# **Introduction to GIT**

## **What is Git?**

Git is a *version control system* which lets you track changes you make to your files over time. You can revert to various states of your files. You can also make a copy of your file, make changes to that copy, and then merge these changes to the original copy.

## **What is GitHub?**

GitHub is an online hosting service for Git repositories.

## **HOW TO GET STARTED**

1. Create a GitHub account from <https://github.com/>
2. Install Git on your local PC

Git comes preinstalled in some Macs and Linux-based systems, but you can always check if you have Git installed in your machine by typing *git version* in your terminal.

You can download Git [here](https://git-scm.com/download) and then select your operating system to download.

1. Connect your GitHub account to your Git account. You'll do this from your terminal. On your local PC open Gitbash and

* Set your username, execute these commands:

*git config --global user.name "YOUR\_USERNAME"*

* And Set your email

*git config --global user.email "YOUR\_EMAIL".*

**Note:** replace "YOUR\_USERNAME" and "YOUR\_EMAIL" with the values you choose.

* To confirm that you have set your Git email correctly, type:

*git config --global user.email and*

*git config --global user.name*

1. Create a new repository on the Github Website: Repositories are like your code folders online.
2. Create a README.md file. The README.md file helps you describe your GitHub profile, and you can use it to show what you're currently learning along with your skills and contributions. You can use the command prompt type:

git add [*Readme.md*](https://readme.md/)

1. Change directory with cd e.g *cd desktop*
2. Create and Initialize a Project in Git: This will tell Git to get ready to start watching your files for every change that occurs. Now to initialize git in your project directory,run

*git init*.

1. Git clone: Make a copy of your repository online to your local system. Navigate to your project folder and Type this on the command line interface:

*git clone https://github.com/kofodym/TCM-devops.git*

But if the project already exists locally so we will use commands:

*git remote add origin https://github.com/kofodym/TCM-devops.git git branch -M main*

*git push -u origin main*

1. Git add : Here we add files to be staged. For our files to be tracked by Git we use this *command git add FileName or git add .*  It lets you add *all* files in the present folder.
2. Git Status: to know what state your file is in, you can run the *git status* command.
3. Git commit: After the staged state is the committed state. To commit our file, we use *the git commit -m "Write your message"*command. Use -m for message followed by the actual message.

Files in the committed stage are files ready to be pushed to the remote repo (on GitHub).

1. Git push: *git push -u origin main*
2. Branching in Git:

git branch creates a new branch which is a new version of the repository as it appears when added. After creating a branch you checkout.

checkout tells Git it is supposed to switch to a new branch. -b tells Git to create a new branch. branchName is the name of the branch to be created and switched to.

Type: *git branch NewBranch*

Then type: *git checkout*

Or *git checkout –b NewBranch*

Set upstream branch: *git remote add origin repository\_link*.

 You can check all the branches that exist in your repo by running the *git branch*

git push -u origin main pushes the code to GitHub. The -u flag creates a tracking reference for the branch, and origin main puts the code in the main branch.

1. Git pull: The online repository updates itself with the contents on the local repository on your system

# Git merge:  merge the changes we made in the Newbranch into the main branch by running

# *git merge NewBranch*