Hello friends :wave: :smiley: , when we talk about **git** and **github** then, there are lot of confusion :confused: exist in our mind. Especially, when we talk about their commands. I also encountered with the same issue and mentors helped me and motivate me to face this and gave me some suggestions in order to overcome this. And thanks to the mentors, who suggested me to write an article about it. So, here I am writing about some concepts, which are related to **github** and **git** and some commands(which you can use it in *git bash*, *cmd* or *vscode terminal*) too, which, I hope, would be very helpful for everyone.

So, here we are discussing about following terms, which might be very important:

- Fork
- Branches
- Push commit to a remote repository

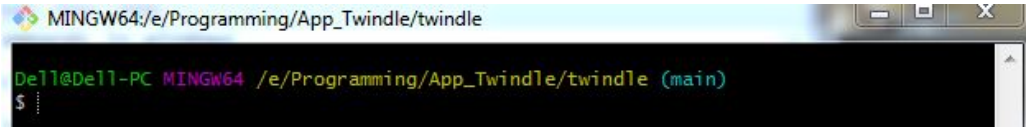
Fork

When we **fork** a repository, in our case we **fork** the **Twindle** repository then, we get a personal copy of the repository in our **github** account, where we are free to make changes without affecting the main project. When we do some changes in our own repository and want to implement these changes in the main project then, we raise pull request(PR). The authority of the main project review the changes and if they think it is good to include this change in the main project then, they add it(you everytime want this to happen) otherwise, reject it(sorry, but it happens too).

Branches

Branches are almost like a new copy of our code at the current state(it means you forked a repository, which becomes "master" version in your repository and then, you make a copy of your "master" version), which can then be used to develop new code.For example, whenever we need to create a new feature, or rewrite any of our code, it's a good thing to create a new branch so that none of our changes affect the "master" version of the code. This is important since it can be very difficult to revert code changes from memory, especially in complex systems. And also there are many changes happen in the "master" version of the main repository like in **Twindle**, people work on the code and raise PR requests. If we try to raise PR request from our "master" version then, the PR requests raised by the people will also get added in our raised PR and this creates mess. So, to avoid these problems, we need to create a branch, raise PR request from that branch, this will avoid others' PR requests and *when the PR request is approved then, that branch must be deleted.*

First of all, when we are creating a branch using *git bash* then, we must assure that we are in the main branch as shown below:



Here, in the above photo, "(main)" shows that we are in the main branch. Now, there are certain commands we should know and these are as follows:

- To initialize a repository for a new or existing project,

```
git init
```

- To create a branch from the current branch

```
git branch <branch-name>
```

  this makes a copy of main branch,

```
Dell@Dell-PC MINGW64 /e/Programming/App_Twindle/twindle (main)
$ git branch branch2
```

- To see how many branches in the git or local,

```
git branch
```

```
Dell@Dell-PC MINGW64 /e/Programming/App_Twindle/twindle (main)
$ git branch
  branch1
  branch2
* main
```

  By seeing the photo, we also conclude that we are in the main branch because of the * .

- To see all the branches (local as well as remote),

```
git branch -a
```

```
Dell@Dell-PC MINGW64 ~
$ cd E:/Programming/App_Twindle/twindle

Dell@Dell-PC MINGW64 /e/Programming/App_Twindle/twindle (main)
$ git branch -a
  branch1
  branch2
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/branch1
  remotes/origin/main
  remotes/origin/revert-779-feature/esm
```

  From the above picture, we can see the local as well as remote branches(in red color).

- To switch from current branch to another branch,

```
git checkout <branch-name>
```

```
Dell@Dell-PC MINGW64 /e/Programming/App_Twindle/twindle (main)
$ git branch
  branch1
  branch2
* main

Dell@Dell-PC MINGW64 /e/Programming/App_Twindle/twindle (main)
$ git checkout branch1
Switched to branch 'branch1'
Your branch is up to date with 'origin/branch1'.

Dell@Dell-PC MINGW64 /e/Programming/App_Twindle/twindle (branch1)
$
```

Here, first we in main branch, see all branches in the local and then, we switch to another branch.

- To delete the branch but first, you need to go into the master or main branch then,

```
git branch -d <branch-name>
```

```
Dell@Dell-PC MINGW64 /e/Programming/App_Twindle/twindle (main)
$ git branch -d branch2
Deleted branch branch2 (was 2dee7d5).

Dell@Dell-PC MINGW64 /e/Programming/App_Twindle/twindle (main)
$ git branch
  branch1
* main
```

And again check how many branches are in your local.

Note: if it shows error: The branch is not fully merged then, please use,

```
git branch -D <branch-name>
```

- To add one or more files to staging(index) area,

```
git add <file_name>
```

- To list the files you have changed and those still need to add or commit,

```
git status
```

- To remove files from staging area(unstage),

```
git rm --cached <file_name>
```

- To commit changes to head(here, head means where you are),

```
git commit -m "commit message"
```

- To see the total commits you have done,

```
git log
```

Push commit to a remote repository

Commit means to save your changes in our local repository. When we do commit in the local repository then, we need to push this commit in remote repository or our github account. We use `git push` to push commits to do this.

The `git push` command takes two arguments:

- A remote name, for example, `origin`
- A branch name, for example, `branch1`

```
git push origin branch1
```

then, go to the github account where, we can see a branch of same name is created with the changes we did in the local. From that same branch that is, branch1, raise pull request. *This PR would be clean and without any mess. After approval or rejection, please remove that branch from remote and local.*

I urge everyone to please come forward and add more commands, if possible try to add it with suitable examples so that, it gives the better idea about it.

Suggestions are welcome :blush: !!