

Git Setup

1. It was done using Git for mac Installer.
2. The installation verification was done by `git --version` command on the prompt.
3. Then after the installation the username and email for authentication was set using the following commands
 - a. `$ git config --global user.name "Name"`
 - b. `$ git config --global user.email "name@domain.com"`

```
[TTNs-MacBook-Pro:~ ttn$ git config --global user.name "Rahul-iOS-ttn" ]
[TTNs-MacBook-Pro:~ ttn$ git config --global user.email "rahul.sharma1@tothenew.com" ]
TTNs-MacBook-Pro:~ ttn$ █
```

Initialize a Git repository

This can be done by the command `git init` in the correct path where one wants to begin a new project.

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git init ]
Initialized empty Git repository in /Volumes/Macintosh SSD - Data/exercise ss/data/Assignment_work/.git/
TTNs-MacBook-Pro:Assignment_work ttn$ █
```

Add files to the repository(Local)

This can be done by creating new files and using `git add` command on them.

```
TTNs-MacBook-Pro:Assignment_work ttn$ touch file.txt
TTNs-MacBook-Pro:Assignment_work ttn$ touch one.txt
TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch master
```

No commits yet

Untracked files:

(use "`git add <file>...`" to include in what will be committed)

```
file.txt
one.txt
```

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

```
TTNs-MacBook-Pro:Assignment_work ttn$ git add .
```

```
TTNs-MacBook-Pro:Assignment_work ttn$ git add .
TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch master
```

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)

```
new file:   file.txt
new file:   one.txt
```

```
TTNs-MacBook-Pro:Assignment_work ttn$
```

Unstage one file from the Staging area.

We can do this in 2 ways

1. By deleting the files and then again running the command `git add .` to again stage everything.

```
TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch master
```

No commits yet

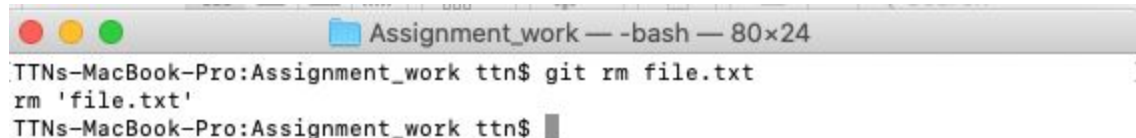
Changes to be committed:
(use "git rm --cached <file>..." to unstage)

```
new file:   file.txt
new file:   one.txt
new file:   three.txt
new file:   two.txt
```

Changes not staged for commit:
(use "git add/rm <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)

```
deleted:    file.txt
```

```
TTNs-MacBook-Pro:Assignment_work ttn$
```

A screenshot of a macOS terminal window titled "Assignment_work — -bash — 80x24". The terminal shows the command `git rm file.txt` being executed, followed by the output `rm 'file.txt'`. The prompt then shows `TTNs-MacBook-Pro:Assignment_work ttn$`.

```
TTNs-MacBook-Pro:Assignment_work ttn$ git rm file.txt
rm 'file.txt'
TTNs-MacBook-Pro:Assignment_work ttn$
```

2. By using `git rm` command to remove the file from unstage area, this inturn will remove the file completely from the staging area and local repository alike.

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   one.txt
new file:   three.txt
new file:   two.txt
```

```
TTNs-MacBook-Pro:Assignment_work ttn$ git rm -f one.txt
rm 'one.txt'
```

```
TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   three.txt
new file:   two.txt
```

```
TTNs-MacBook-Pro:Assignment_work ttn$ █
```

3. If we don't want to track a file then we can simply use `git rm --cached <file name>`

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   three.txt
new file:   two.txt
```

```
TTNs-MacBook-Pro:Assignment_work ttn$ git rm --cached two.txt
rm 'two.txt'
```

```
TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   three.txt
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
two.txt
```

```
TTNs-MacBook-Pro:Assignment_work ttn$ █
```

Commit the files

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "initial commit"
[master (root-commit) 378a491] initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 three.txt
TTNs-MacBook-Pro:Assignment_work ttn$
```

