Introduction to Git (Part-1)

Introduction:

- Ever felt frustrated because of not being able to recover a small **code snippet** which **got deleted accidentally**? If the answer to the above questions is yes, then this tutorial is for you.
- This tutorial assumes that you already have a **GitHub account** and the **Git Bash application installed** on your system If not follow step 1 or watch lecture video.

Defining Keywords

Before learning to use GitHub, let's understand some common terminologies which you will encounter throughout this tutorial:

- **Repository** — In layman terms, this is analogous to a project folder that contains all your project files. Standard practice is to have one repository per project.
- **Branch** — Generally, developers **use different branches for maintaining different modules of the project**. Another common scenario that warrants the use of branches is when **multiple members of the team want to work on the same piece of code.** This is when each one can have its branch. By default, **each newly created repository has a central branch** named "**master**".

- **Clone** — Cloning is like copying and pasting the repository from one drive(developer's folder on GitHub) to another (our local folder).
- **Stage & Commit** — Creation of a new project version, on your git repository, is a 2 step process. The first step is to collect all the files which are required to be a part of the new version. This is called **staging** the files. The second step is to create the new version of your project which is called **committing**. Only those files which are staged, can be committed to a new version.
- **Push & Pull** — Given our focus on GitHub, push and pull is about interacting with repositories stored on GitHub's cloud. A pull is like downloading the latest version and a push is synonymous to uploading your latest version on GitHub

Step-1: Git installation

- Firstly, install Git tools on your computer. There are different Git software, but it's better to install the basic one to start with. We will use the command line to communicate with GitHub.
- Once you are more comfortable with the command line, you can download Git software with a user interface.

For Ubuntu:

First, update your packages:

$ sudo apt update

Next, install Git with apt-get:

$ sudo apt-get install git

Finally, verify that Git is installed correctly:

```
$ git --version
```

For MacOSX:

First, download the latest Git for Mac installer.

Next, follow instructions on your screen.

Finally, open a terminal and verify that Git is installed correctly:

```
$ git --version
```

For Windows:

First, download the latest Git for Windows installer.

Next, follow instructions on your screen (you can leave the default options).

Finally, open a terminal (example: powershell or git bash) and verify that Git is installed correctly:

```
$ git --version
```

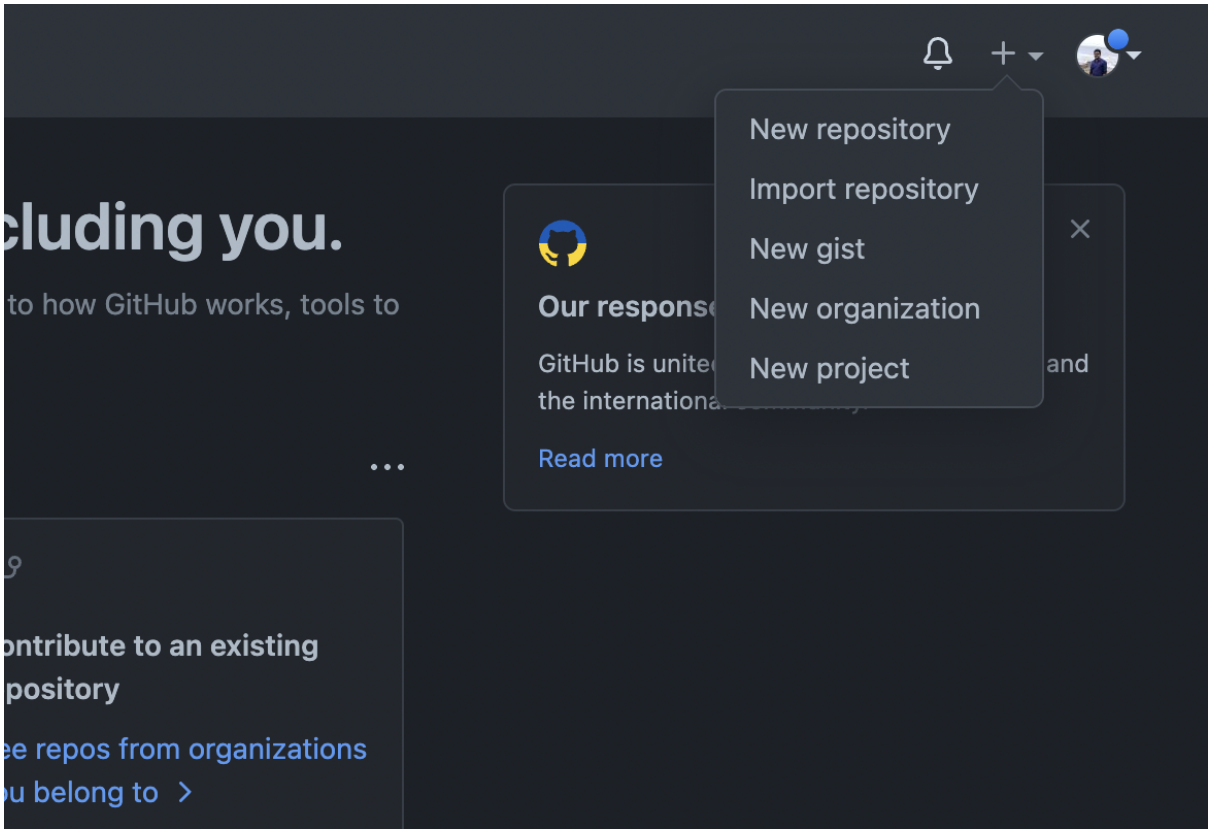
For all users:

