**Git Basics Cheatsheet**

**To configure git**

git config --global user.email "*useremail@gmail.com*"

git config --global user.name "*username*"

**To create local git repository**

|  |  |
| --- | --- |
| cd *projectfoldername* | Change directory to the project folder. |
| git init | Initialize empty Git repository (add a local Git repository to the project). |

**Staging and committing the code**

**Committing** is the process in which the code is added to the local repository. Before committing the code, it has to be in the **staging** area. The staging area is there to keep track of all the files which are to be committed. Any file which is not added to the staging area will not be committed. This gives the developer control over which files need to be committed.

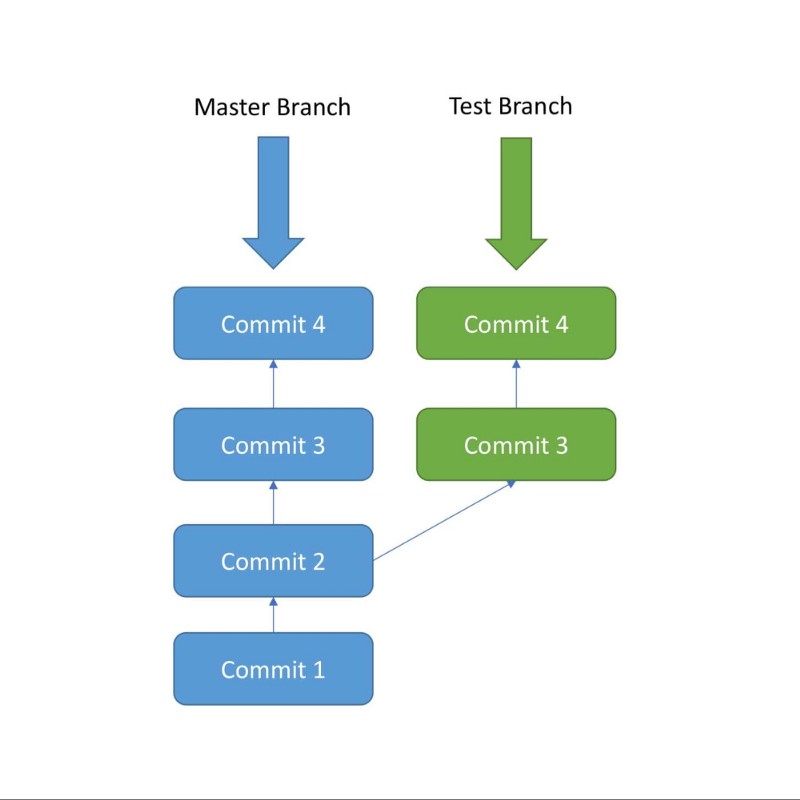
|  |  |
| --- | --- |
| git add *demo.txt*  git add *file1 file2 file3*  git add *.* | Staging the file (single,  multiple, or  all from project folder). |
| git commit -m "Initial Commit" | Commit the file. “Initial Commit” is the commit message here. Enter relevant commit message to indicate what code changes were done in that particular commit. |

**Git Status and Git Log**

|  |  |
| --- | --- |
| git status | To find out information regarding what files are modified and what files are in the staging area, and other information. |
| git log | To print out all the commits which have been done up until now. The log shows the author of each commit, the date of the commit, and the commit message. |

**Branches**

By default, Git commits go into the **master** branch. A branch is a pointer to the latest commit in the Git repository. Multiple branches are needed to support multiple parallel developments. Example below,



|  |  |
| --- | --- |
| git branch test | Create new branch called ‘test’. |
| git checkout test | To switch to the ‘test’ branch. |
| git branch | List out all the branches in local. |
| git checkout master  get merge test | Go back to the master branch and  merge the code from the ‘test’ branch into the master branch (Have to resolve conflicts, if any) |

**Remote Git Repository (e.g., GitHub)**

Each developer will work in their local repository but eventually, they will push the code into a remote repository. Once the code is in the remote repository, other developers can see and modify that code.

Create new Git repository in Github. Example below shows with repository name = ‘test’

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| git remote add origin *[repository url]* | To point the local repository to the remote repository. |
| git push --all -u | To push **all** the branches from local repository into the remote repository. |
| git push -u origin master | This pushes the code from the master branch in the local repository to the master branch in the remote repository. |
| git pull origin master | To pull the latest changes from the remote repository into the local repository. |
| git clone *[repository url]* | To clone an existing remote repository into your computer. |

\*-u short for –set-upstream