Git 02 Secondary Cheat Sheet

```
<!-- ***|Configuration|*** -->
<!-- to config user name -->
$ git config --global user.name "SyedMuhammadArsalanShah"
<!-- to config user email -->
$ git config --global user.email "smarsalanshah125@gmail.com"
<!-- to config list -->
$ git config --list

<!-- ***|Status|*** -->
<!--Status 4 types ke hoty hain
    1=> untracked  => New File
    2=> modified   => Changed
    3=> staged     => ready for commit means (add )
    4=> unmodified => Unchanged Committed-->
<!-- to check status -->
$ git status
<!-- to check status (shorthand) -->
$ git status -s

<!-- ***|Initialization|*** -->
$ git init

<!-- ***|Add Files|*** -->
<!-- to add a single file  -->
$ git add "fileNameHere"
<!-- to adding all files -->
$ git add .
$ git add -A

<!-- ***|Commit Files|*** -->
<!-- commit by vm-->
$ git commit
<!-- shorthand commit -->
$ git commit -m "message here"
<!-- shorthand add and commit -->
$ git commit -a -m "message here"

<!-- ***|Remote|*** -->
<!-- to add a remote repo -->
$ git remote add origin "New-git-URL-Here"
<!-- to check the remote repo -->
$ git remote -v
<!-- for changing the URI (URL) for a remote Git repository -->
$ git remote set-url origin "New-git-URL-Here"

<!-- ***|Push|*** -->
<!-- for push in the branch -->
$ git push origin "BranchNameHere"
<!-- for push shorthand only for the first time use this  -->
$ git push -u origin "BranchNameHere"
<!-- then always use this  -->
$ git push

<!-- ***|Clone|*** -->
<!-- For Clone And Download -->
$ git clone "git-URL-Here"
<!-- For Clone a specific branch  -->
$ git clone -b "br" "git-URL-Here"

<!-- ***|Checkout Files|*** -->
<!-- to recover/discard a single file from the last commit  -->
$ git checkout "fileNameHere"
<!-- to recover/discard all files from the last commit -->
$ git checkout -f
```

Git 02 Secondary Cheat Sheet

```
<!-- ***|Log|*** -->
<!-- to view all commits  -->
$ git log
<!-- to view a specific log  -->
$ git log -p -1

<!-- ***|diff|*** -->
<!-- to compare the working dir to the stage area -->
$ git diff
<!-- to compare the stage area to the working tree (commit)-->
$ git diff --staged

<!-- ***|remove file|*** -->
<!-- to remove the working dir and stage area -->
$ git rm "fileNameHere"
<!-- to compare the stage area to the working tree (commit)-->
$ git rm --cached "fileNameHere"

<!-- ***|Branch|*** -->
<!-- to check all branches -->
$ git branch
<!-- to rename branch name -->
$ git branch -M "BranchNameHere"
<!-- to navigate to a branch  -->
$ git checkout "BranchNameHere"
<!-- to create a new branch   -->
$ git checkout -b "NewBranchNameHere"
<!-- to delete a branch -->
$ git branch -d "BranchNameHere"

<!-- ***|Undo Changes|*** -->
<!-- to reset from stage  -->
$ git reset "fileNameHere"
<!-- to reset all from the stage -->
$ git reset
<!-- to reset the last commit -->
$ git reset HEAD~1
<!-- to reset a specific commit  -->
$ git reset "CommitHash"
<!-- to reset a specific commit with stage and change in the editor also -->
$ git reset --hard "CommitHash"

<!-- ***|git merge|*** -->
<!--
    1st way by github repo
    create pull request for the base branch merge
    then pull into the local repo
-->
<!-- pull command  -->
$ git pull origin "BranchNameHere"
<!-- 2nd way by merge cmd -->
$ git merge "BranchNameHere"

<!-- ***|gift command alias ♡☺♡|*** -->
$ git config --global alias.YourAliasName "add ."

<!-- ***|git extras  command|*** -->
<!-- to create a new file  -->
$ touch "fileNameHere"
<!-- to view the complete URL of the dir -->
$ pwd
<!-- to view all files in your working dir   -->
$ ls
$ ls -lart
```