- One last step is needed to complete the installation correctly! You need to run the following commands in your terminal with your information to set a default username and email when you are going to save your work.
- **This should be done only once and not for every project, since you are setting this up globally.**

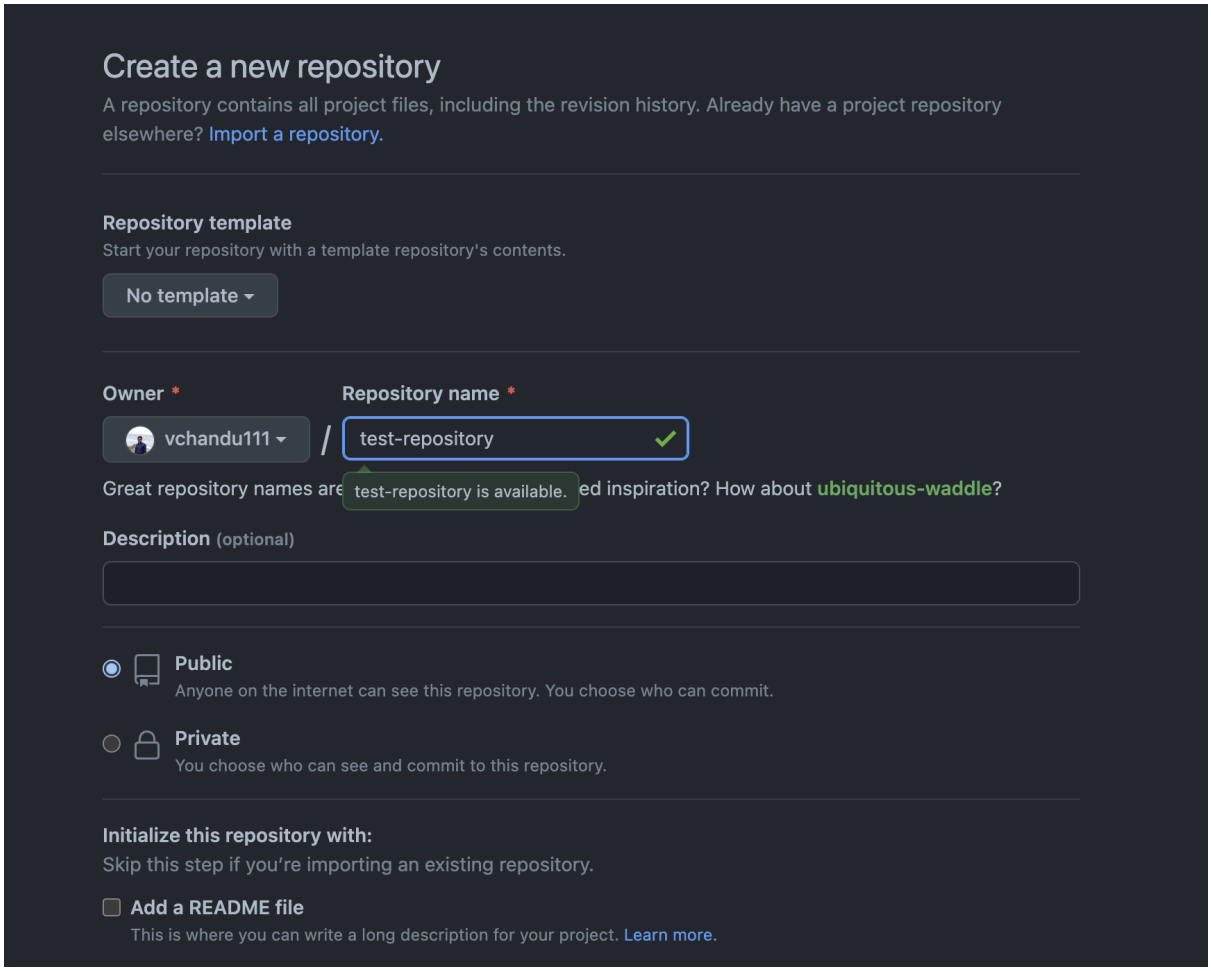
```
git config --global user.name "<your_name_here>"
git config --global user.email "<your_email@email.com>"
```

