

Mobile Computing

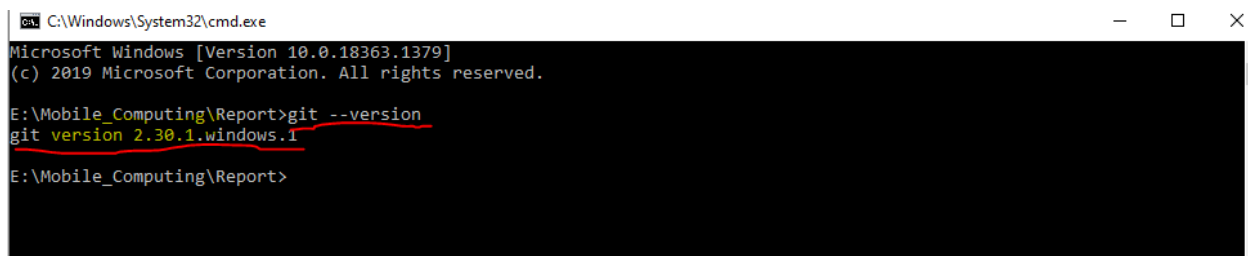
Ahtisham Ali

MCSF19M002

Introduction to Git & GitHub

Lecture 1-3:

Git Installation confirmation:

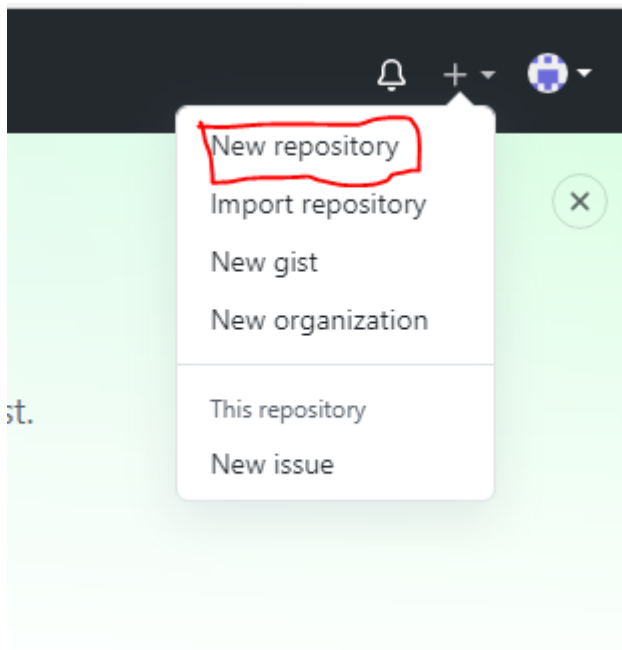


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

E:\Mobile_Computing\Report>git --version
git version 2.30.1.windows.1

E:\Mobile_Computing\Report>
```

Creating a new repository on GitHub




Choosing a suitable name and making it public:

Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere?


[Import a repository.](#)

Owner *  ahtisham735 / Repository name * MC_Reports ✓

Great repository names are short, simple, and easy to remember. MC_Reports is available. Need inspiration? How about [stunning-fortnight?](#)

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Creating clone :

Coping repository link and execute “git clone url” in command prompt

```
Quick setup — if you've done this kind of thing before
Set up in Desktop or HTTPS SSH https://github.com/ahtisham735/MC_Reports.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

E:\Mobile_Computing\Report>git --version
git version 2.30.1.windows.1

E:\Mobile_Computing\Report>git clone https://github.com/ahtisham735/MC_Reports.git
Cloning into 'MC_Reports'...
warning: You appear to have cloned an empty repository.

E:\Mobile_Computing\Report>
```

Clone has been created

For uploading a file on Git from local machine, we need to perform following commands:

1:git add filename

It tells **Git** that you want to include updates to a particular file in the next commit. However, **git add** doesn't really affect the repository in any significant way—changes are not actually recorded until you run **git commit** .

```
E:\Mobile_Computing\Report\MC_Reports>git add champion.py  
E:\Mobile_Computing\Report\MC_Reports>
```

2:git commit -m "Any message"

The **git commit** command captures a snapshot of the project's currently staged changes

The **git commit** command saves all staged changes

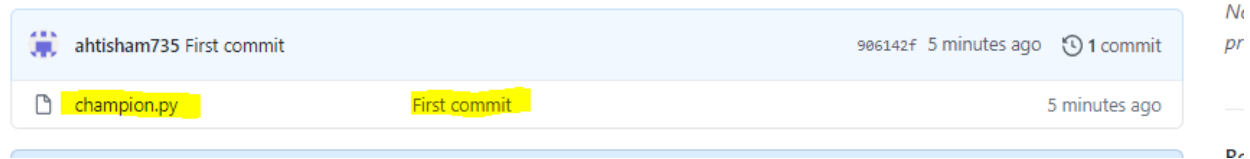
```
E:\Mobile_Computing\Report\MC_Reports>git commit -m "First commit"  
[master (root-commit) 906142f] First commit  
1 file changed, 35 insertions(+)  
create mode 100644 champion.py  
E:\Mobile_Computing\Report\MC_Reports>
```

3: git push

The **git push** command is used to upload local repository content to a remote repository.

