Github Notes

Commonly faced problems when we didn't use Git

- Hardisk failure & computer crash
- Hard for multiple people to work on same code
- Difficult to track changes revert back to previous version

- git clone https://github.com/milindzuge01/test.git
- check git status --> git status
- To push in stagging area --> git add hungry.py

git add hungry.py will add hungry.py file to staging area. Later on staging area will be used to commit files when you run git commit

- Commit the changes locally --> git commit -m 'first version of hungry code'

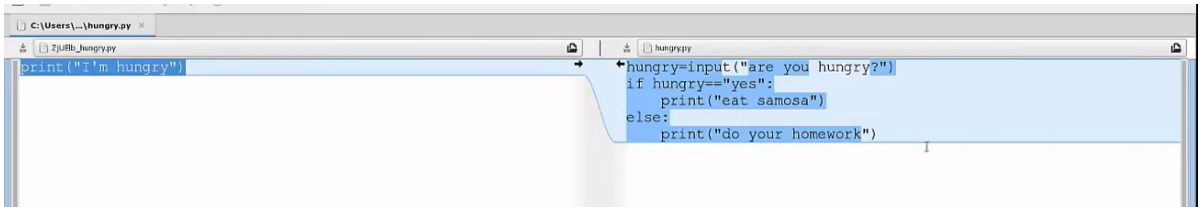
git commit adds your changes to local version control database. You have still not pushed these changes to remote server (github in this case)

- To check log of your code --> git log
- To push code on main github server --> git push

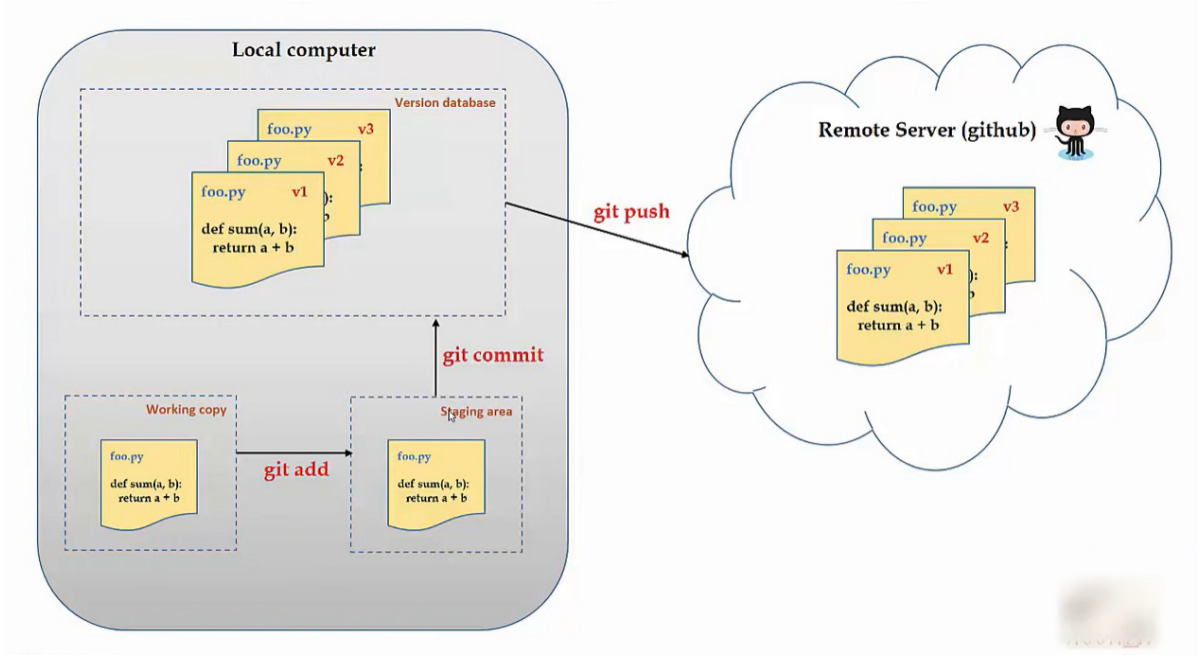
git push will now push commited changes to remote server (github in this case)

