



*git*

<https://git-scm.com/>



Git is a *free* and *open source* distributed version control system designed to handle everything from small to very large projects with speed and efficiency.



- **Tracking files in a project**
- **Language independent (You can use git for any project with any type of file e.g., HTML, XML, Java, C, C++ etc)**
- **It is a distributed version control (VC) system or decentralized VC system. (i.e., many developers can work on a project from different physical locations or networks)**
- **Git track every single change that is made in the project or on a single file.**
- **You can revert back to a change you have made earlier.**

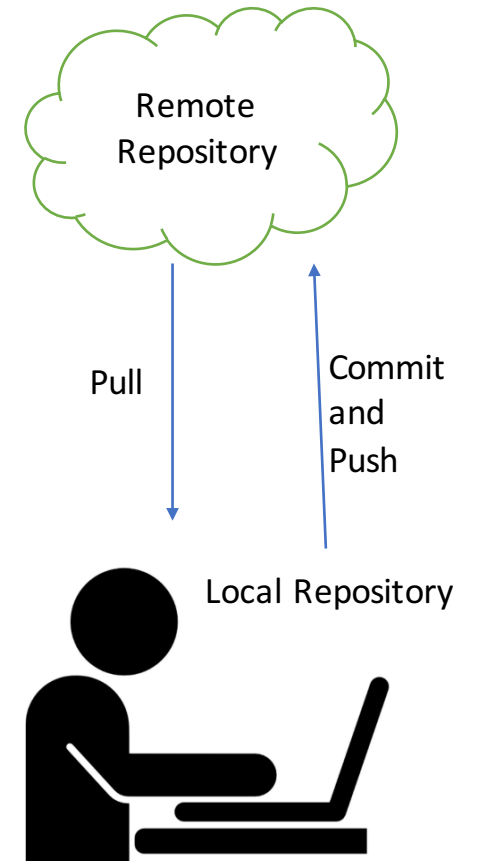


## How does it work?

You typically have two repositories when you are working with a VC:

1. Local repository
  2. Remote repository
1. You have a repository in your local machine. You usually only work and develop your project on your local machine in your local repository. (No internet needed)
  2. You have a remote repository that your local repository is linked to the remote repository. (It needs internet )

Whenever you made some changes in your project (local repository), you can **upload/push** your **files/projects** into the remote repository.





## git's features

1. Git can keep track of your code history
2. Snapshots: Every time you make changes and commit them into your remote repository, you are basically creating a snapshot of the current status of your project.
3. You can visit any snapshot at any time you want.
4. Your code is very safe on git. If you make a mistake in your project, you can easily revert back to the previous status (snapshot).



## git's commands

1. *git init* → It initialize local Git Repository on your machine.
2. *git add <file>* → add files and make them ready to commit
3. *git status* → you can check the status of working tree (e.g., you have files added but not committed yet)
4. *git commit* → when you are happy with your changes, you can commit your files. Commit should be done after adding them to the tree
5. *git push* → push file to the remote repository
6. *git pull* → pull the latest version of the project from remote repository.
7. *git clone* → It copies the repository from remote repository into your machine.



# Installing GIT

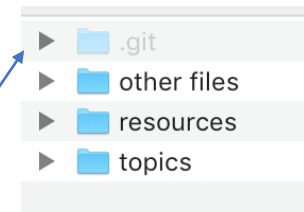
1. Mac users: <https://git-scm.com/download/mac>
2. Windows users: <https://git-scm.com/download/win>

This gives you a command line environment to work with git.



# Creating a local repository

1. Open the git terminal at your project folder
2. Type: `git init`
3. There will be a hidden folder created in that directory called `.git`

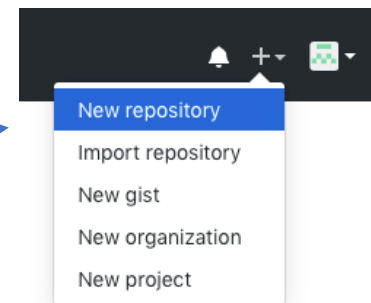
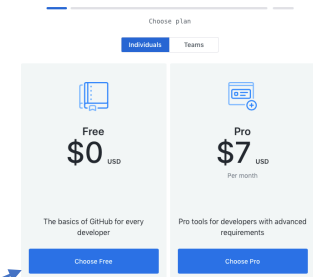






# Creating a remote repository

- In order to create a remote repository, first you need to choose a **version control repository hosting service**.
- There are number of VC repository hosing services such as github, gitlab and bitbucket.
- We will be working with github.
- Open an account at <https://github.com> if you don't already have one.
- Select the free option
- Once you finished with sign-up you can create a new repository; click on plus symbol and New repository (This will be your remote repository)





# Creating a remote repository

- You can have either public or private repository.
- Choose a name for your repository
- Press Create repository

Owner:  / Repository name:

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

Description (optional):

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

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

Skip this step if you're importing an existing repository.