```
E:\Mobile_Computing\Report\MC_Reports>git push  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 744 bytes | 186.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/ahtisham735/MC_Reports.git  
* [new branch]      master -> master
```

File has been uploaded on git, **BRAVO!!**



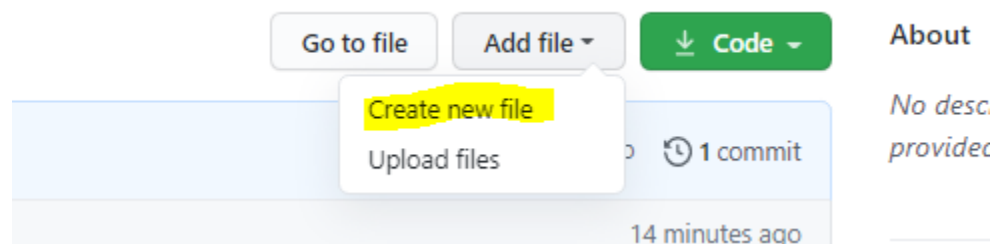
Creating/Updating file from Git:

We can create a new file or update an existing file from git.

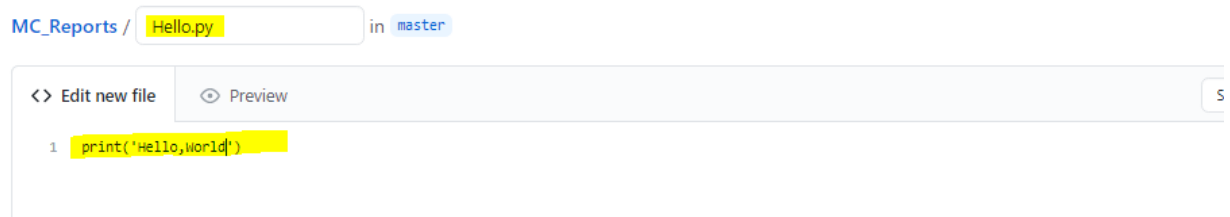
In order to reflect changes made on Git in local repository, we need

Run **git pull** command

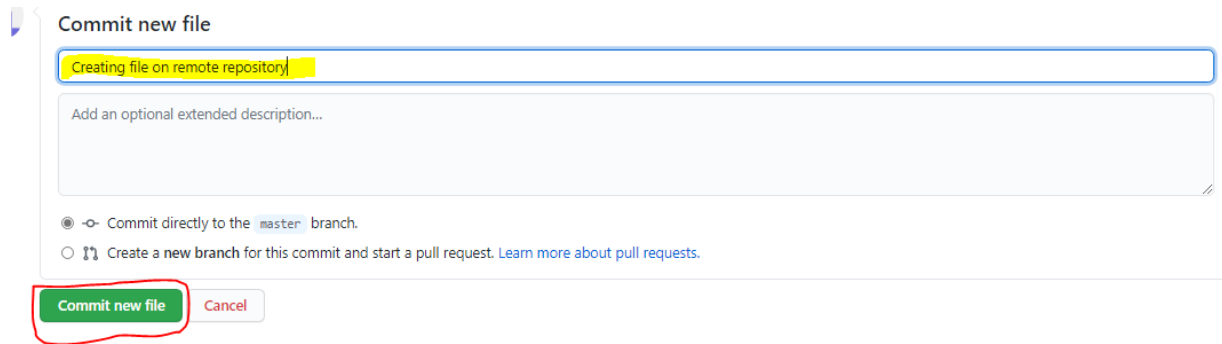
Creating a new file on remote repository:



Name the file & making some changes:



Giving a message & commit the new file:



File has been uploaded :

ahtisham735 Creating file on remote repository		c12f68a in 10 seconds	🕒 2 commits
📄	Hello.py	Creating file on remote repository	now

Git Pull:

In order to reflect changes made on remote repository ,we run **git pull** command:

