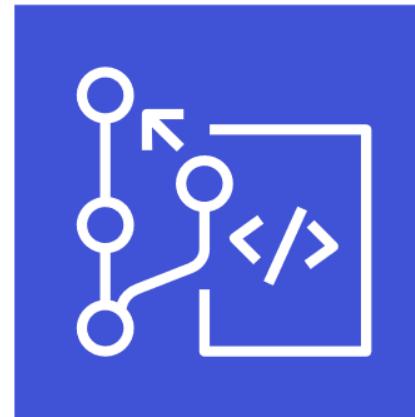
Bitbucket



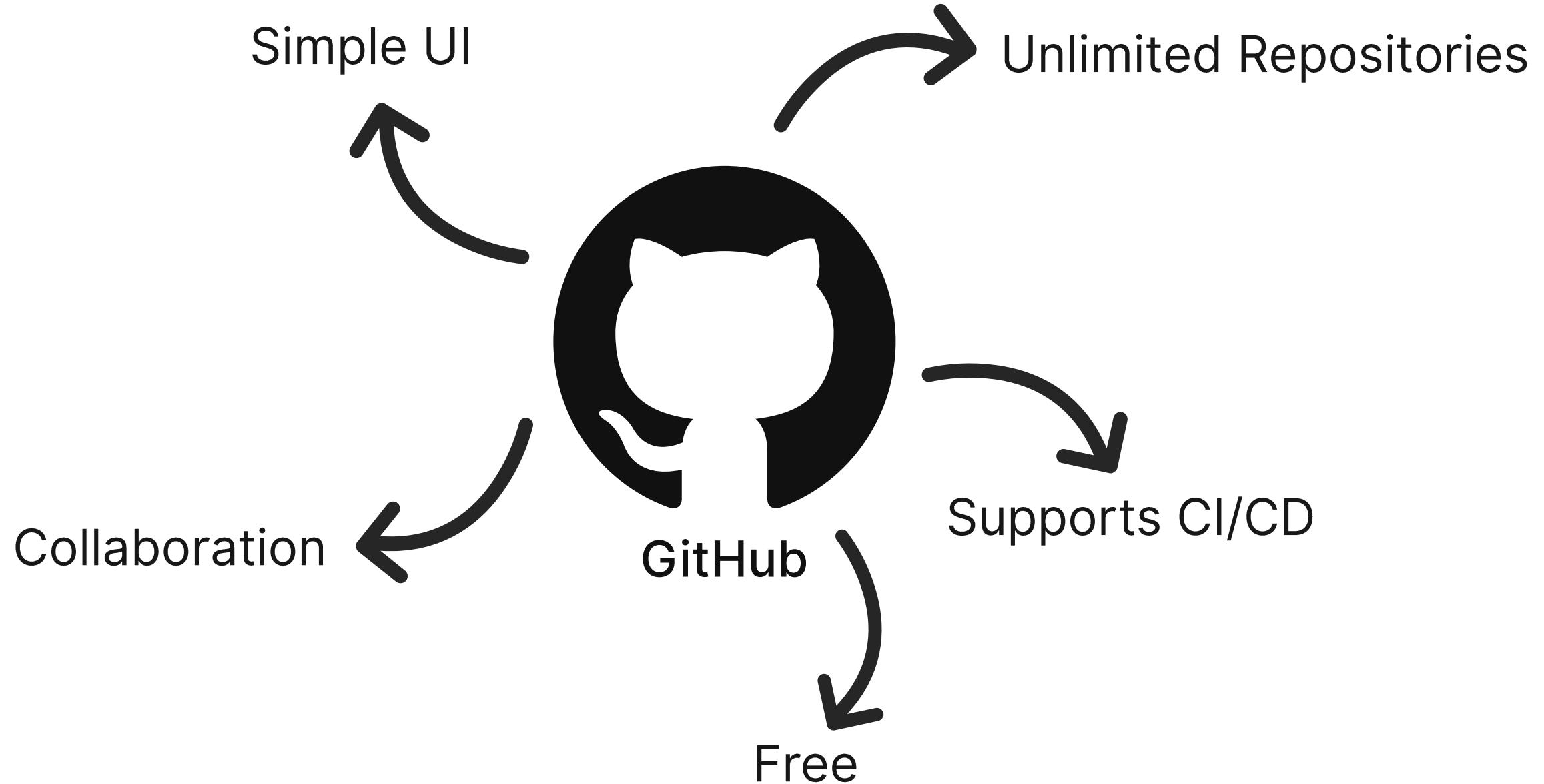
GitHub



SourceForge



AWS CodeCommit



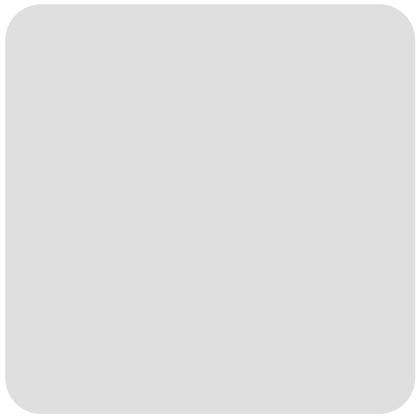
# **Configuring GitHub**

# **Creating Remote Repository**

# Creating Remote Repository

- Creating a GitHub repository is like making a folder for our project online.
- This folder keeps all our files, tracks changes, and lets us work with others easily.

# **Adding Remote**



Remote Repository

GitHub



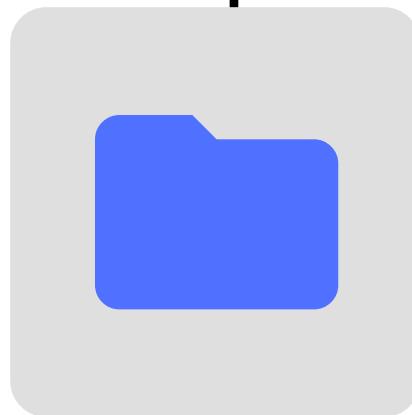
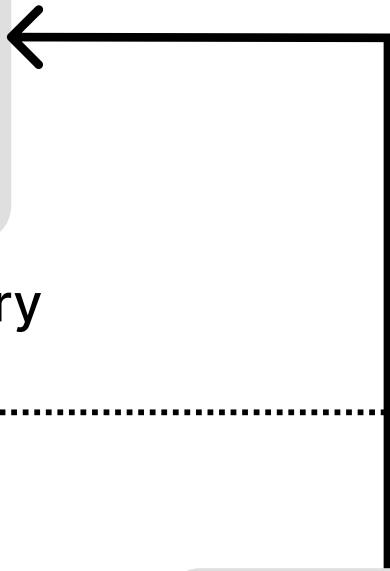
Local Repository

Computer

---



Remote Repository



Local Repository

GitHub

Computer

# Adding Remote

- Imagine we have a project on our computer (local), but we want to store it online (GitHub).
- By adding a remote, we connect our local project to an online repository so we can back it up, access from anywhere, share & collaborate with others.

```
git remote add remote-name <url>
```

# **Viewing Remote**

# Viewing Remote

- We can view our remote using the following:

```
git remote
```

```
git remote -v
```

# **Configuring Git Username**

# Configuring Git Username

- When we use Git with GitHub, you often need to authenticate (log in) to push or pull code.
- Setting up our Git username globally helps Git recognize which user is making changes across all repositories.

```
git config --global credential.username "username"
```