Add a Remote repository

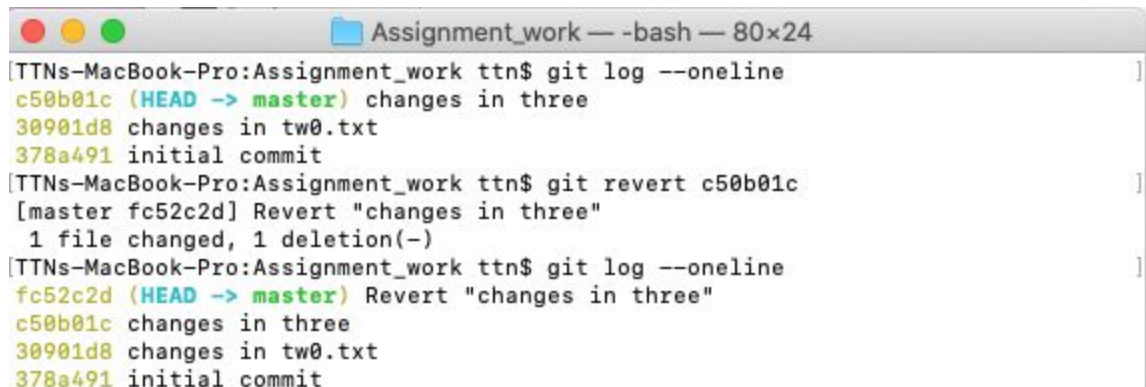
We do this by first creating an empty repository on GitHub and then adding a remote connection to our local repository.

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git remote add origin https://github.com/Rahul-iOS-ttn/Repository_for_assignment.git
TTNs-MacBook-Pro:Assignment_work ttn$ git remote -v
origin https://github.com/Rahul-iOS-ttn/Repository_for_assignment.git (fetch)
origin https://github.com/Rahul-iOS-ttn/Repository_for_assignment.git (push)
TTNs-MacBook-Pro:Assignment_work ttn$
```

Undo changes to a particular file

We can do this by using 2 commands

1. Git revert - in this the change gets reverted back by only one commit and it creates a new commit saying that it reverted the previous commit. If we say it in a tree term it is something like, 1->2->3->2.



```
Assignment_work — -bash — 80x24
[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
c50b01c (HEAD -> master) changes in three
30901d8 changes in tw0.txt
378a491 initial commit
[TTNs-MacBook-Pro:Assignment_work ttn$ git revert c50b01c
[master fc52c2d] Revert "changes in three"
 1 file changed, 1 deletion(-)
[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
fc52c2d (HEAD -> master) Revert "changes in three"
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
```

- a. With this you also have to type a message afterwards it is something similar to the commit message that we usually type.

```
Revert "changes in three"
```

```
This reverts commit c50b01c83d8a386f3e512729354631c154fabfa2.
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#       modified:   three.txt
#
```

- b. We can revert a revert change

```
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
fc52c2d (HEAD -> master) Revert "changes in three"
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ git revert fc52c2d ]
[master ff4bf6f] Revert "Revert "changes in three"" Time paradox
1 file changed, 1 insertion(+)
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
ff4bf6f (HEAD -> master) Revert "Revert "changes in three"" Time paradox
fc52c2d Revert "changes in three"
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
```

2. Git Reset - This is a more permanent change in which it just reverts back to the repository state that a user specifies and you can not undo this change, so once this is done it can not be changed, for instance - 1->2->3->4, so after reset to 2 the tree will be 1->2.

```
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
ff4bf6f (HEAD -> master) Revert "Revert "changes in three"" Time paradox
fc52c2d Revert "changes in three"
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ git reset --hard c50b01c ]
HEAD is now at c50b01c changes in three
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
c50b01c (HEAD -> master) changes in three
30901d8 changes in tw0.txt
378a491 initial commit
```

- a. This also has types like soft, mixed and hard. The above mentioned tree state is for git hard reset.
- b. Soft reset is more like a checkout command in which we can move our pointer to a desired commit and then work our way through it. Soft commit just removes the commit without changing the environment which means that the files will still be staged just the commit will be changed.