- To see the changes between current code & previous code --> git difftool HEAD

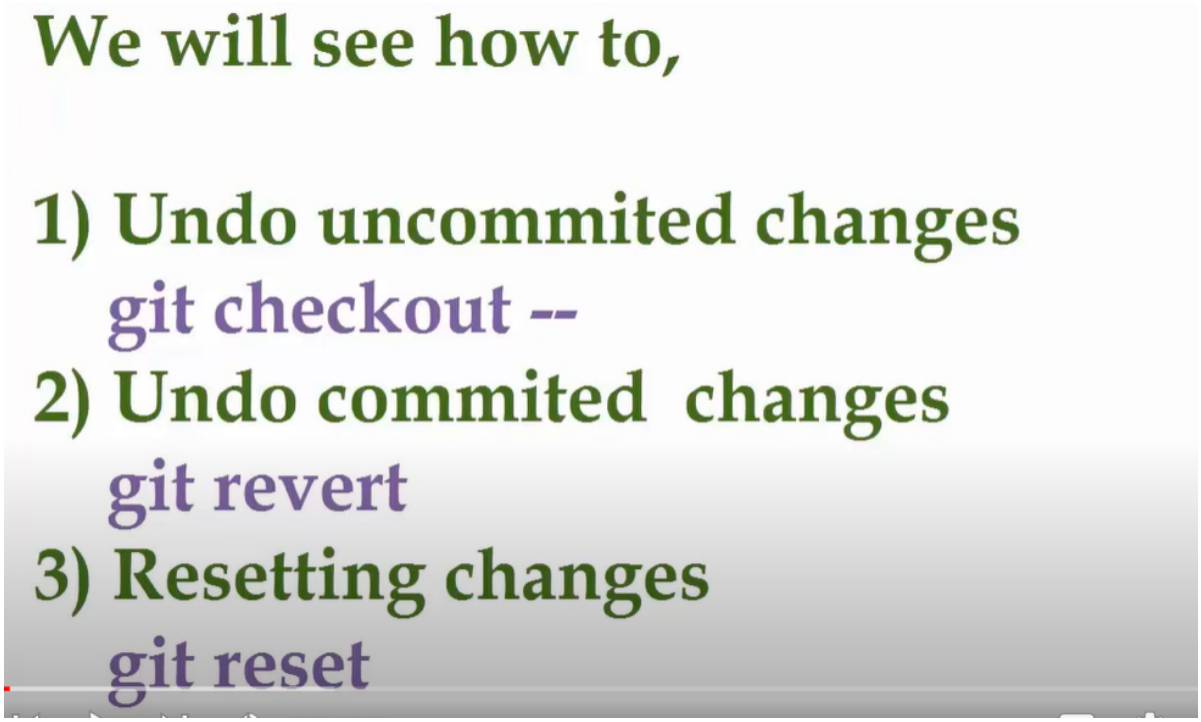
git difftool shows difference between your local changes and previous version of the file



- step 1: git add hungry.py
- step 2: git commit -m 'first version of hungry code'
- STEP 3: git push



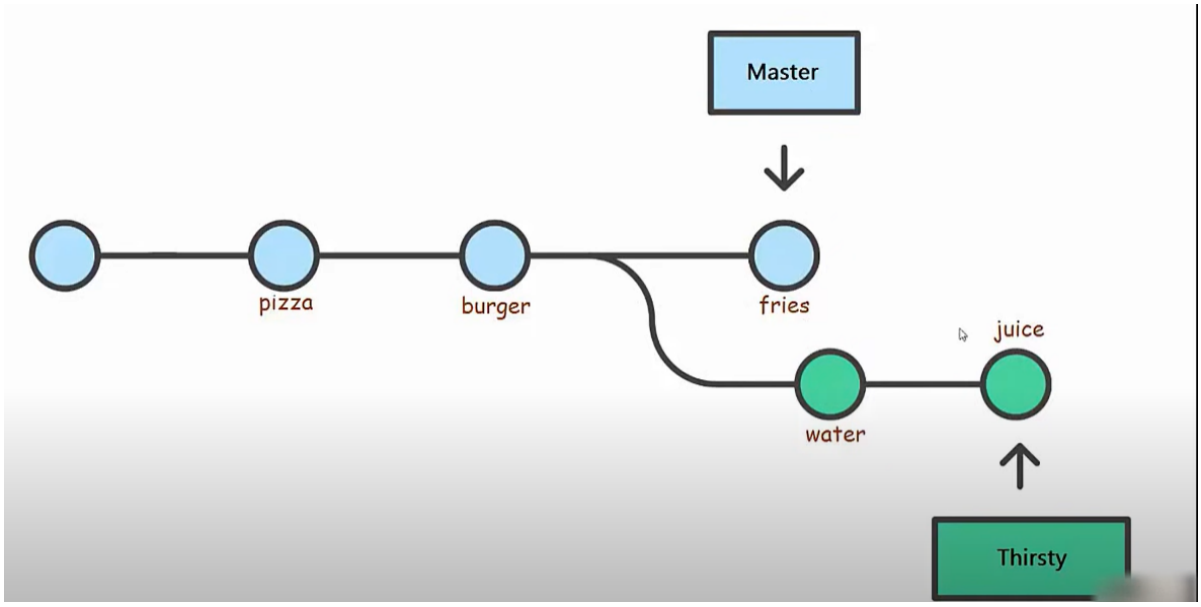
Git Tutorial 5: Undoing/Reverting/Resetting code changes