# **Push**



Remote Repository

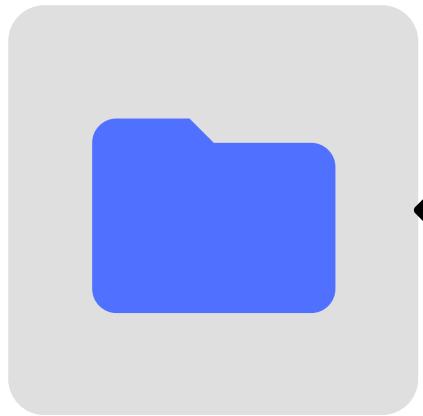
GitHub



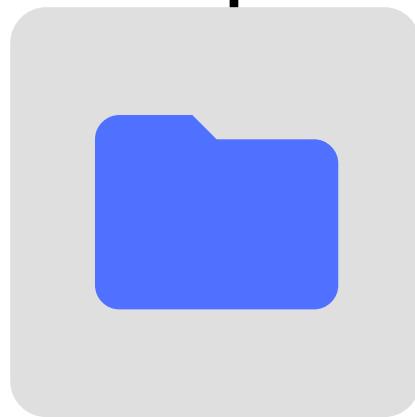
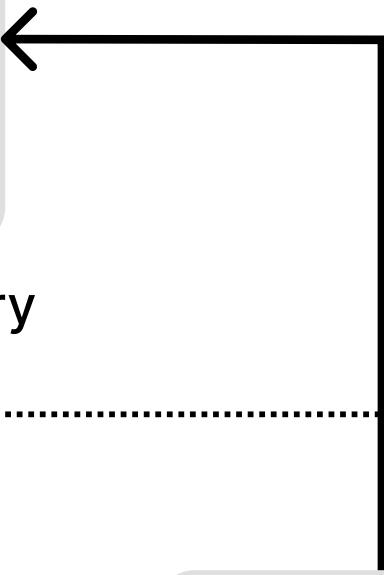
Local Repository

Computer

---



Remote Repository



Local Repository

GitHub

Computer

# **Push**

- Push allows us to sync/upload our files from local to our remote repository.

**git push origin <branch-name>**

**Warmup 15** ↗

# Warmup 15 ↧

- Create a new directory called **wup15** and navigate into it.
- Initialize a Git repository within the directory.
- Create three files: **src/main.py**, **app.py** and **temp.txt**.
- Add and commit them with the message “***initial commit***”.

Contd..

# Warmup 15 ↪

- Write `print("line 1")` in both `src/main.py` and `app.py`, then add and commit these files to Git with the message “**add line 1**”.
- Write `print("line 2")` in both files, then add and commit these changes with the message “**add line 2**”.
- Again, write `print("line 3")` in both files, then add and commit these changes with the message “**add line 3**”.
- View the commit history.

Contd..

# Warmup 15 ↧

- Ignore the **temp.txt** file using Git ignore, then add Git ignore file and commit to Git with the message “**add gitignore**”.
- Check the status and view the commit history.
- Create a remote repository in GitHub named as “**remote-wap-15**”.
- Add remote to the local repository to connect with remote repository created.
- View the remote.

Contd..

# Warmup 15 ↧

- Now, push the local repository to the remote repository created.
- Go to your remote repository on GitHub and refresh the page. What do you observe?

**Warmup 16** ↗

# Warmup 16 ↧

- Navigate to “**wup15**” directory created previously.
- Write ***print("new line")*** in both ***src/main.py*** and ***app.py***, then add and commit these files to Git with the message **“add new line”**.
- Go to your remote repository on GitHub and refresh the page. What do you observe?

Contd..

# Warmup 16 ↪

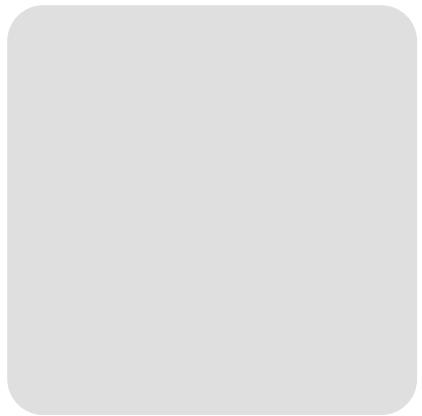
- Now, push the updated local repository to the remote repository.
- View your remote repository.