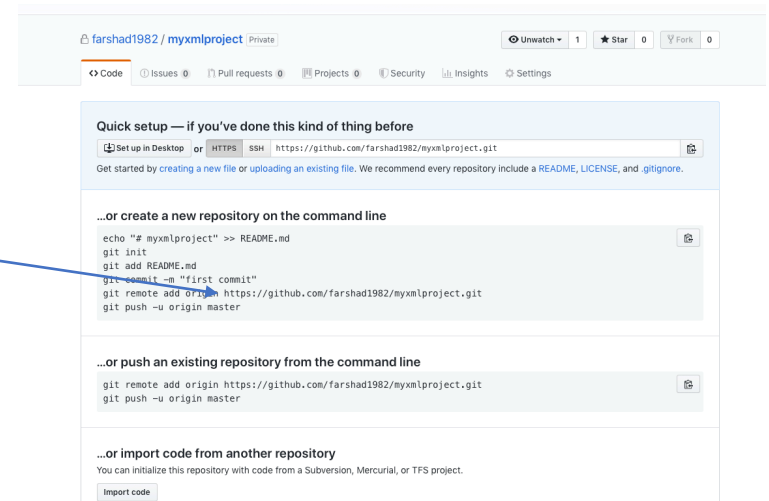
☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

Add .gitignore:  Add a license:  [?](#)



# Creating a remote repository for the first time

- Open the git terminal and provide the following command lines:
  - `git init`
  - `git add .`
  - `git commit -m 'Add your commit message'`
  - `git remote add origin https://XXXXXXXXXXXXXX`
  - `git push -u origin master`
- It asks your github username and password



Dr Farshad Ghassemi Toosi

CIT (Cork Institute of Technology)



## What to do next?

- Every time you make some changes on your project (local repository) you should push it into the remote repository to have it in a safe place.
- In order to do that, you can use the following commands:
  - ***git add .*** → dot (.) means push all the files in the project, if you want to push only one particular file then you use: ***git add myfile.xml***
  - ***git commit -m "always provide good message so in future you know what was the update about"***
  - ***git push origin master*** → this pushes the files you have added earlier on the remote repository.



# What is git branch

One of the very useful features of git is the functionality to create multiple branches for the project.

## What is a branch ?

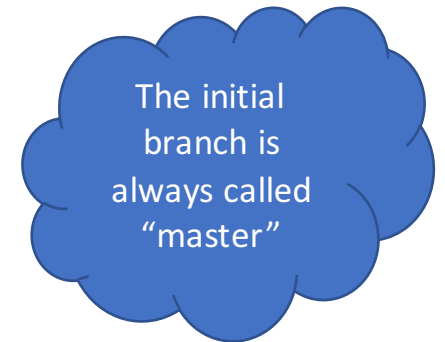
- **Branch** is simply an independent copy of your project.
- Using **branches** you can create multiple copies of your project.

## Benefits

- Multiple developers can work on different branches without conflict.
- One developer can work on multiple different branches working on different parts of a project independently.
- Branches can be merged into one branch.



# What is git branch

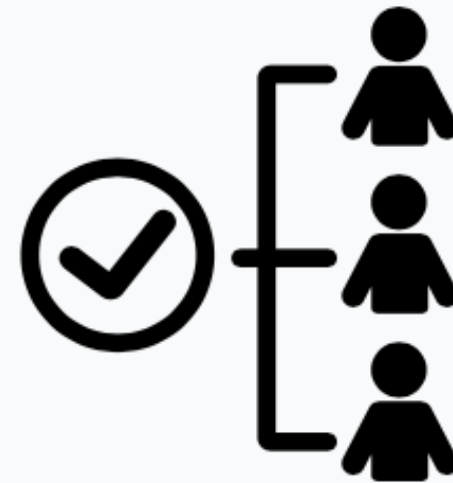


- ***git branch mynewbranch*** → this will create a new branch (copy) of your project with the name "mynewbranch"
- ***git checkout mynewbranch*** → this will navigate to your new branch and the changes will be applied on the new branch.
- ***git checkout master*** → this will bring the master (initial) branch



## Working as a group on a project using git

Git provides you with a secure and organized environment to work as a group on a project.





First you need to share your project with your colleagues.