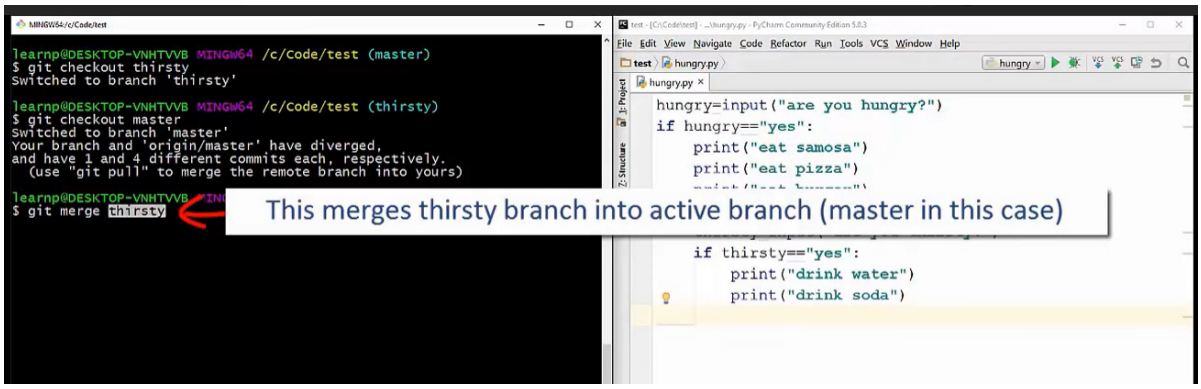


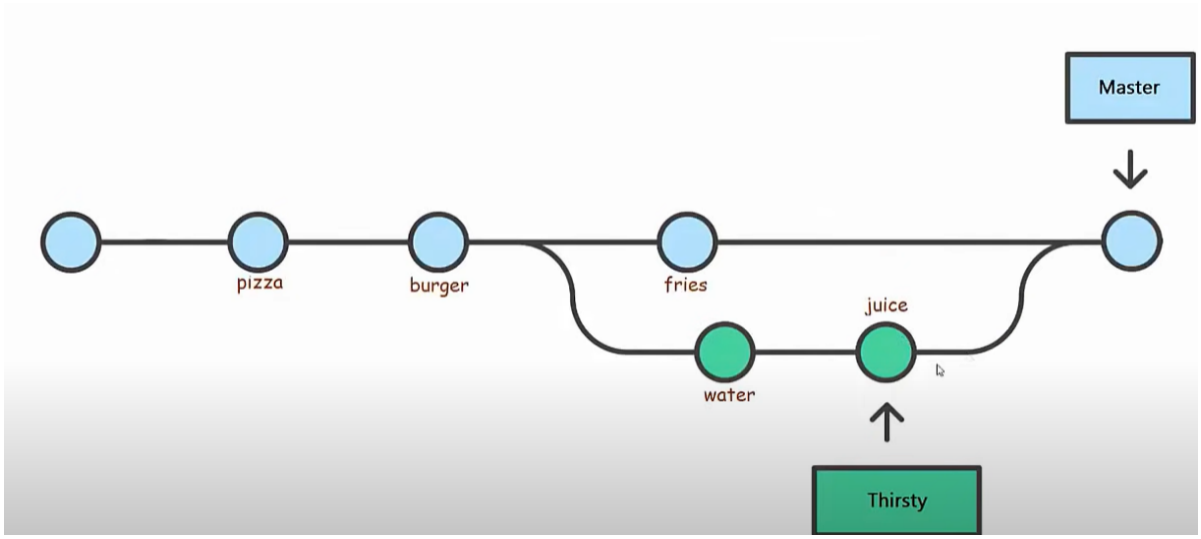
- undo uncommited changes in working directory --> git restore file_name or git checkout -- file_name
  - undo uncommit changes for all file --> git checkout -- .
- undo commited changes in staging directory --> git revert commit_id
  - explicitly commit revert changes in the staging dirctory --> git revert -n commit_id
    - to explicitly commit after reverting --> git commit -m 'revert change name'