```

TTNs-MacBook-Pro:Assignment_work ttn$ touch one.txt
TTNs-MacBook-Pro:Assignment_work ttn$ touch four.txt
TTNs-MacBook-Pro:Assignment_work ttn$ git add .
TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "new files added"
[master 09bf5a3] new files added
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 four.txt
 create mode 100644 one.txt
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
09bf5a3 (HEAD -> master) new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ git reset --soft

TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
09bf5a3 (HEAD -> master) new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ git reset --soft c50b01c
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
c50b01c (HEAD -> master) changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   four.txt
        new file:   one.txt

TTNs-MacBook-Pro:Assignment_work ttn$

```

- c. Mixed reset is somewhere in between the hard and soft, it is also the default flag option for reset in git. Now if we do a git reset soft then we are just going back in the previous state while without deleting or removing any files so it's like removing a commit from a remote repository without destroying the changes in the local repository. To state a difference between soft reset and mixed, it will unstage the files.

```

TTNs-MacBook-Pro:Assignment_work ttn$ touch one.txt
TTNs-MacBook-Pro:Assignment_work ttn$ touch four.txt
TTNs-MacBook-Pro:Assignment_work ttn$ git add .
TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "new files added"
[master 09bf5a3] new files added
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 four.txt
 create mode 100644 one.txt
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
09bf5a3 (HEAD -> master) new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ git reset --soft

```

Push changes to Github

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git push origin master ]
Username for 'https://github.com': Rahul-iOS-ttn
[Password for 'https://Rahul-iOS-ttn@github.com': ]
Counting objects: 24, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (20/20), done.
Writing objects: 100% (24/24), 2.26 KiB | 771.00 KiB/s, done.
Total 24 (delta 7), reused 0 (delta 0)
remote: Resolving deltas: 100% (7/7), done.
To https://github.com/Rahul-iOS-ttn/Repository_for_assignment.git
 * [new branch]      master -> master
[TTNs-MacBook-Pro:Assignment_work ttn$ git push origin dev ]
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:      https://github.com/Rahul-iOS-ttn/Repository_for_assignment/pull/new
/dev
remote:
To https://github.com/Rahul-iOS-ttn/Repository_for_assignment.git
 * [new branch]      dev -> dev
TTNs-MacBook-Pro:Assignment_work ttn$ █
```

Clone the repository

For this we first copy the link of the repository from GitHub and the clone using git clone

```
[TTNs-MacBook-Pro:day2 ttn$ git clone https://github.com/Rahul-iOS-ttn/Repository_for_assignment.git]
Cloning into 'Repository_for_assignment'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 24 (delta 7), reused 24 (delta 7), pack-reused 0
Unpacking objects: 100% (24/24), done.
TTNs-MacBook-Pro:day2 ttn$ █
```

Add changes to one of the copies and pull the changes in the other.

```
[TTNs-MacBook-Pro:Assignment_work ttn$ pwd
/volumes/Macintosh SSD - Data/exercise_ss/day2/Assignment_work
[TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   dev1.txt
        modified:   four.txt

no changes added to commit (use "git add" and/or "git commit -a")
[TTNs-MacBook-Pro:Assignment_work ttn$ git add .
[TTNs-MacBook-Pro:Assignment_work ttn$ git commit -a -m "changed from intial dir"

[master dec3b81] changed from intial dir
 2 files changed, 2 insertions(+)
TTNs-MacBook-Pro:Assignment_work ttn$ █

[TTNs-MacBook-Pro:Assignment_work ttn$ git push origin master
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 366 bytes | 366.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Rahul-iOS-ttn/Repostiory_for_assingment.git
   ada2d67..dec3b81  master -> master
TTNs-MacBook-Pro:Assignment_work ttn$ █

TTNs-MacBook-Pro:day2 ttn$ cd Repostiory_for_assingment/
TTNs-MacBook-Pro:Repostiory_for_assingment ttn$ pwd
/volumes/Macintosh SSD - Data/exercise_ss/day2/Repostiory_for_assingment
TTNs-MacBook-Pro:Repostiory_for_assingment ttn$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 4 (delta 1), reused 4 (delta 1), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/Rahul-iOS-ttn/Repostiory_for_assingment
   ada2d67..dec3b81  master    -> origin/master
Updating ada2d67..dec3b81
Fast-forward
 dev1.txt | 1 +
 four.txt | 1 +
 2 files changed, 2 insertions(+)
TTNs-MacBook-Pro:Repostiory_for_assingment ttn$ █
```