# **Git Cloning**



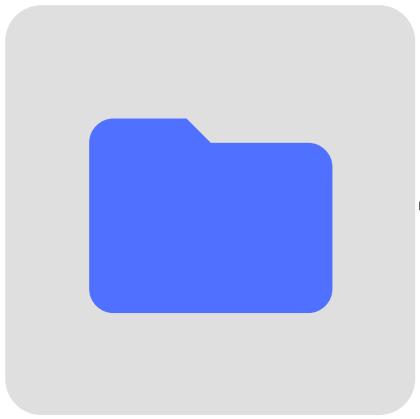
Remote Repository

GitHub



Local Repository

Computer



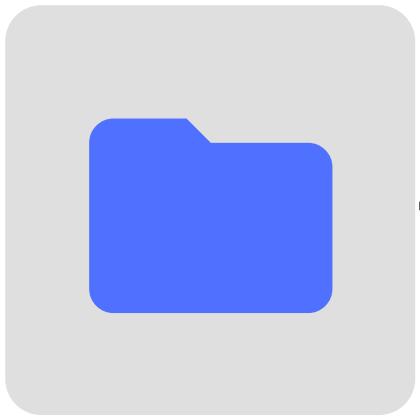
Remote Repository

GitHub



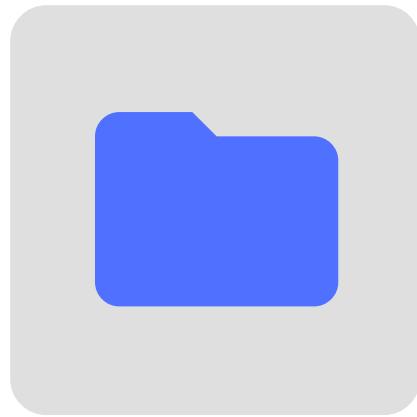
Local Repository

Computer



Remote Repository

GitHub



Local Repository

Computer



# Git Cloning

- Git cloning is like making a copy of a project from the internet (GitHub) to our own computer.
- It is helpful when we want the copy of our project back to our local repository in case of failure.

**git clone <url>**

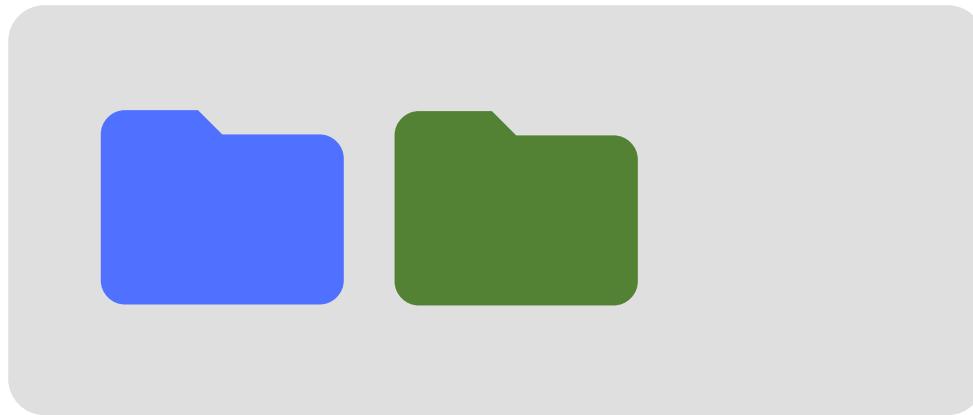
**git clone <url> repository-name**

**Warmup 17 ♪**

# Warmup 17 ↗

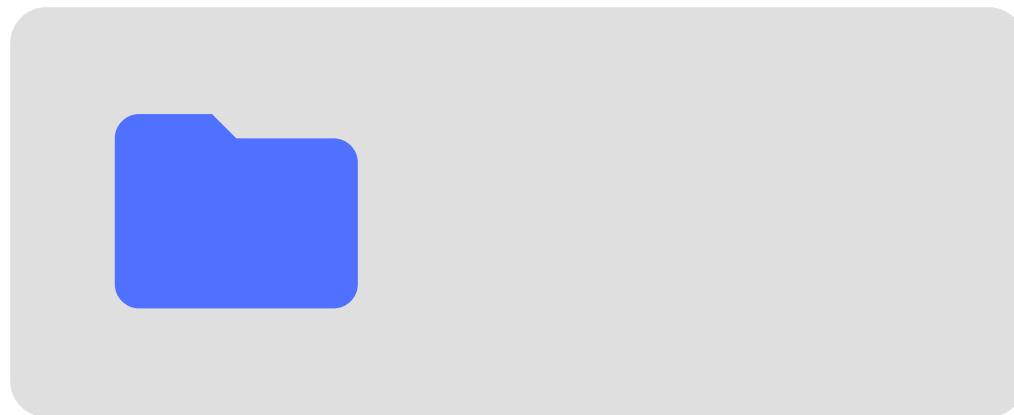
- Go to your previous remote repository “**remote-wap-15**” on GitHub.
- Clone the repository back into your computer.

