Git & GitHub

Master Branch – is the Core Branch Original

Step 1
- Create a Dir then Cd into it.

Step 2
- initialise the Git by typing: < git init > this makes it a git Repo

Step 3
- Staging Area type: git add . Or git add <name of the file with ext>

Step 4
- Check what we have done type: git status to unstage type: git

rm --cached <file>... to unstage

Step 5
- Is to commit the files (all this is still takes place your local machine)

git commit -m “<your message>” (under 50 chars only)

step 6
- After commit check with git status if you have made any changes to your files.

```
#> On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
      modified:   Readme.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Step 7
- recommit after changes made git add . (. means add all files)
- git status

```
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
      modified:   Readme.md
```

again you need to commit git commit -m “<your message>” (under 50 chars only)

```
[master d9a18be] 2nd Commit
 1 file changed, 3 insertions(+), 1 deletion(-)
```

Master Branch

Check Out to new BRANCH to avoid messing up the master branch

git checkout -b <name of new branch>

eg

```
git checkout -b new
      Switched to a new branch 'new'
```

```
#> git commit -m "1st Commit on new Branch called New"
      [new 303ced5] 1st Commit on new Branch called New
      1 file changed, 3 insertions(+), 1 deletion(-)
```

To change Branches

type: git checkout <Branch Name>

Eg: git checkout master

```
      Switched to branch 'master'
```

How to Bring files from one branch to another (merge) so in the NEW branch type the command to merge everything in new (or select) to the master

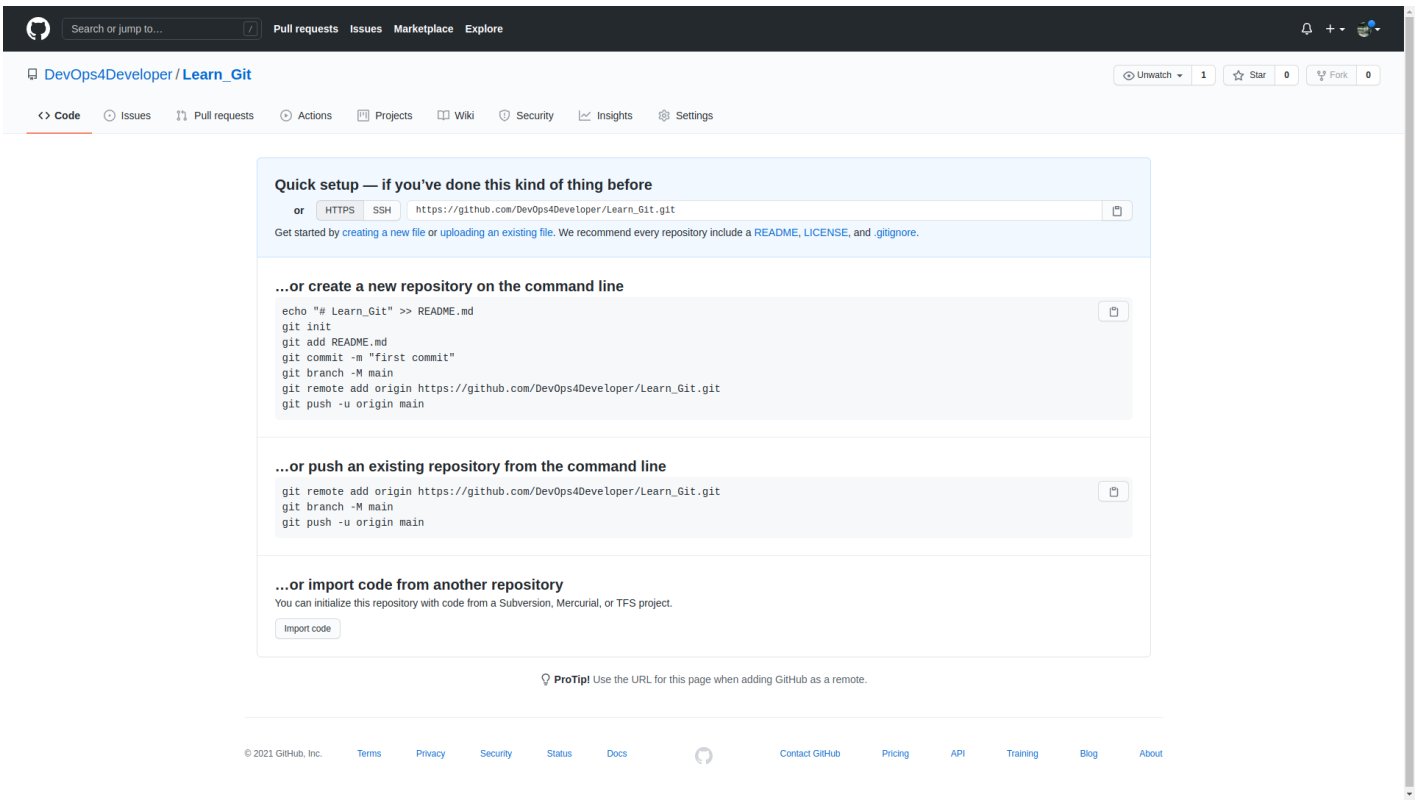
git merge master (or NEW to Master)

Eg. we are in MASTER branch and we want to merge to NEW Master > New because we want to NOT damage the master is why we created a NEW branch once satisfied we can bring our master branch to our new branch and visa versa just remember the branch that you are in is fulling in from what ever Master branch you want.

GITHUB

Now we want to upload our commits to github

goto your github account > Create New Repository



Quick setup — if you've done this kind of thing before

Or

can use : https : https://github.com/DevOps4Developer/Learn_Git.git

or SSH : git@github.com:DevOps4Developer/Learn_Git.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

HTTPS

…or create a new repository on the command line