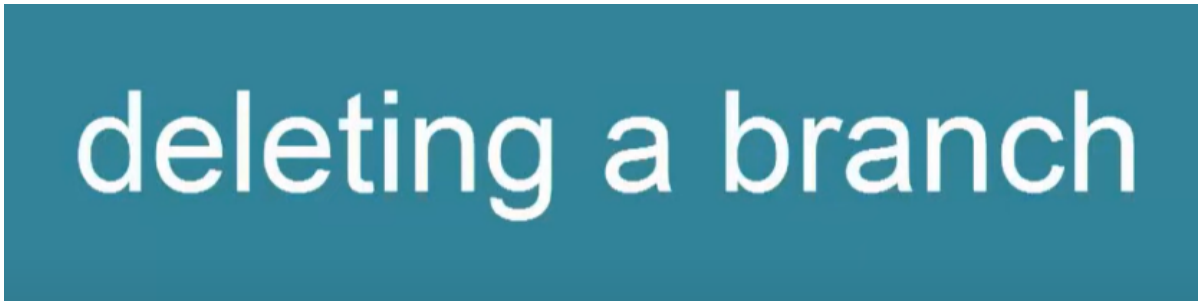
Git Tutorial 6: Branches (Create, Merge, Delete a branch)

- shows branches in a prompt --> git branch
- star denotes the active branch
- create new branch --> git branch thirsty
- change the branch to using --> git checkout branch_name







