



FEBRUARY 17, 2023

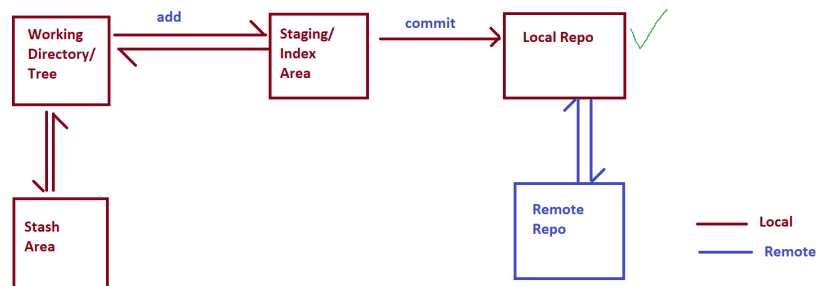
DevOps Classroomnotes 17/Feb/2023

Git Contd

- What Git Commit is?



- Work done so far



- We have create a new folder and done the activites as shown below

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git add .
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   src/main.py
    new file:   test/main.py

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git commit -m "1: create basic structure"
[master (root-commit) 3e331ba] 1: create basic structure
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 src/main.py
create mode 100644 test/main.py
```

- Now lets check for status

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git log
commit 3e331bac7279a0583a31b046328acb43691bc2c5 (HEAD -> master)
Author: shaikkhajaibrahim <qtkhajacloud@gmail.com>
Date:   Fri Feb 17 08:35:33 2023 +0530

1: create basic structure
```

- Lets make 3 changes (2 in existing files in commit and create a new file)

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   src/main.py
        modified:   test/main.py

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

no changes added to commit (use "git add" and/or "git commit -a")
```

- In Git Untracked files are the files that were not part of local repo and a new file is created.

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git add .

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Readme.md
        modified:   src/main.py
        modified:   test/main.py
```

- Now commit the changes

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git commit -m "2. Added documentation"
[master 052343c] 2. Added documentation
3 files changed, 2 insertions(+)
create mode 100644 Readme.md

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git log
commit 052343c9904700273eaa0163a3c993a89571034 (HEAD -> master)
Author: shaikkhajaibrahim <qtkhajacloud@gmail.com>
Date:   Fri Feb 17 08:42:34 2023 +0530

2. Added documentation

commit 3e331bac7279a0583a31b046328acb43691bc2c5
Author: shaikkhajaibrahim <qtkhajacloud@gmail.com>
Date:   Fri Feb 17 08:35:33 2023 +0530

1: create basic structure
```

- HEAD is a pointer which generally points to latest commit, Working tree will populate the contents based on HEAD which refers to commit

- Add some changes to staging area

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        modified:   src/main.py
        modified:   test/main.py

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git restore --staged README.md

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/main.py
        modified:   test/main.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
```

- To remove the changes from working tree for a tracked file

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git restore README.md

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/main.py
        modified:   test/main.py

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$
```

- Lets see how to remove untracked which is in staging area

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/main.py
        modified:   test/main.py

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

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git add .

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/bymistake.py
        modified:   src/main.py
        modified:   test/main.py

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$
```

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git restore --staged src/bymistake.py

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/main.py
        modified:   test/main.py

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

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git clean -fd src/
Removing src/bymistake.py

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$
```

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/main.py
        modified:   test/main.py
```

- Now lets create a 3 commit

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git commit -m "3. Added code"
[master f30671f] 3. Added code
 2 files changed, 8 insertions(+), 2 deletions(-)

Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git log
commit f30671fc0635c4451f0f4805a7677d1558aa3aa3 (HEAD -> master)
Author: shaikkhajaibrahim <qtkhajacloud@gmail.com>
Date:   Fri Feb 17 08:58:40 2023 +0530

    3. Added code

commit 052343c9904700273eaf0163a3c993a89571034
Author: shaikkhajaibrahim <qtkhajacloud@gmail.com>
Date:   Fri Feb 17 08:42:34 2023 +0530

    2. Added documentation

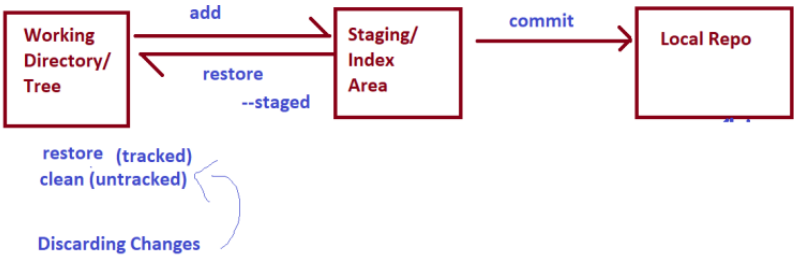
commit 3e331bac7279a0583a31b046328acb43691bc2c5
Author: shaikkhajaibrahim <qtkhajacloud@gmail.com>
```

- git log in short form git log --oneline

```
Dell@DESKTOP-TM7SH71 MINGW64 /c/temp/GitPractice/helloGit (master)
$ git log --oneline
f30671f (HEAD -> master) 3. Added code
052343c 2. Added documentation
3e331ba 1: create basic structure
```

- In Git by default there will be one branch which is called as master which looks at latest commit id.
- Checkout commands moves the HEAD position. By default as master points to latest commit, HEAD points to master. IF you want navigate back in history git checkout <commit-id> to get back to latest commit git checkout master

- Note: checkout has other flavors which we are yet to know.



Leave a Reply

Enter your comment here...

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)



About continuous learner

devops & cloud enthusiastic learner

[VIEW ALL POSTS](#)

◀ PREVIOUS POST

[Azure Classroomnotes 17/Feb/2023](#)

NEXT POST

[AWS Classroomnotes 17/Feb/2023](#)