1. Navigate to your repositories
2. Click on the repository you want to share
3. Click on setting
4. Click on collaborators
5. Provide the username of a person you want to share your repository with and press add collaborator

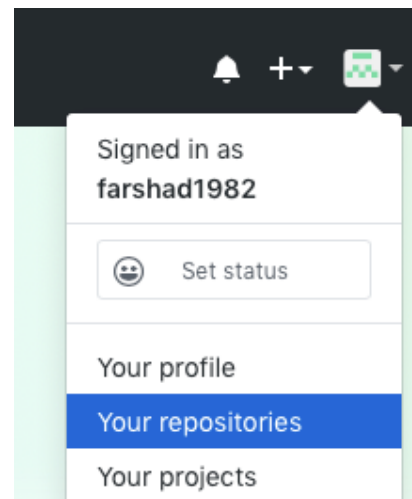
See the visual steps in the next slides:





# How to share your project with your colleagues.

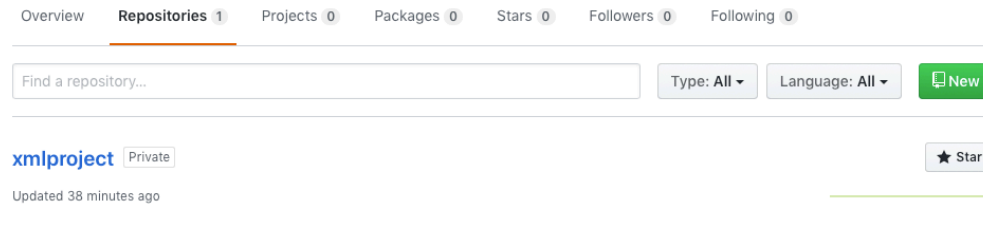
1. Navigate to your repositories





## How to share your project with your colleagues.

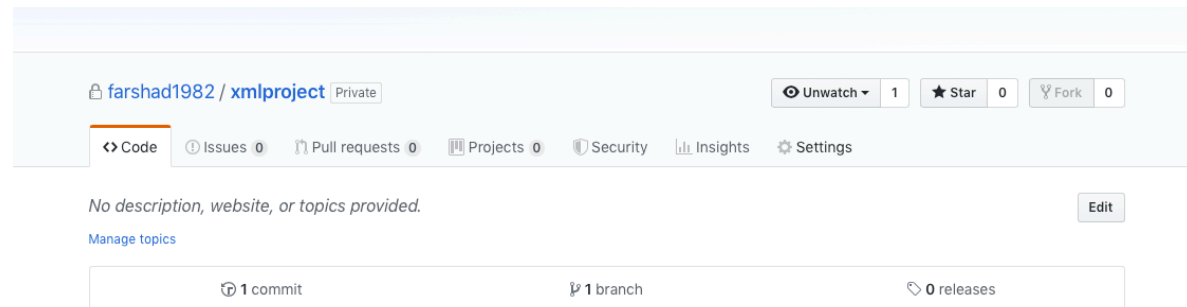
1. Navigate to your repositories
2. Click on the repository you want to share





## How to share your project with your colleagues.

1. Navigate to your repositories
2. Click on the repository you want to share
3. Click on setting





## How to share your project with your colleagues.

1. Navigate to your repositories
2. Click on the repository you want to share
3. Click on setting
4. Click on collaborators

Options	Settings
Collaborators	Repository name
Branches	<input type="text" value="xmlproject"/> <button>Rename</button>
Webhooks	<input type="checkbox"/> <b>Template repository</b>
Notifications	<small>Template repositories let users generate new repositories with the same name. For example, farshad1982/xmlproject can be used as a template for creating other repositories.</small>
Integrations & services	Social preview
Deploy keys	



## How to share your project with your colleagues.

1. Navigate to your repositories
2. Click on the repository you want to share
3. Click on setting
4. Click on collaborators
5. Provide the username of a person you want to share your repository with and press [add collaborator](#)

A screenshot of the GitHub 'Collaborators' settings page for a repository. On the left is a sidebar with navigation links: 'Options', 'Collaborators' (highlighted with an orange bar), 'Branches', 'Webhooks', 'Notifications', 'Integrations & services', and 'Deploy keys'. The main content area is titled 'Collaborators' with a subtitle 'Push access to the repository'. It shows a user 'fery13' with a status 'Awaiting fery13's response' and buttons for 'Copy invite link' and 'Cancel invite'. Below this is a search section titled 'Search by username, full name or email address' with a note: 'You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.' A search input field contains 'fery13' and an 'Add collaborator' button. At the bottom, it shows '1 of 3 collaborators' with a green progress bar.