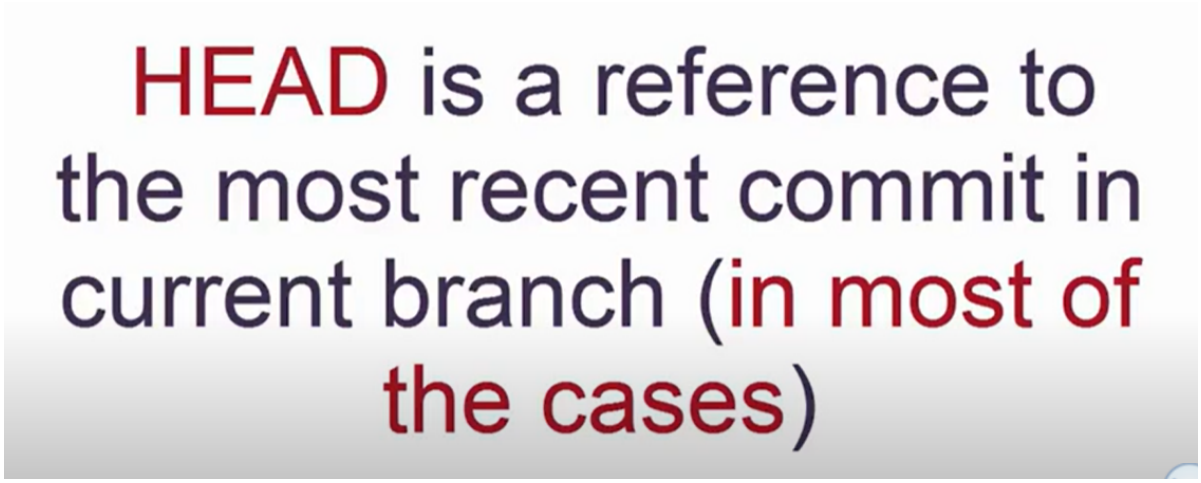


- creating new branch --> git checkout -b dummy

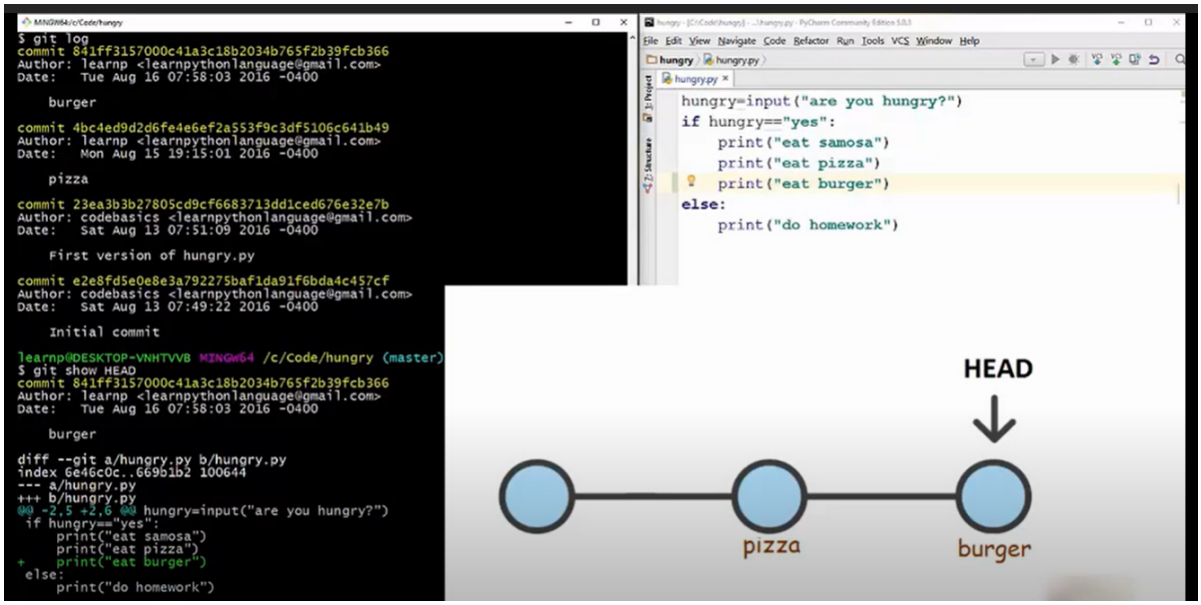
```
learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (thirsty)
$ git checkout -b dummy
Switched to a new branch 'dummy'

learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (dummy)
$ git branch
* dummy
  master
  thirsty
```

- delete a branch --> git branch -d dummy

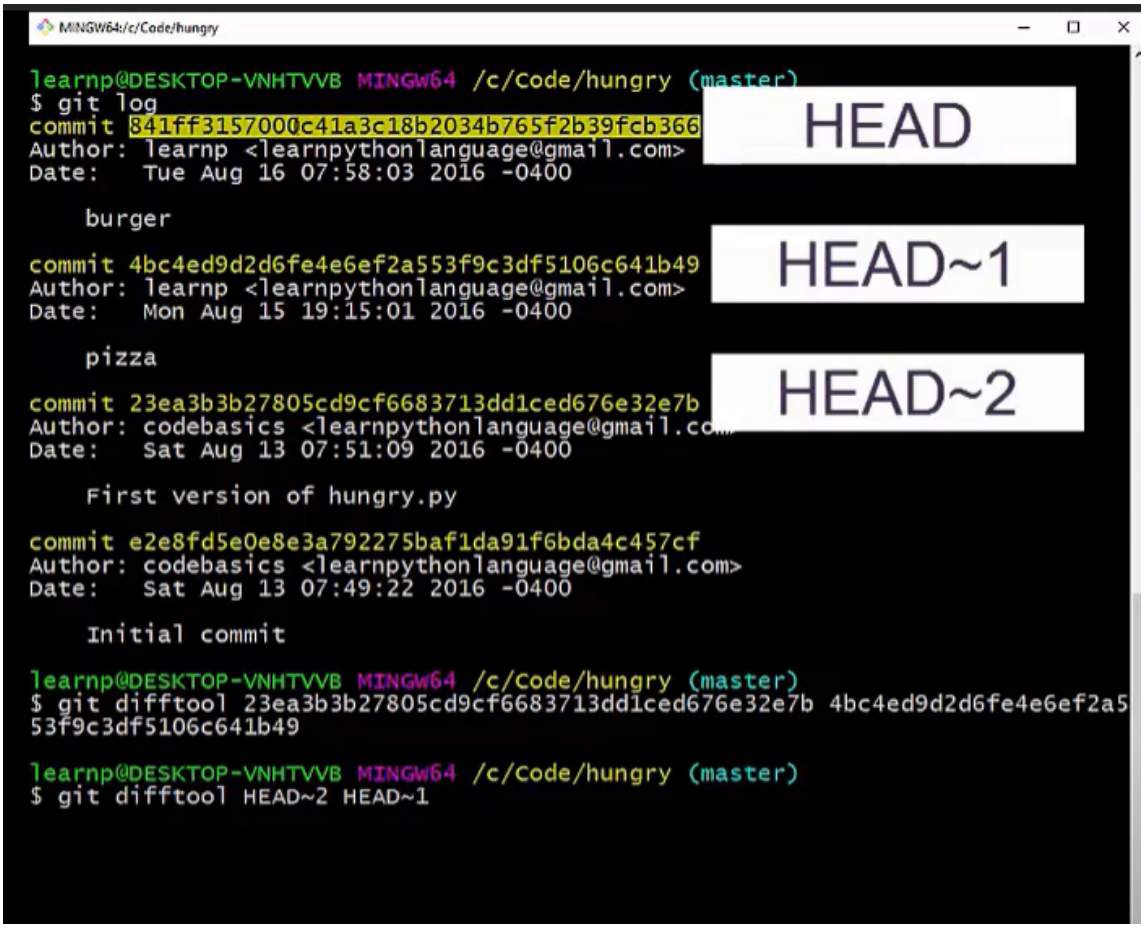
Git Tutorial 7: What is HEAD?



