Git

What is Git: Git is a version-control system (i.e., a piece of software) that helps you keep track of your computer programs and files and the changes that are made to them over time. It also allows you to collaborate with your peers on a program, code, or file.

Where to download git: https://git-scm.com/

Using Git with teammates:

Connect your GitHub repo with your computer

Now, it's time to connect your computer to GitHub with the command:

```
git remote add origin https://github.com/<your_username>/Demo.git
```

Let's look at this command step by step. We are telling Git to add a remote called origin with the address https://github.com/<your_username>/Demo.git (i.e., the URL of your Git repo on GitHub.com). This allows you to interact with your Git repository on GitHub.com by typing origin instead of the full URL and Git will know where to send your code. Why origin? Well, you can name it anything else if you'd like.

Now we have connected our local copy of the *Demo* repository to its remote counterpart on GitHub.com. Your terminal looks like this:

Update & merge

to update your local repository to the newest commit, execute

- git pull

In your working directory to *fetch* and *merge* remote changes.

To merge another branch into your active branch (e.g. master), use

- git merge <branch>

In both cases git tries to auto-merge changes. Unfortunately, this is not always possible and results in *conflicts*. You are responsible to merge those *conflicts* manually by editing the files shown by git. After changing, you need to mark them as merged with

- git add <filename>

Before merging changes, you can also preview them by using

- git diff <source_branch> <target_branch>

Branching

Branches are used to develop features isolated from each other. The *master* branch is the "default" branch when you create a repository. Use other branches for development and merge them back to the master branch upon completion.

Create a new branch named "feature_x" and switch to it using

- git checkout -b feature_x

Switch back to master

- git checkout master

And delete the branch again

- git branch -d feature_x

A branch is *not available to others* unless you push the branch to your remote repository

- git push origin <branch>

Pushing Changes

Your changes are now in the **HEAD** of your local working copy. To send those changes to your remote repository, execute

- git push origin master

Change *master* to whatever branch you want to push your changes to.

If you have not cloned an existing repository and want to connect your repository to a remote server, you need to add it with

- git remote add origin <server>

Now you are able to push your changes to the selected remote server

Synchronize changes

Synchronize your local repository with the remote repository on GitHub.com

- git fetch

Downloads all history from the remote tracking branches

- git merge

Combines remote tracking branches into current local branch

- git push

Uploads all local branch commits to GitHub

- git pull

Updates your current local working branch with all new commits from the corresponding remote branch on GitHub. git pull is a combination of git fetch and git merge