Getting started with GIT

Setting the GIT account and syncing remote repository to GitHub is sometimes tricky. Follow the steps for setting the remote GIT repository and GIT account.

1. Generate SSH key from remote server and add it to GITHUB SSH KEYs.

(Don't use '$' sign in the beginning as in do not copy it to the terminal)

```
$ git config --global user.name "username"
$ git config --global user.email "user-email-address"
```

Remove the previous origin if present

```
$ git remote rm origin
```

Add the Git repo URL as origin

```
$ git remote add origin SSH-url-of-GIT-repo
```

Generating the SSH key for adding it to the GIT account

```
$ cd ~/.ssh
$ ssh-keygen
```

Opening the generated key for displaying it in the terminal. Copy the key and add it to the git account.

```
$ cat ~/.ssh/id_rsa.pub
```

Add this key to your github account.

```
$ ssh -T git@github.com
```

You will get a welcome message in your console.

Now go your project folder. git push -u origin master, now it works!

2. Steps to follow before pushing the files from the repository to GIT account.

```
$ git init
```

```
$ git remote add origin SSH-URL-of-GIT-repo
```

(Do not forget to add all the files and commit the changes before pushing the files to GIT account)

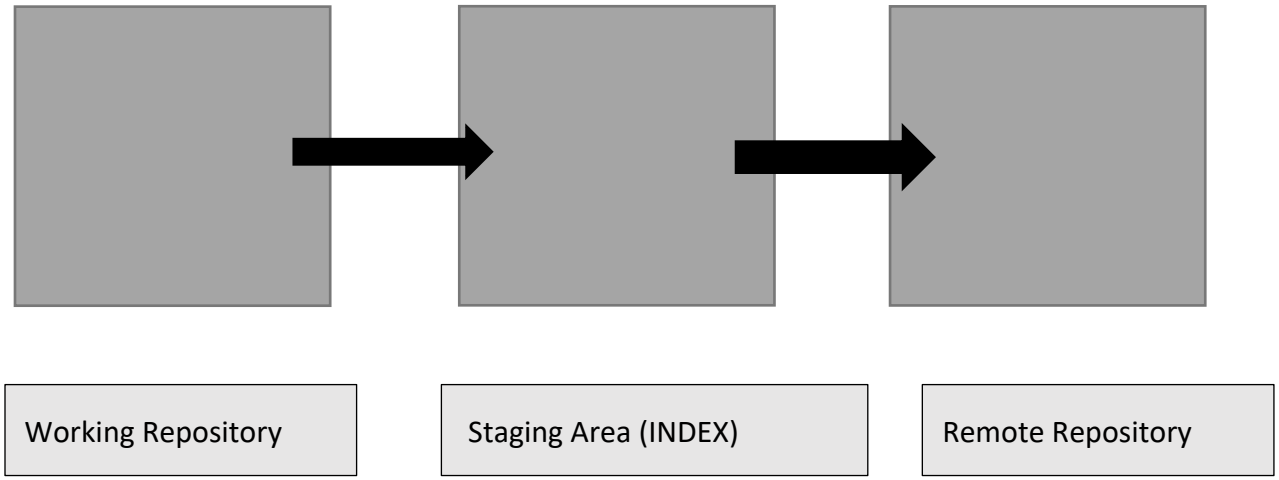
```
$ git add .
```

```
$ git commit -m "initial commit"
```

```
$ git push origin master
```

---

3. There are three different areas while you work on your project using Git, they are working repository, staging area (also known as INDEX) and the remote repository (also known as origin).



`$ git add .`
It adds all the changes that you have made on the working repository to the staging area.

`$ git commit -m “initial commit”`
It commits all the changes to the staging area to the remote repository.

`$ git log`
Use this command to see all the commit history.

`$ git status`
Use this command to see all the modifications done to the staging area and working repository.

`$ git diff`
Use this command to see the differences between the working reposiroty and the staging area.

`$ git diff --staged`
Use this command to view the difference between the staging area and the remote repository.

HEAD: HEAD is actually your current commit, or you can say it is your latest commit.
Origin: The remote repository is known as origin, which is present on the GitHub account.

`$ git reset --hard HEAD`
It will remove all the tracked and the untracked changes, it means it will remove all the work done that is saved to the staging area ($ git add .) or not saved to the staging area until the latest commit. The removed data can-not be brought back.

`$ git reset --hard origin/master`
It will remove all the uncommitted changes from the current branch and will make it same as remote repository.

`$ git push origin master`
Meaning: It means to push all local changes from master to your remote repository origin.

`$ git push origin master:my_data_branch`
It means that you are pushing changes from your local master to remote origin, but it is renamed to origin/my_data_branch.

`$ git branch new_branch`
It means you are creating a new branch named as new_branch.

`$ git checkout new_branch`
It means you are now moved to the new_branch.

`$ git checkout -b new_branch`
It means that you have created a new branch new_branch and you have moved to the newly created branch new_branch.

`$ git merge new_branch master`
Note: Before merging the two branches, use git checkout master and then use this command so that you stays on Master and the new_branch is merged to the master branch.

`$ git -d new_branch`
Used for deleting a branch.

`$ git rm --cached -f *.DS_Store`
Used for removing .DS_Store folder from the remote repository as it was already present before adding the .gitignore to the remote repository.