Check differences between a file and its staged version

```
TTNs-MacBook-Pro:Assignment_work ttn$ git status ]
On branch master
nothing to commit, working tree clean
TTNs-MacBook-Pro:Assignment_work ttn$ git status ]
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   dev1.txt

no changes added to commit (use "git add" and/or "git commit -a")
TTNs-MacBook-Pro:Assignment_work ttn$ git diff dev1.txt ]
diff --git a/dev1.txt b/dev1.txt
index e69de29..6249094 100644
--- a/dev1.txt
+++ b/dev1.txt
@@ -0,0 +1 @@
+Changes in dev1
```

Ignore a few files to be checked in

This part can be done by creating a gitignore file.

So here .gitignore is a file which we add into our project as an index from where git can inquire about the files that it needs to be ignored. Those files can be static files or log files etc.

NOTE - Sometime when you add a gitignore in the middle of the project, the git will still track the files this is because the git remembers those files meaning they are still in the git cache which can be fixed by using { git rm -r --cached }

```
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
b0521cb (HEAD -> master) new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ touch .gitignore ]
TTNs-MacBook-Pro:Assignment_work ttn$ mkdir Autogen ]
TTNs-MacBook-Pro:Assignment_work ttn$ cd Autogen/ ]
TTNs-MacBook-Pro:Autogen ttn$ touch a.txt b.txt c.txt ]
TTNs-MacBook-Pro:Autogen ttn$ cd .. ]
TTNs-MacBook-Pro:Assignment_work ttn$ git status ]
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore
        Autogen/
```

```
#adding the Autogen folder in for ignoring it.  
Autogen/*
```

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git status ]  
On branch master  
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)  
[TTNs-MacBook-Pro:Assignment_work ttn$ git add . ]  
[TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "added gitignore after cache settings"]  
[master 7a8465e] added gitignore after cache settings  
 1 file changed, 2 insertions(+)  
 create mode 100644 .gitignore  
[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]  
7a8465e (HEAD -> master) added gitignore after cache settings  
b0521cb new files added  
c50b01c changes in three  
30901d8 changes in two.txt  
378a491 initial commit
```

Create a new branch.

We can do this by creating a new branch in the project. We can do this in two ways

1. We can create a new branch in the repo using `git branch <branch name>` and then jump to that new branch using `checkout`.

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git branch dev  
[TTNs-MacBook-Pro:Assignment_work ttn$ git checkout dev  
Switched to branch 'dev'  
[TTNs-MacBook-Pro:Assignment_work ttn$ git branch -a  
* dev  
  master  
TTNs-MacBook-Pro:Assignment_work ttn$ ]
```

2. A much easier way to do this is by just using `checkout` with `-b` option which creates a new branch and sets it as a working directory altogether.

```
TTNs-MacBook-Pro:Assignment_work ttn$ git checkout -b bugs ]  
Switched to a new branch 'bugs'  
TTNs-MacBook-Pro:Assignment_work ttn$ git branch -a ]  
* bugs  
  dev  
  master  
TTNs-MacBook-Pro:Assignment_work ttn$ git checkout master ]  
Switched to branch 'master'  
TTNs-MacBook-Pro:Assignment_work ttn$ git branch -a ]  
  bugs  
  dev  
* master
```

Diverge them with commits

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
7a8465e (HEAD -> master, dev, bugs) added gitignore after cache settings
b0521cb new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
[TTNs-MacBook-Pro:Assignment_work ttn$ git checkout dev ]
Switched to branch 'dev'
[TTNs-MacBook-Pro:Assignment_work ttn$ touch dev1.txt ]
[TTNs-MacBook-Pro:Assignment_work ttn$ git add . ]
[TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "added file in dev branch" ]
[dev 975e14e] added file in dev branch
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dev1.txt
[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
fatal: unrecognized argument: --oneline
[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
975e14e (HEAD -> dev) added file in dev branch
7a8465e (master, bugs) added gitignore after cache settings
b0521cb new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ █

TTNs-MacBook-Pro:Assignment_work ttn$ touch dev2
TTNs-MacBook-Pro:Assignment_work ttn$ git add .
TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "added dev2"
[dev 8d72f7c] added dev2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dev2
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
8d72f7c (HEAD -> dev) added dev2
975e14e added file in dev branch
7a8465e (master, bugs) added gitignore after cache settings
b0521cb new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ git checkout master
Switched to branch 'master'
TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch master
nothing to commit, working tree clean
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
7a8465e (HEAD -> master, bugs) added gitignore after cache settings
b0521cb new files added
c50b01c changes in three
30901d8 changes in tw0.txt
```

Edit the same file at the same line on both branches and commit

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git checkout master  
Already on 'master'  
TTNs-MacBook-Pro:Assignment_work ttn$
```



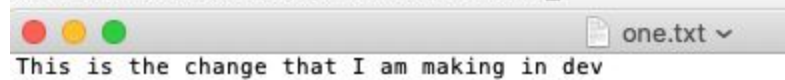
This is the change I am making in master

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git checkout master  
Already on 'master'  
[TTNs-MacBook-Pro:Assignment_work ttn$ git status  
On branch master  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)  
  (use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   one.txt
```

```
no changes added to commit (use "git add" and/or "git commit -a")  
[TTNs-MacBook-Pro:Assignment_work ttn$ git add .  
[TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "one.txt changed in master"  
[master 114a42a] one.txt changed in master  
  1 file changed, 1 insertion(+)  
[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline  
114a42a (HEAD -> master) one.txt changed in master  
7a8465e (bugs) added gitignore after cache settings  
b0521cb new files added  
c50b01c changes in three  
30901d8 changes in tw0.txt  
378a491 initial commit
```

```
TTNs-MacBook-Pro:Assignment_work ttn$ git checkout dev
Switched to branch 'dev'
TTNs-MacBook-Pro:Assignment_work ttn$
```



```
Switched to branch 'dev'
TTNs-MacBook-Pro:Assignment_work ttn$ git status
On branch dev
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   one.txt
```

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

```
TTNs-MacBook-Pro:Assignment_work ttn$ git add .
TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "changes on one.txt in dev"
[dev 048ae2a] changes on one.txt in dev
 1 file changed, 1 insertion(+)
TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline
048ae2a (HEAD -> dev) changes on one.txt in dev
8d72f7c added dev2
975e14e added file in dev branch
7a8465e (bugs) added gitignore after cache settings
b0521cb new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$
```


Try merging and resolve merge conflicts

```
[TTNs-MacBook-Pro:Assignment_work ttn$ git checkout master ]
Switched to branch 'master'
[TTNs-MacBook-Pro:Assignment_work ttn$ git status ]
On branch master
nothing to commit, working tree clean
[TTNs-MacBook-Pro:Assignment_work ttn$ git merge dev ]
Auto-merging one.txt
CONFLICT (content): Merge conflict in one.txt
Automatic merge failed; fix conflicts and then commit the result.
TTNs-MacBook-Pro:Assignment_work ttn$ █

[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
3a56517 (HEAD -> master) changes in one
7a8465e (bugs) added gitignore after cache settings
b0521cb new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
[TTNs-MacBook-Pro:Assignment_work ttn$ git reset --mixed 7a8465e ]
Unstaged changes after reset:
M    one.txt
[TTNs-MacBook-Pro:Assignment_work ttn$ git status ]
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   one.txt

no changes added to commit (use "git add" and/or "git commit -a")
[TTNs-MacBook-Pro:Assignment_work ttn$ git add . ]
TTNs-MacBook-Pro:Assignment_work ttn$ git commit -m "cleared merge conflict"█

[TTNs-MacBook-Pro:Assignment_work ttn$ git merge dev ]
Merge made by the 'recursive' strategy.
 dev1.txt | 0
 dev2     | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dev1.txt
 create mode 100644 dev2
[TTNs-MacBook-Pro:Assignment_work ttn$ git log --oneline ]
ada2d67 (HEAD -> master) Merge branch 'dev' This is the merge of dev with master
b96d461 cleared merge conflict
048ae2a (dev) changes on one.txt in dev
8d72f7c added dev2
975e14e added file in dev branch
7a8465e (bugs) added gitignore after cache settings
b0521cb new files added
c50b01c changes in three
30901d8 changes in tw0.txt
378a491 initial commit
TTNs-MacBook-Pro:Assignment_work ttn$ █
```

Stash the changes and pop them