- to see most recent commit --> git show HEAD



- to check changes between any two commit --> git difftool old_commit_id new_commit_id
- above also achieved by using below :

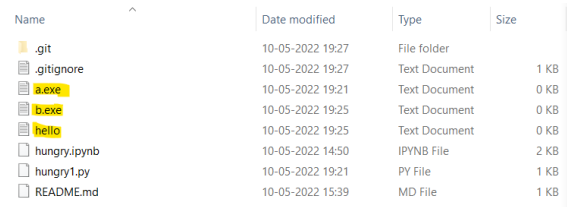


- move head to any commit --> git checkout commit_id

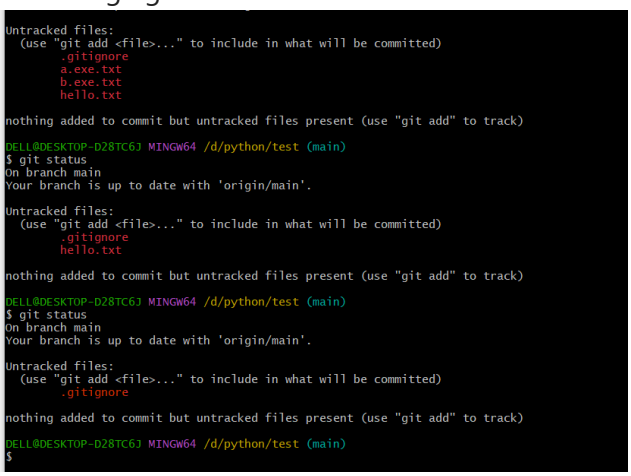
Git Tutorial 8 - .gitignore file

- create file in main local directory using --> touch .gitignore
- add file name which we want to hide from git status in it & saved it.

here hided yellow highlighted files & it all included in .gitignore file



Name	Date modified	Type	Size
.git	10-05-2022 19:27	File folder	
.gitignore	10-05-2022 19:27	Text Document	1 KB
a.exe	10-05-2022 19:21	Text Document	0 KB
b.exe	10-05-2022 19:25	Text Document	0 KB
hello	10-05-2022 19:25	Text Document	0 KB
hungry.ipynb	10-05-2022 14:50	IPYNB File	2 KB
hungry1.py	10-05-2022 19:21	PY File	1 KB
README.md	10-05-2022 15:39	MD File	1 KB



```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        a.exe.txt
        b.exe.txt
        hello.txt

nothing added to commit but untracked files present (use "git add" to track)

DELL@DESKTOP-D28TC6J MINGW64 /d/python/test (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        hello.txt

nothing added to commit but untracked files present (use "git add" to track)

DELL@DESKTOP-D28TC6J MINGW64 /d/python/test (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)

DELL@DESKTOP-D28TC6J MINGW64 /d/python/test (main)
$
```