Step-2: Your first GitHub project!

- Go to the main GitHub page and click on the “+” icon in the menu bar. Once you click on this button, a new menu appears with a “*New repository*” entry. Click on it!

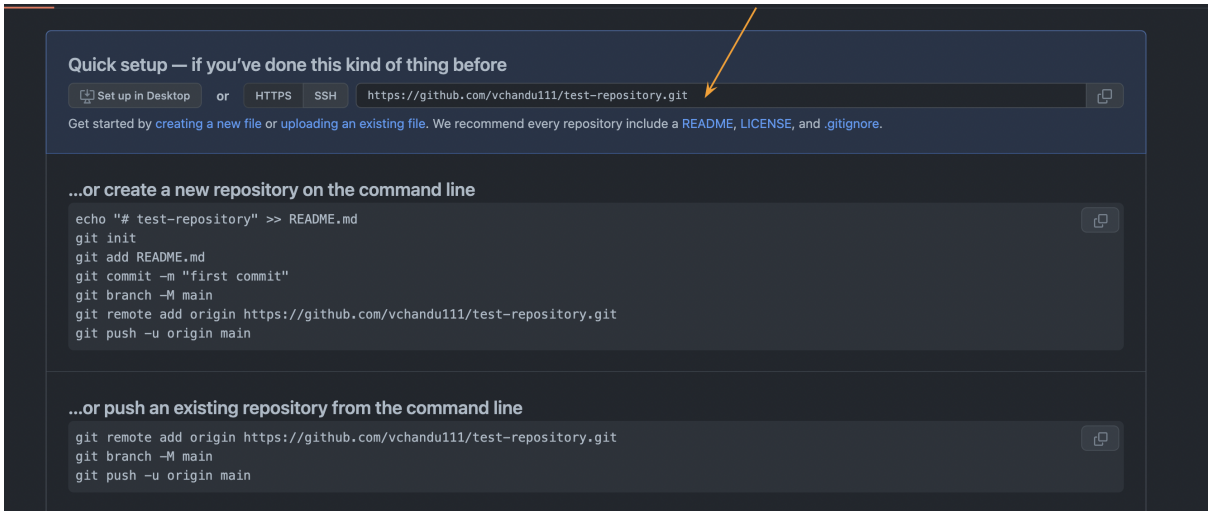


- The repository creation page will appear. Choose a cool name for your first repository and put a small description before clicking on the "*Create repository*" button.



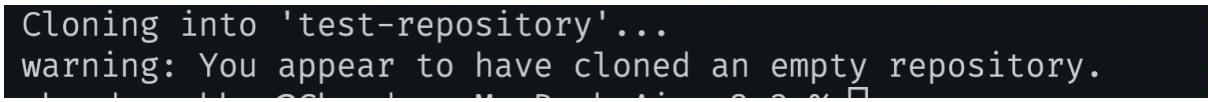
Step-3: A local version of your project

- Now get a copy of the repository on your computer. To do that, you need to "*clone*" the repository. On the repository page, you need to get the "*HTTPS*" address.



- Once you had the address of the repositories, you need to use your terminal (through shell commands) to move in the place where you want to put the directory copy (for example you can move in your "*Documents*" folder). When you are ready, you can enter:

```
$ git clone [HTTPS ADDRESS]
```



- Now, your repository is on your computer. You need to move in it with:

```
$ cd [NAME OF REPOSITORY]
```

- Now you can add all of your files to the staging area using command

```
git add .
```

- You can commit these changes with the command

```
git commit -m "<add a commit message here>"
```

- Once you've added and committed all your files, run this command to push all your commits to your repo.

```
git push origin <branchname>
```

Some Useful commands for Navigation from terminal