**Pull**



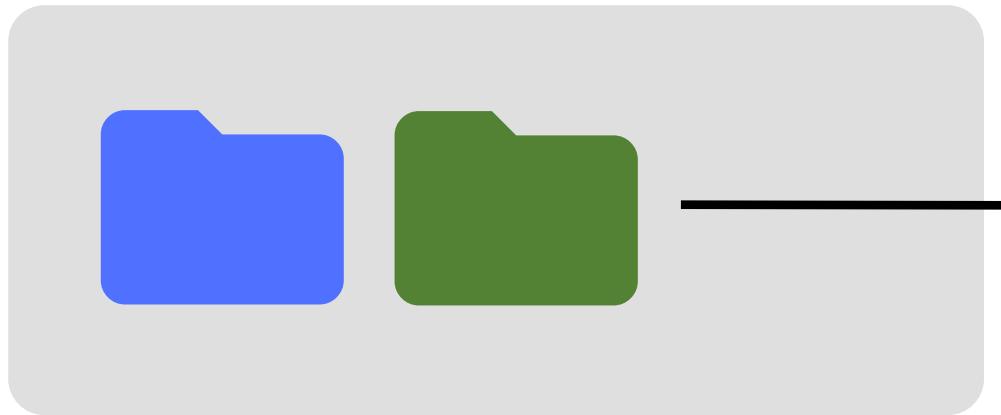
Remote Repository

GitHub



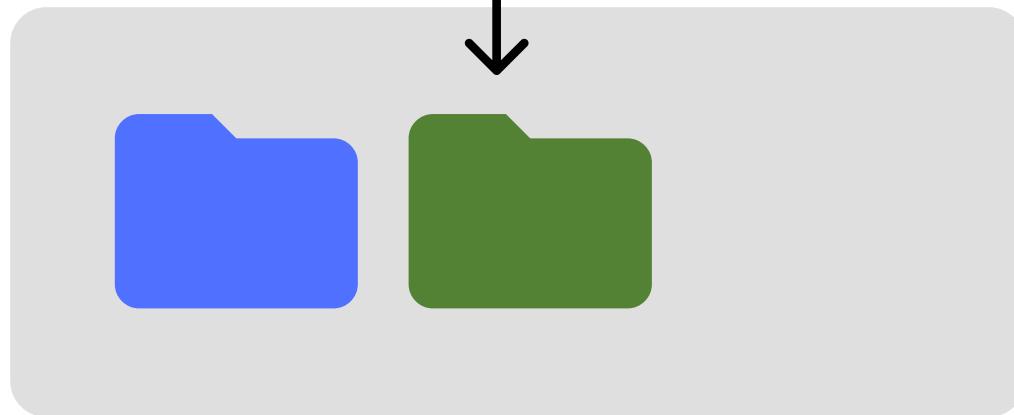
Local Repository

Computer



Remote Repository

GitHub



Local Repository

Computer

# Pull

- Push allows us to sync/download our files from remote to our local repository.

`git fetch`

`git pull origin <branch-name>`

**Warmup 18** ↗

# Warmup 18 ↪

- Create a new directory called **wup18** and navigate into it.
- Initialize a Git repository within the directory.
- Create a Python file: **greet.py**.
- Add and commit with the message “***initial commit***”.
- Write the following in “**greet.py**”:  
***english = “Hello”***

Now, add and commit with the message “***add English greet***”. View the commit history.

Contd..

# Warmup 18 ↧

- Create a remote repository in GitHub named as “**remote-wup-18**”.
- Add remote to the local repository to connect with remote repository created.
- Now, push the local repository to the remote repository created. Refresh the page.
- Clone the repository back into your computer.

Contd..

# Warmup 18 ନେପାଲୀ

- Navigate to cloned repository “**remote-wup-18**” in computer.
- Write the following in “**greet.py**”:  
**nepali = “Namaste”**  
Now, add and commit with the message “add Nepali greet”.
- Now, push the changes back to the remote repository.
- Navigate to old repository “**wup18**” in computer and view the commit history.

Contd..

# Warmup 18 ↪

- Now, fetch to the local repository “**wup18**” and view the commit history.
- Then, pull to the local repository “**wup18**” to sync the latest changes in your current local repository.



**Module 3 Completed**