```
E:\Mobile_Computing\Report\MC_Reports>git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 694 bytes | 3.00 KiB/s, done.
From https://github.com/ahtisham735/MC_Reports
 906142f..c12f68a master    -> origin/master
Updating 906142f..c12f68a
Fast-forward
 Hello.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Hello.py
E:\Mobile_Computing\Report\MC_Reports>
```

File has been pulled from remote: **BRAVO!!**

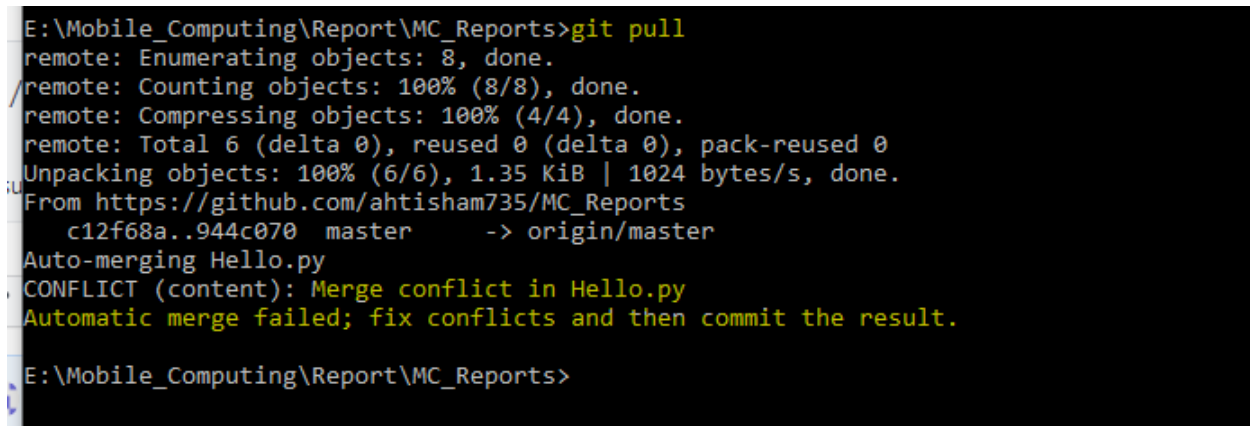
Merge Conflict:

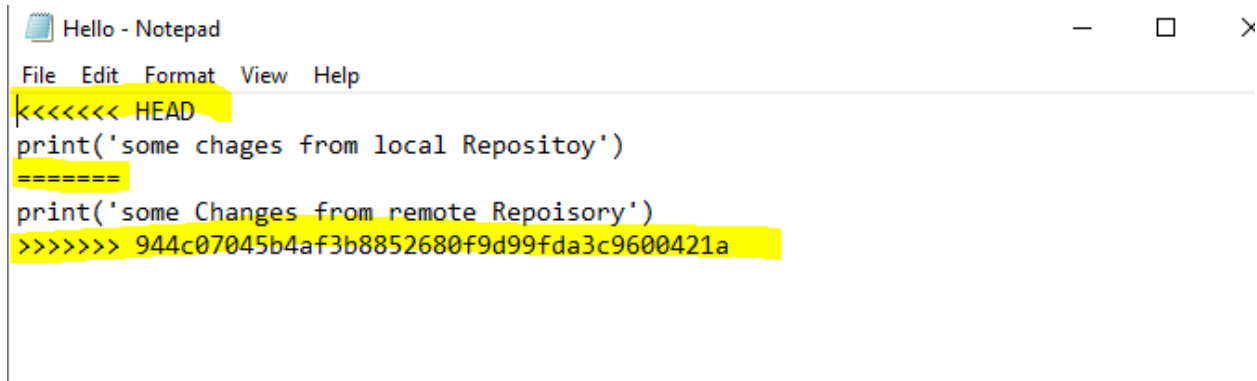
A **merge conflict** is an event that takes place when Git is unable to automatically resolve differences in code between two commits. Git can **merge** the changes automatically only if the commits are on different lines or branches.

Let's change the Hello.py file remotely and locally in order to understand merge Conflict



Now if we try to push/pull the changes, we will be facing **merge conflict**:





```
File Edit Format View Help
<<<<<< HEAD
print('some chages from local Repositoy')
=====
print('some Changes from remote Repoisory')
>>>>>> 944c07045b4af3b8852680f9d99fda3c9600421a
```

WHAT TO DO NOW??

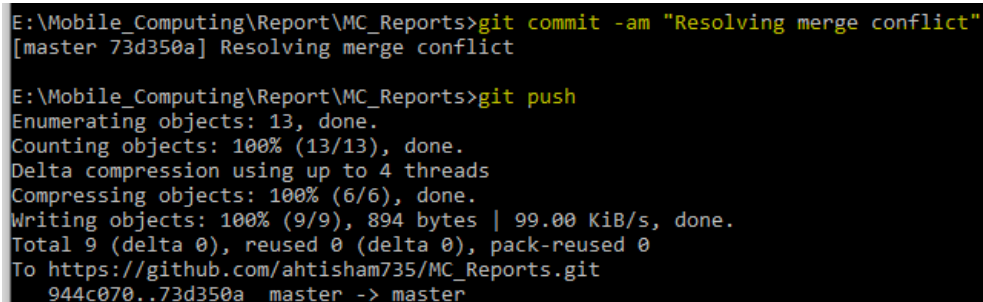
To resolve this conflict, just taking the best of both worlds.



```
File Edit Format View Help

print('some chages|')
```

Commit the changes and push these.



```
E:\Mobile_Computing\Report\MC_Reports>git commit -am "Resolving merge conflict"
[master 73d350a] Resolving merge conflict

E:\Mobile_Computing\Report\MC_Reports>git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 894 bytes | 99.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ahtisham735/MC_Reports.git
 944c070..73d350a master -> master
```

Merge Conflict has gone. **Bingo!**