Git Tutorial 9: Diff and Merge using meld

- install meld --> https://meldmerge.org/
- configure .gitconfig file -->

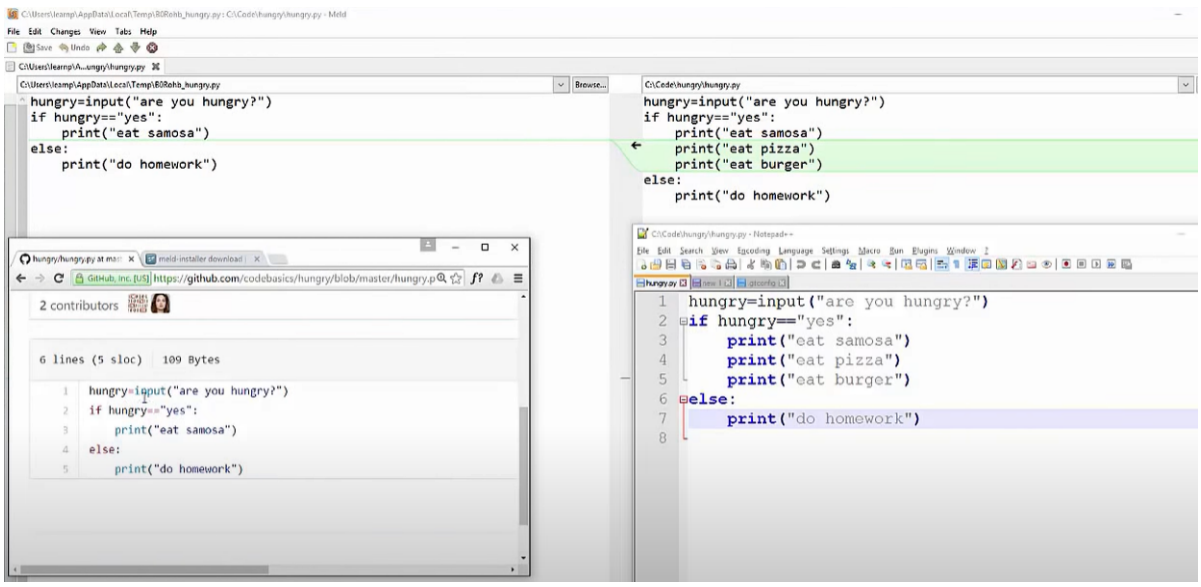
[diff] tool = meld [difftool "meld"] path = C:/Program Files (x86)/Meld/Meld.exe [difftool] prompt = false

*.gitconfig - Notepad

File Edit Format View Help

```
[user]
        email = milindzuge01@gmail.com
        name = Milind

[diff]
        tool = meld
[difftool "meld"]
        path = C:/Program Files (x86)/Meld/Meld.exe
[difftool]
        prompt = false
```

- command to see differences between saved local code & git repository --> git difftool origin/main



- lanch specific tool from multiple diff tool using -->
  - git difftool -t meld
  - git difftool -t vimdiff

when git repository's & local code is out of sink then we have a merge conflict using --> git pull

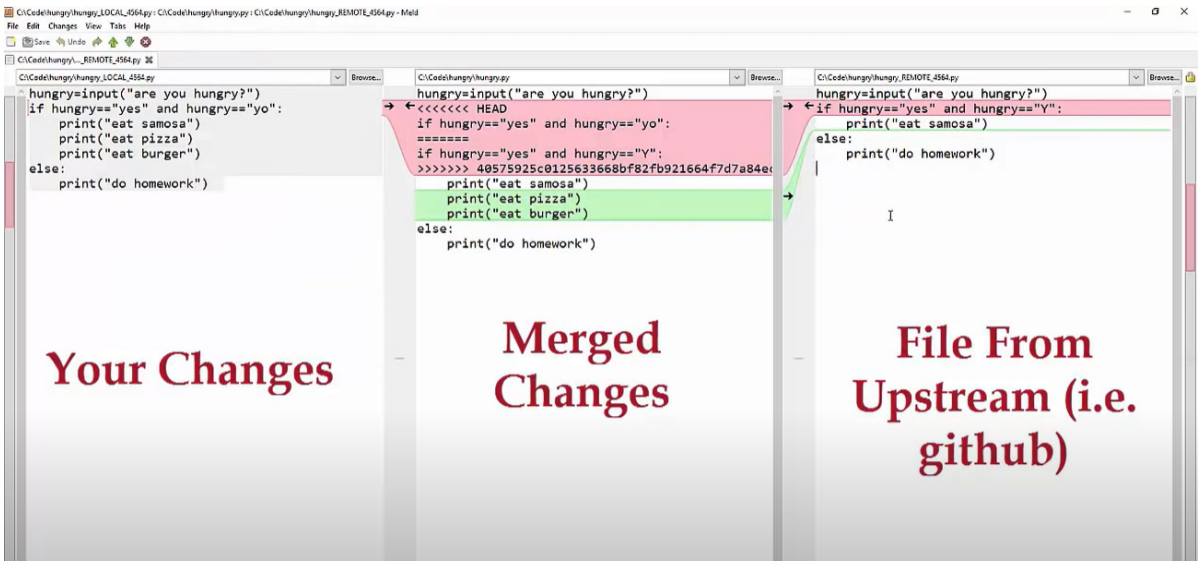
```
DELL@DESKTOP-D28TC6J MINGW64 /d/python/test (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 676 bytes | 56.00 KiB/s, done.
From https://github.com/milindzuge01/test
   01c77e1..b34f075  main       -> origin/main
Auto-merging hungry2.py
CONFLICT (content): Merge conflict in hungry2.py
Automatic merge failed; fix conflicts and then commit the result.
```