```
echo "# Learn_Git" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/DevOps4Developer/Learn_Git.git
git push -u origin main
```

…or push an existing repository from the command line

```
git remote add origin https://github.com/DevOps4Developer/Learn_Git.git
git branch -M main
```

git push -u origin main

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

!! IMPORTANT !!

MAKE SURE YOU CHECK WHAT BRANCH YOU ARE IN FIRST

TYPE: git checkout master

at this stage since you have aleady used the previous commands upto :

git branch -M main

you only need the last 2 lines also know as UPSTREAM

git remote add origin https://github.com/DevOps4Developer/Learn_Git.git

&

git push -u origin main

```
#> git push -u origin master
Username for 'https://github.com': DevOps4Developer
Password for 'https://DevOps4Developer@github.com':

Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 722 bytes | 361.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/DevOps4Developer/Learn_Git.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

So your local computer will remember your Username and Password so next time you just need to type :

git push -u origin <branch name> **(git checkout <branch name>)**

if it does not work ie push to remote repository the set the account up like so:

git config –global user.name “Name for ID”

git config –global user.email “your github email”

SSH ssh you will need to set up ssh public/private keys on local computer and github.

…or create a new repository on the command line

```
echo "# Learn_Git" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:DevOps4Developer/Learn_Git.git
git push -u origin main
```

…or push an existing repository from the command line

```
git remote add origin git@github.com:DevOps4Developer/Learn_Git.git
git branch -M main
git push -u origin main
```

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

How to PULL from the Repository

Any changes you make on the remote Repository you can PULL from to your local repo on your PC.

Type: `git pull origin master`

Will give You:

```
#> git pull origin master

      remote: Enumerating objects: 5, done.
      remote: Counting objects: 100% (5/5), done.
      remote: Compressing objects: 100% (3/3), done.
      remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
      Unpacking objects: 100% (3/3), 705 bytes | 705.00 KiB/s, done.
      From https://github.com/DevOps4Developer/Learn_Git
       * branch            master     -> FETCH_HEAD
       8be2f8e..9a98bde  master     -> origin/master
      Updating 8be2f8e..9a98bde
      Fast-forward
      Readme.md | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

???? if you get 2 of the same files you get a conflict so you need to pull from the remote repo to the local pc then ====> merge conflict

******So if you made changes to the remote repo but not reflected on your local repo on pc the you will be asked to pull from remote repo first.**

You will need to match the content of the local and remote repo to make the commit work.

Don't forget to : `git add .` Then `git commit -m "change"` and `git push`

So NOW to PUSH to another BRANCH

So switch to the branch you want to use : `git checkout <Branch Name>`

Now you want to PUSH to the REMOTE NEW branch

type: `git push origin <new-branch name>`

may ask you for Credentials

if ## fatal: The current branch new has no upstream branch.
To push the current branch and set the remote as upstream, use

```
 git push --set-upstream origin new
```

SUMMERY

create a new repository on the command line

Goto the folder you are working on Cli and make your files as normal

Initialise your folder with git (.git)

#> git init

#> git add <files name> or .(for all files)

#> git commit -m "first commit"

#> git branch -M main

To push to remote repository

#> git remote add origin https://github.com/DevOps4Developer/Learn_Git.git
The URL can be found on the github account when you create a repo.

#> git push -u origin main

push an existing repository from the command line

```
git remote add origin https://github.com/DevOps4Developer/Learn_Git.git
git branch -M main
git push -u origin main
```

The Same can be done using ssh but will need to configure ssh public private keys first before commit and push to remote

!! IMPORTANT !!

Just Remember only files in folders were git init was executed will be your repo, you cant go back a folder and expect git to work, you can git init a whole folder set or individual folders within that folder.