How to use GIT

First give a look to this page https://help.github.com/articles/set-up-git

There are two ways to use repositories, with HTTPS urls or SSH. In the set-up page you can see more information and choose what to use.

Creating a Repo

https://help.github.com/articles/create-a-repo here you can see how to create a repository. A repository is like a folder that you will upload to the internet and you can choose who can collaborate on it.

You need to create it in the github page and in your computer as a folder in your hard-disk.

They can have different names.

When you already created it on internet and your local user of git (done in the set-up) is already set, you can go with GIT Bash to the folder you created in your hard-disk.

You might do that with commands like

```
cd project-folder
```

You can see in what folder are you in GIT Bash with commands like

`pwd` *this shows you the path of your folder*

or

`ls` *this shows you what is in your folder*

When you are in the folder you want to upload do the next commands:

`git init` *this make a .git folder inside*

`git add .` *this adds the folder (. means the folder where you are) to be uploaded when you submit your changes*

`git commit –m ‘message‘` *this commits a local version of your changes. It won’t upload the internet version yet. You can do* *as many commits as you want before submitting the final version to internet.*

```
git remote add origin https://github.com/username/project-name.git
```

this will finally connect your local (on the hard-disk of your PC) version with the internet version of the repository. You may change the URL given in this command to the exact URL of the project you created on internet. You can see the URL on the github page. If you are using SSH instead of HTTPS URL you may change the URL to something like this:

- git@github.com:user_in_github/repo_name.git

And to submit the changes to the internet you will need to do this command

```
git push origin master
```

Then you have your repo created.

You can create a repo in other way too, first doing the webpage version and then going to the folder where you want to store your local version of the repo, for example in ~/workspace (the ~ means the user folder for example C:/Users/username/) and then downloading the internet version to your hard-disk, so it will be an empty folder with the git settings inside(the .git folder). For example this:

```
cd workspace
git clone git@github.com:user_in_github/repo_name.git
cd repo_name
```

and then you have your repo with the local and internet version and you don't need to do `"git init"`

Downloading a repo already created

If someone is sharing a repo to you, you should "clone" it in the folder you want to download it. This is the command:

```
git clone https://github.com/username/project-name.git
```

Remember you may change the URL for the one you are using whether HTTPS or SSH.

Submitting changes

So there will always be two versions of the repository, the internet version (that you can see in the github page) and the local version (in your PC).

Sometimes the two versions can be the same… Sometimes the internet version is different because a collaborator submitted changes and you haven’t updated your version yet.

And sometimes your local version is different to the internet version because you haven’t submitted your changes yet.

So for those problems GIT has the next commands:

```
git pull origin master
```

Updates your local version to the internet version.

```
git add file
```

Adds the file so it will be submitted the next time you commit and push, you will probably change “file” for ” .” so that will add all the directory or for “main.cpp” so will only change that file.

```
git commit –m ‘you can write a message about your commit here’
```

Commits the changes in the added files but just in a local version, this won’t change the internet version yet.

```
git push origin master
```

This will upload the added and committed files to the internet version.

Always do those commands in that order: pull, add, commit and push. Remember to pull always before submitting changes.

Usual conflicts

If your local version is different to the internet version for example in the “main.cpp” and you use pull, sometimes GIT cannot merge the versions and it will give you an error.

If you prefer to discard your changes and have the internet version you may do:

```
git checkout HEAD .
```

HEAD means the last commit done, you can change “HEAD” for other commit than the last one using the hexadecimal number that represent a commit and you can change “.” for any file, the point just means that this will change all the folder, but you can put just “main.cpp” for example and will only change that file.

If you prefer to submit your changes any way you may commit first, then pull, and then push.