- configure .gitconfig file -->

[merge] tool = meld [mergetool "meld"] path = C:/Program Files (x86)/Meld/Meld.exe [mergetool] keepBackup = false

```
[merge]
        tool = meld
[mergetool "meld"]
        path = C:/Program Files (x86)/Meld/Meld.exe
[mergetool]
        keepBackup = false
```

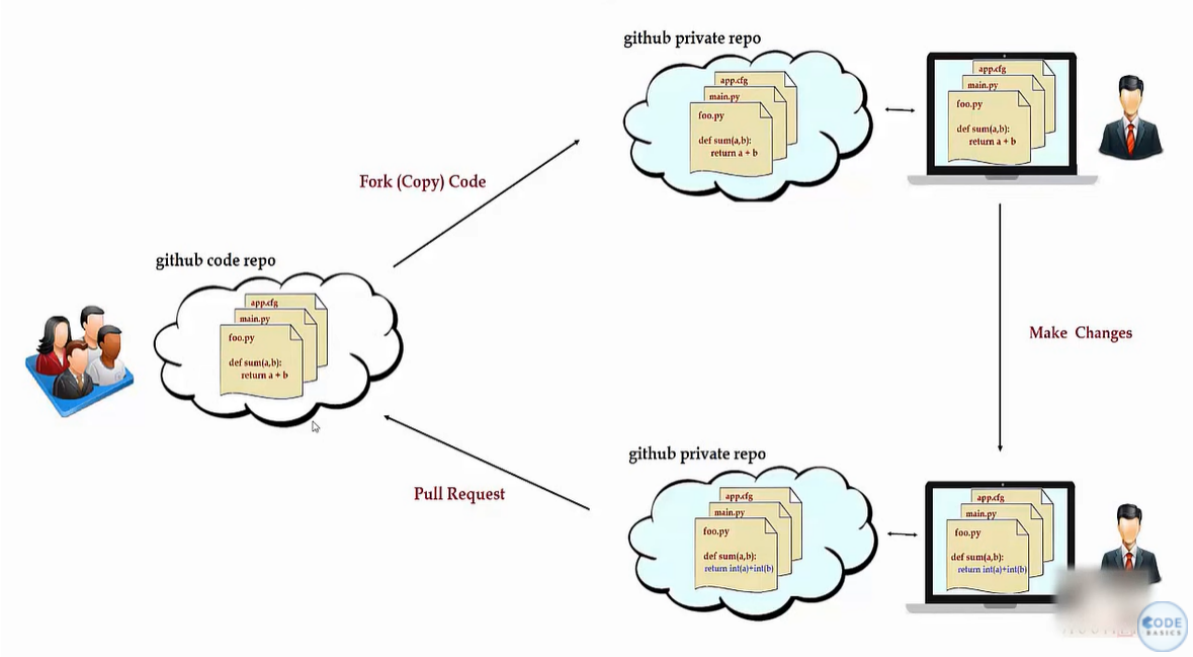
- command to see differences between saved local code,git repository with merging --> git mergetool



Git Github Tutorial 10: What is Pull Request?

What is Pull Request

- On github, owner can share his code with others
- Other person can make code changes and send a request to owner to pull/merge his code changes into owner's repository



In [ ]: