



GIT AND GITHUB

Dr. Neeraj Kumar

Ruchika Verma

Research Associate, Case Western Reserve University

Ph.D. Candidate, Case Western Reserve University

VERSION CONTROL SYSTEMS

Version control helps in managing multiple versions of documents, programs, etc.

- Used universally in most organizations for real time project management
- Essential for collaborative team projects

Scenario 1 (personal project)	Scenario 2 (team project)
<ul style="list-style-type: none">• You wrote a story• Now you want to write two different climaxes of the same story (git branch)• Get both climaxes reviewed• Add the most acceptable climax to the main story (git merge)• Delete the other climax (git branch -d)	<ul style="list-style-type: none">• Writing a feature extraction code for segmented objects• You are working in a team<ul style="list-style-type: none">• Neeraj is coding for texture features• Ruchika is coding for shape features• Both programmers make changes to local copy of the main code<ul style="list-style-type: none">• Later review the changes and merge them with the main code

WHY VERSION CONTROL?

For working by yourself:

- Gives you a “time machine” for going back to earlier versions
- Gives you great support for different versions of the same basic project

For working with others:

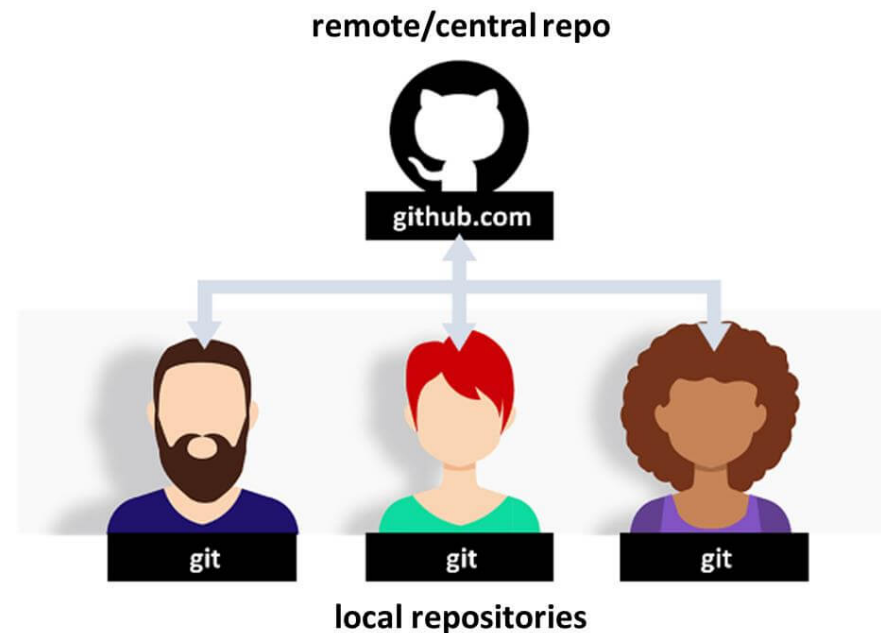
- Greatly simplifies concurrent work, merging changes

For getting an internship or job:

- Any company with a clue uses some kind of version control
- Companies without a clue are bad places to work

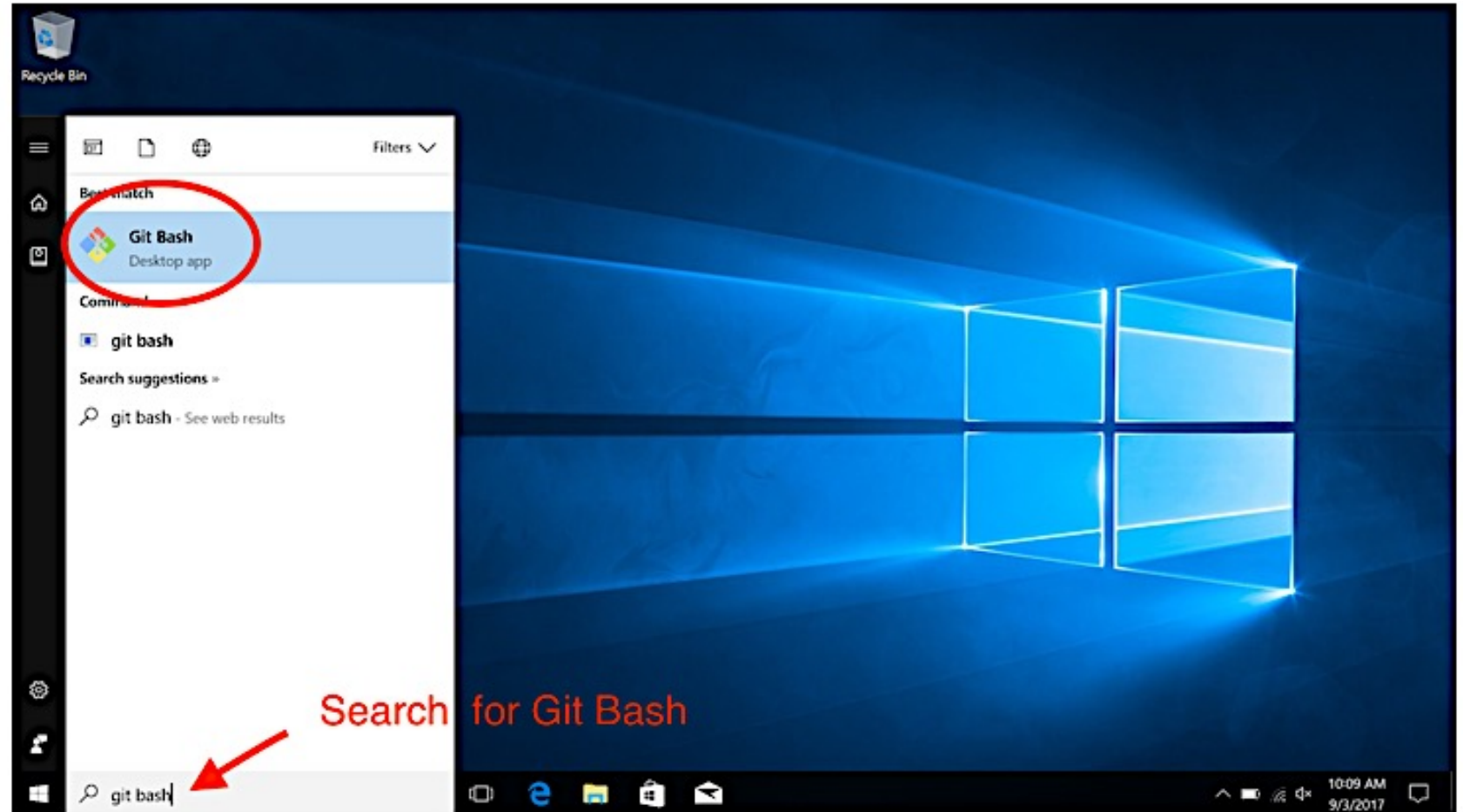
WHAT IS GIT AND GITHUB?

- Git is a free and open source distributed version control system designed to track changes to source code during software development.
 - But can be used to track changes in any set of files
- GitHub is a web-based Git repository hosting service

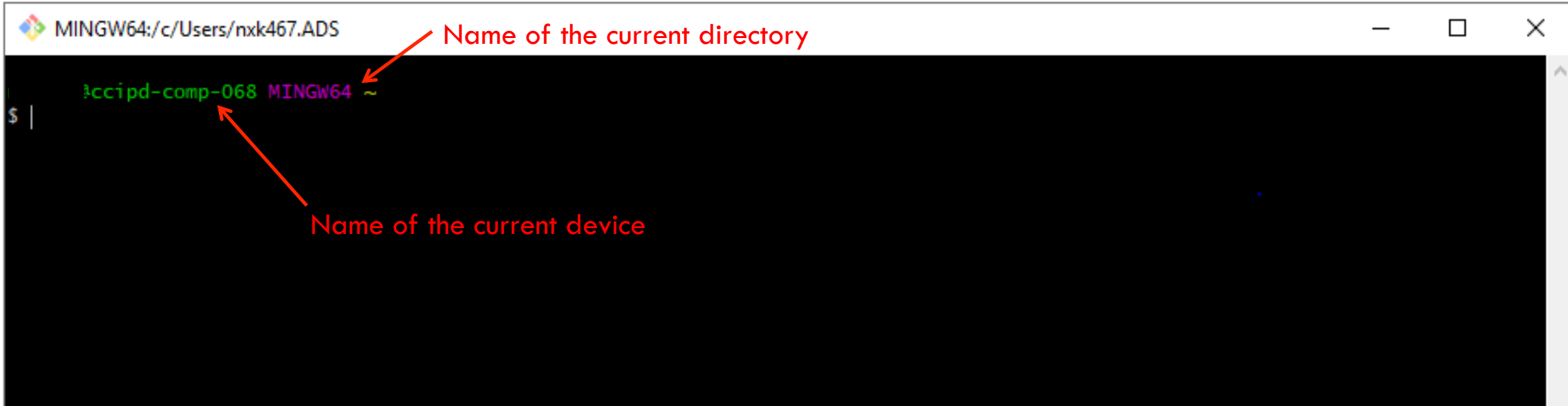


GETTING STARTED

- Download Git Bash from the following link
 - gitforwindows.org
- Open git bash as shown



GIT COMMAND LINE

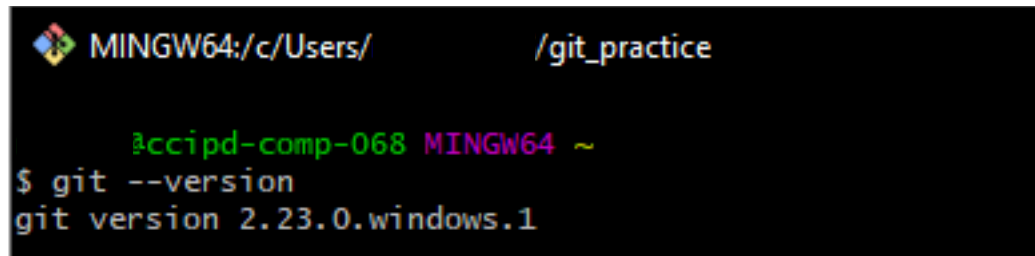


A terminal window titled 'MINGW64:/c/Users/nxk467.ADS'. The prompt is 'MINGW64 ~'. Two red arrows point to the prompt: one from the text 'Name of the current directory' pointing to 'MINGW64', and another from the text 'Name of the current device' pointing to '~'. The prompt is preceded by a green icon and the text 'MINGW64 ~'.

```
MINGW64:/c/Users/nxk467.ADS
MINGW64 ~
```

Check version of the git installed using the following command

`git --version`



A terminal window titled 'MINGW64:/c/Users/ /git_practice'. The prompt is 'MINGW64 ~'. The command 'git --version' has been entered and executed, resulting in the output 'git version 2.23.0.windows.1'.

```
MINGW64:/c/Users/ /git_practice
MINGW64 ~
$ git --version
git version 2.23.0.windows.1
```

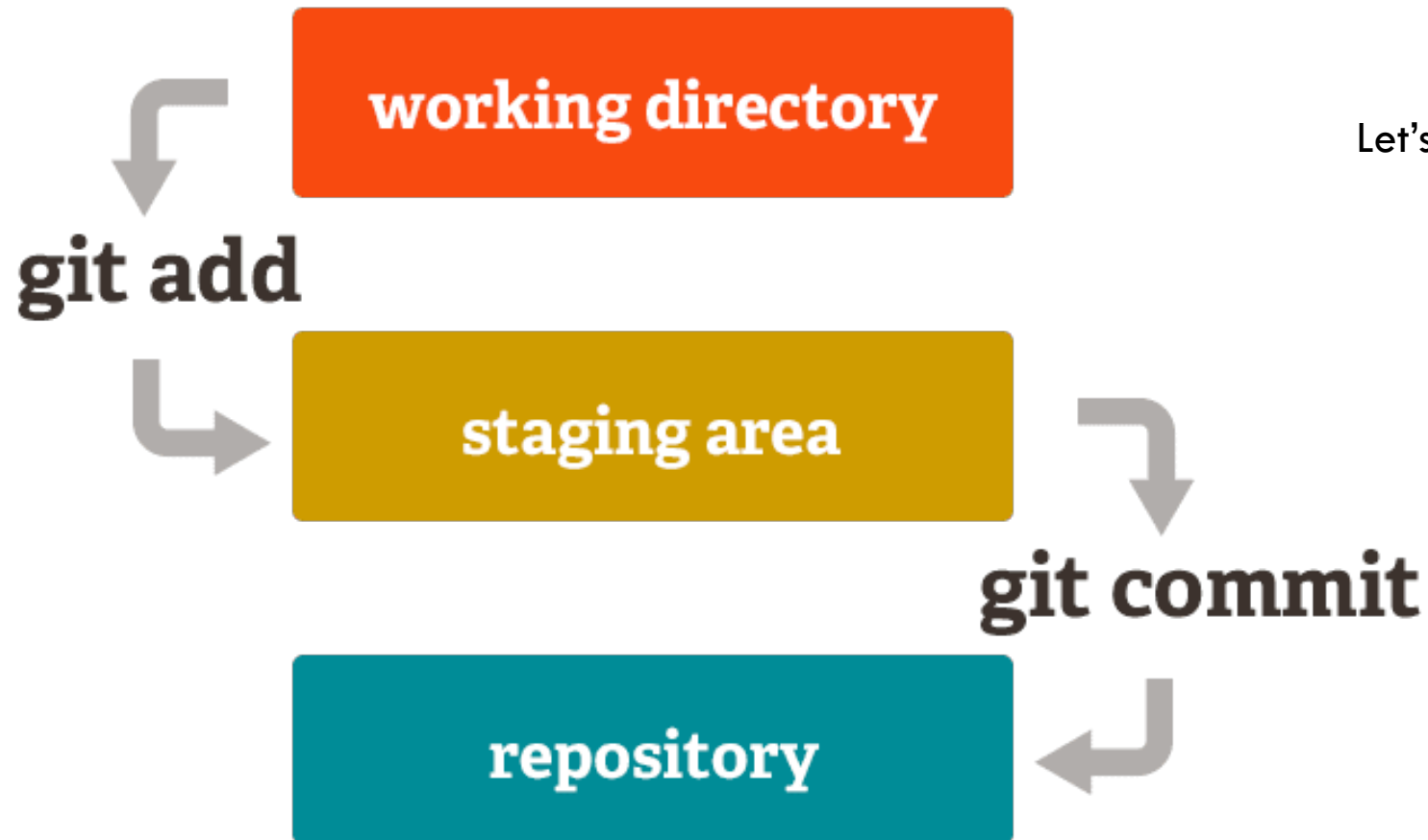
INTRODUCE YOURSELF TO GIT

- Set git username for every repository on your computer
 - `git config --global user.name "Mona Lisa"`
- Set git email for every repository on your computer
 - `git config --global user.email xyz@server.com`
- Confirm that the user name and email are set up correctly
 - `git config user.name`
 - `git config user.email`
- Want to use a different user name and email for a specific project?
 - `cd project_directory`
 - Use git config commands, without the `--global`

CREATING GIT REPOSITORY

- Move to the desired working directory using cd
 - `cd D:\`
- Create a new directory to practice git skills
 - `mkdir git_practice`
- `cd git_practice` to make the new directory your working directory
- `git init` to turn the current directory into a fresh git repository
- `echo "Hello Git and GitHub" >> README.txt` to create a new readme file with some text

GIT WORKFLOW



Let's try committing our readme

- `git add README.txt`
Ignore LF warning, if it appears
- `git commit -m "First Commit"`

GIT TRACKS CHANGES

- Open README.txt file in a text editor and enter the following line
 - Warning: You must read this file before proceeding!
- Save the README.txt file and return to Git Bash
- `git status` checks the status of the tracked files and also displays untracked files

```
accipd-comp-068 MINGW64 /d/git_practice (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:   README.txt

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

GIT DIFF

- `git diff` shows the changes that are not staged yet

```
@ccipd-comp-068 MINGW64 /d/git_practice (master)
$ git diff
warning: LF will be replaced by CRLF in README.txt.
The file will have its original line endings in your working directory
diff --git a/README.txt b/README.txt
index 41f3533..426f9e5 100644
--- a/README.txt
+++ b/README.txt
@@ -1,3 @@
 Hello Git and GitHub
+
+Warning: You must read this file before proceeding!
\ No newline at end of file
```

- `git add README.txt`
- `git commit -m "Warning added to readme file."`

GIT LOG

- `git log` shows a running log of commits
- A full log has the following pieces
 - A commit hash (SHA)
 - Commit author metadata- name and email address of the commit author
 - Commit date metadata- a date time stamp
 - Commit message- what the commit was about

```
lccipd-comp-068 MINGW64 /d/git_practice (master)
$ git log
commit f87c09ed87b1ee55c64f19ab9a36bc1ffc4bd42f (HEAD -> master)
Author: Neeraj Kumar <24767040+neerajkumarvaid@users.noreply.github.com>
Date: Sun Nov 3 19:04:31 2019 -0500

    Warning added to readme file.

commit a298edc4d5c7e6af1b341f17508232c4707afb5d
Author: Neeraj Kumar <24767040+neerajkumarvaid@users.noreply.github.com>
Date: Sun Nov 3 18:50:50 2019 -0500

    First Commit
```

BACKTRACKING I

- The commit you are currently on is known as the HEAD commit.
- `git show HEAD`

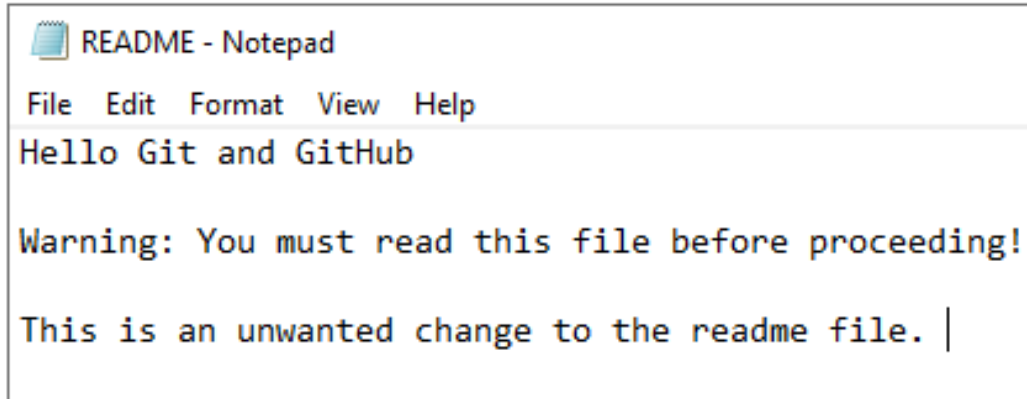
```
accipd-comp-068 MINGW64 /d/git_practice (master)
$ git show HEAD
commit f87c09ed87b1ee55c64f19ab9a36bc1ffc4bd42f (HEAD -> master)
Author: Neeraj Kumar <24767040+neerajkumarvaid@users.noreply.github.com>
Date:   Sun Nov 3 19:04:31 2019 -0500

    Warning added to readme file.

diff --git a/README.txt b/README.txt
index 41f3533..426f9e5 100644
--- a/README.txt
+++ b/README.txt
@@ -1,3 @@
 Hello Git and GitHub
+
+Warning: You must read this file before proceeding!
\ No newline at end of file
```

BACKTRACKING II

- Open README.txt and add the following line.
 - This is an unwanted change to the readme file.
- Save and close README.txt file.



```
README - Notepad
File Edit Format View Help
Hello Git and GitHub

Warning: You must read this file before proceeding!

This is an unwanted change to the readme file. |
```

- `git checkout HEAD README.txt` will restore the file.
- Open README.txt again.

BACKTRACKING III

- Accidentally added changed file to staging area.
- `git reset HEAD filename` unstages the files before a commit.
- Open README.txt and add the following line.
 - This is an unwanted change to the readme file.
- Save and close README.txt file.
- `git status` (check the status of git repository)
- `git add README.txt` (adds file to the staging area)
- `git status` (check the status of git repository)
- `git reset HEAD README.txt` (unstages the file)
- `git status` (check the status of git repository)
- `git checkout HEAD README.txt` (restores file to last commit)

```
i ~ :ipd-comp-068 MINGW64 /d/git_practice (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:   README.txt

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

i ~ :ipd-comp-068 MINGW64 /d/git_practice (master)
$ git add README.txt

i ~ :ipd-comp-068 MINGW64 /d/git_practice (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.txt

i ~ :ipd-comp-068 MINGW64 /d/git_practice (master)
$ git reset HEAD README.txt
Unstaged changes after reset:
M       README.txt

i ~ :ipd-comp-068 MINGW64 /d/git_practice (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:   README.txt

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

i ~ :ipd-comp-068 MINGW64 /d/git_practice (master)
$ git checkout HEAD README.txt
Updated 1 path from daef8d6
```

BACKTRACKING IV

- Accidentally made a commit of file changes.
- `git reset commit_SHA` restores file to previous commit.
- Add the following line to README.txt and save it.
 - This is an unwanted change to the readme file.
- `git add README.txt`
- `git commit -m "Unwanted line added."`
- `git log` (see the HEAD commit)
- `git reset f87c09e` (first 7 characters of commit SHA)
- `git log` (see the HEAD commit)
- `git checkout HEAD README.txt` (restores the file)

```
pd-comp-068 MINGW64 /d/git_practice (master)
$ git add README.txt

pd-comp-068 MINGW64 /d/git_practice (master)
$ git commit -m "Unwanted line added."
[master c0b8733] Unwanted line added.
1 file changed, 3 insertions(+), 1 deletion(-)

pd-comp-068 MINGW64 /d/git_practice (master)
$ git log
commit c0b8733438c586649f9ab6f7334ad388ae033c23 (HEAD -> master)
Author: Neeraj Kumar <24767040+neerajkumarvoid@users.noreply.github.com>
Date: Sun Nov 3 21:29:37 2019 -0500

    Unwanted line added.

commit f87c09ed87b1ee55c64f19ab9a36bc1ffc4bd42f
Author: Neeraj Kumar <24767040+neerajkumarvoid@users.noreply.github.com>
Date: Sun Nov 3 19:04:31 2019 -0500

    Warning added to readme file.

commit a298edc4d5c7e6af1b341f17508232c4707afb5d
Author: Neeraj Kumar <24767040+neerajkumarvoid@users.noreply.github.com>
Date: Sun Nov 3 18:50:50 2019 -0500

    First Commit

pd-comp-068 MINGW64 /d/git_practice (master)
$ git reset fa7c09e
fatal: ambiguous argument 'fa7c09e': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

pd-comp-068 MINGW64 /d/git_practice (master)
$ git reset f87c09e
Unstaged changes after reset:
M README.txt

pd-comp-068 MINGW64 /d/git_practice (master)
$ git show HEAD
commit f87c09ed87b1ee55c64f19ab9a36bc1ffc4bd42f (HEAD -> master)
Author: Neeraj Kumar <24767040+neerajkumarvoid@users.noreply.github.com>
Date: Sun Nov 3 19:04:31 2019 -0500

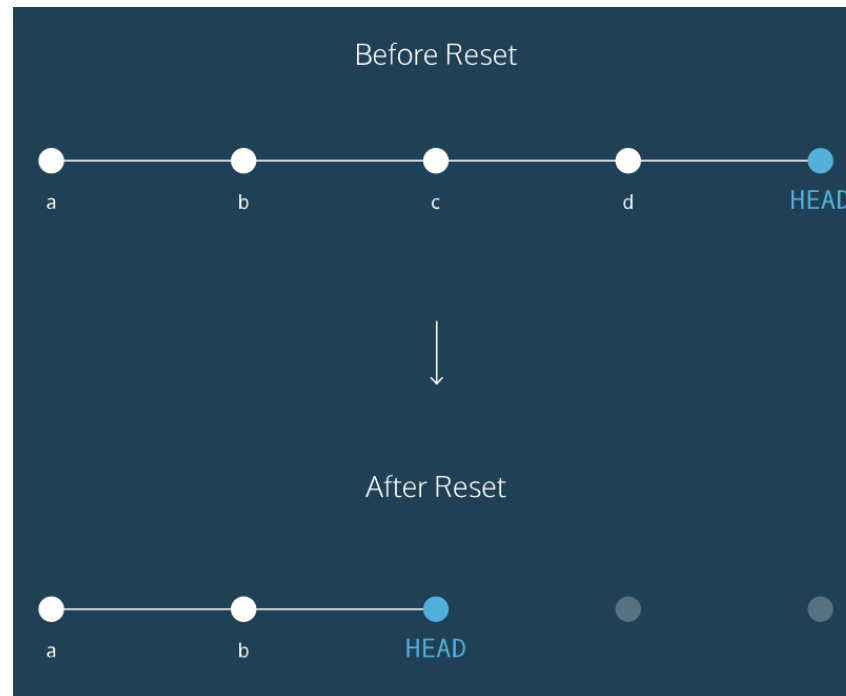
    Warning added to readme file.

diff --git a/README.txt b/README.txt
index 41f3533..426f9e5 100644
--- a/README.txt
+++ b/README.txt
@@ -1,3 @@
+Hello Git and GitHub
+
+Warning: You must read this file before proceeding!
+ \ No newline at end of file

pd-comp-068 MINGW64 /d/git_practice (master)
$ git checkout HEAD README.txt
Updated 1 path from daef8d6
```

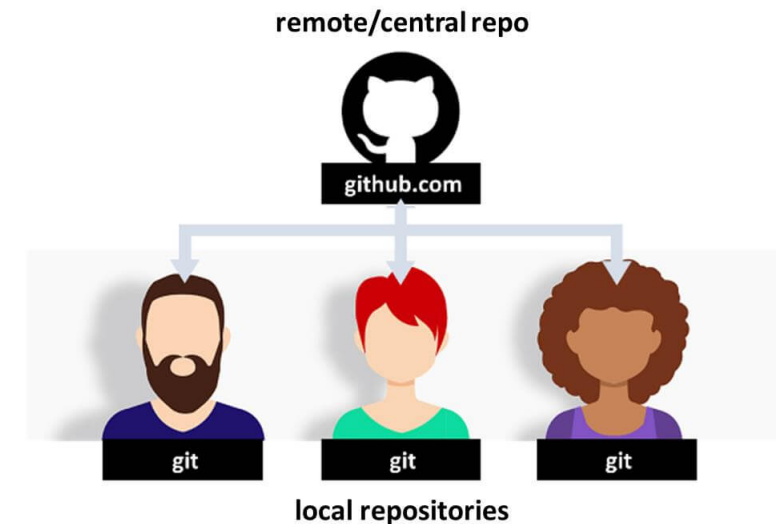

BACKTRACKING SUMMARY

- `git checkout HEAD filename` discards changes made in the working directory
- `git reset HEAD filename` unstages the file from the staging area
- `git reset commit_SHA` resets to a previous commit in your commit history.



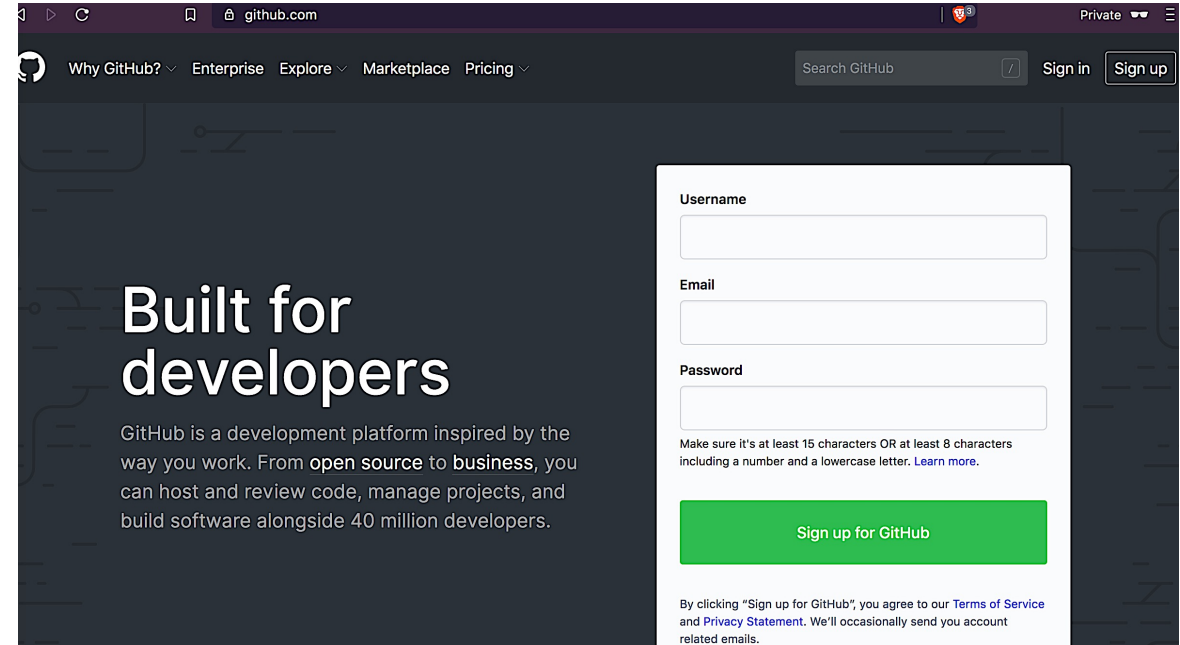
WHAT IS GIT AND GITHUB?

- GitHub is a web-based Git repository hosting service
- Allows collaborative development
- Allows you to know who made the changes and when
- Allows you to revert any changes and go back to a previous state

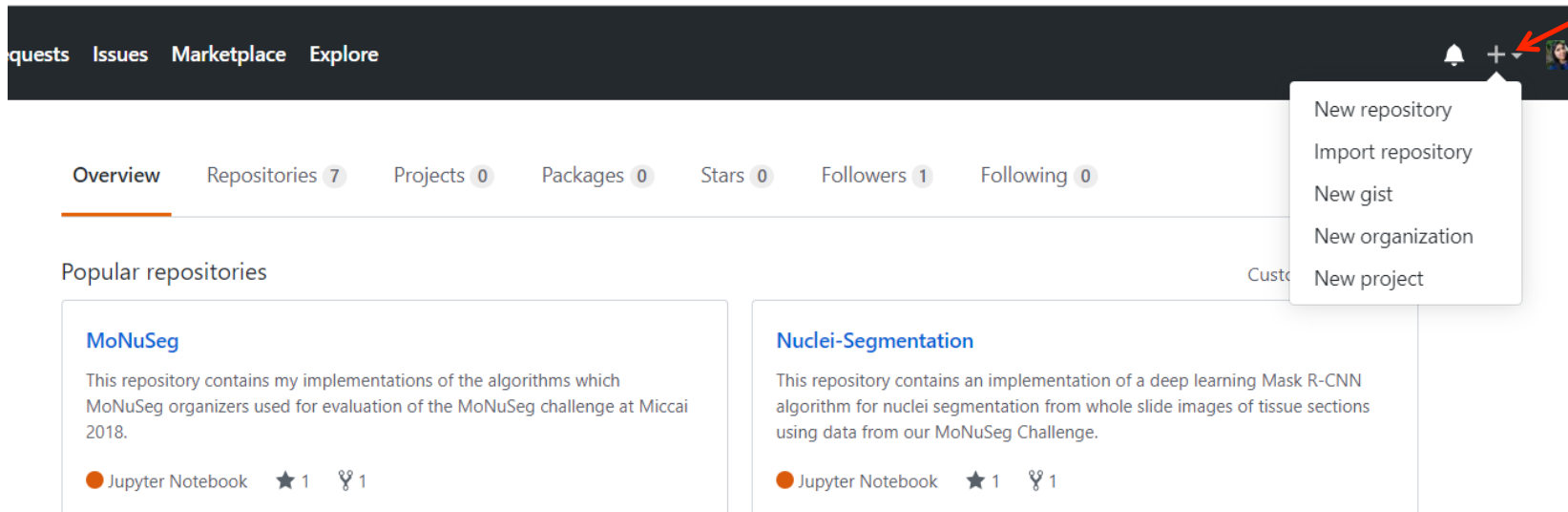


GITHUB

- Creating an account on github.com
- If you already have an account, just log in to it



- Create a new repository on github by clicking on the '+' sign



Select new repository

GITHUB REPOSITORY

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

/ git_practice ✓


Name it as "git_practice"

Great repository names are short and memorable. Need inspiration? How about **automatic-parakeet**.

Description (optional)

Add description if you want
"This is my first git repository on github"

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

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

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▼

Add a license: **None** ▼



Create repository

Click here when done

LOCAL TO GITHUB REPOSITORY

The screenshot shows the GitHub repository page for 'ruchikaverma-iitg / git_practice'. The repository has 1 watch, 0 stars, and 0 forks. The 'Code' tab is selected, showing options for cloning the repository. The 'Quick setup' section offers three methods: 'Set up in Desktop', 'HTTPS' (selected), and 'SSH'. The HTTPS URL is 'https://github.com/ruchikaverma-iitg/git_practice.git'. Below this, there are instructions for creating a new repository on the command line and pushing an existing repository from the command line. Red arrows point from external text to the 'HTTPS' tab and the 'git push' command in the second section.

[ruchikaverma-iitg](#) / [git_practice](#) Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or **HTTPS** [SSH](#) `https://github.com/ruchikaverma-iitg/git_practice.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# git_practice" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/ruchikaverma-iitg/git_practice.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/ruchikaverma-iitg/git_practice.git
git push -u origin master
```

Make sure that 'https' is selected


Use these commands to push files from your local git repository to this github repository

LOCAL TO GITHUB

- Copy following two commands from GitHub and paste in Git Bash

...or push an existing repository from the command line

```
git remote add origin https://github.com/  
git push -u origin master
```

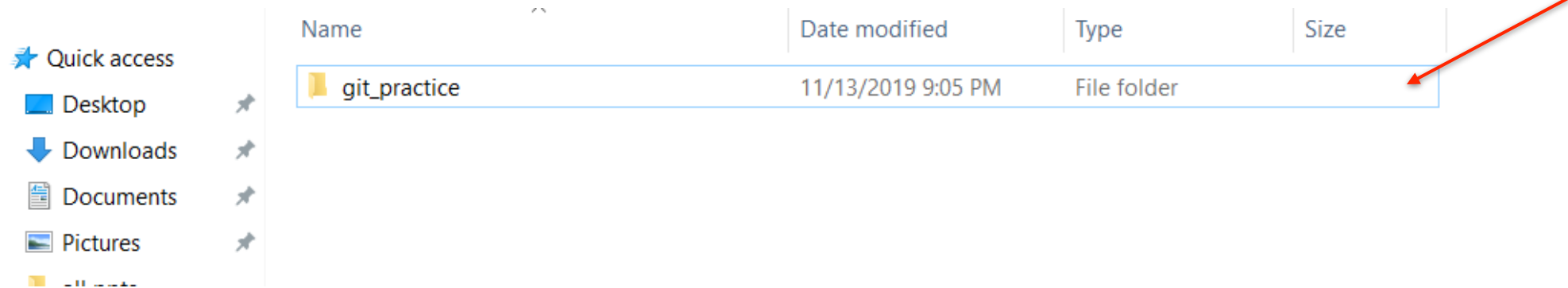


- When asked for username and password, enter your GitHub username and password
- When you see the following in Git bash, refresh GitHub webpage

```
-92Q60TF4 MINGW64 /d/git_practice (master)  
$ git remote add origin https://github.com/ruchikaverma-iitg/git_practice.git  
  
92Q60TF4 MINGW64 /d/git_practice (master)  
$ git push -u origin master  
Enumerating objects: 6, done.  
Counting objects: 100% (6/6), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (6/6), 542 bytes | 271.00 KiB/s, done.  
Total 6 (delta 1), reused 0 (delta 0)  
remote: Resolving deltas: 100% (1/1), done.  
To https://github.com/ruchikaverma-iitg/git_practice.git  
* [new branch]      master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

DELETE LOCAL

- Delete local git_practice



CLONING GITHUB REPOSITORY INTO LOCAL

- `git clone path` (Path of the remote directory)
Clone remote directory into the current directory

```
92Q60TF4 MINGW64 ~ (master)
$ cd D:
92Q60TF4 MINGW64 /d
$ git clone https://github.com/ruchikaverma-iitg/git_practice
Cloning into 'git_practice'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 1), reused 6 (delta 1), pack-reused 0
Unpacking objects: 100% (6/6), done.
```



PC > DATA (D:) > git_practice

Name	Date modified	Type	Size
.git	11/13/2019 9:24 PM	File folder	
README	11/13/2019 9:24 PM	Text Document	1 KB

README - Notepad

File Edit Format View Help

"Hello Git and GitHub"

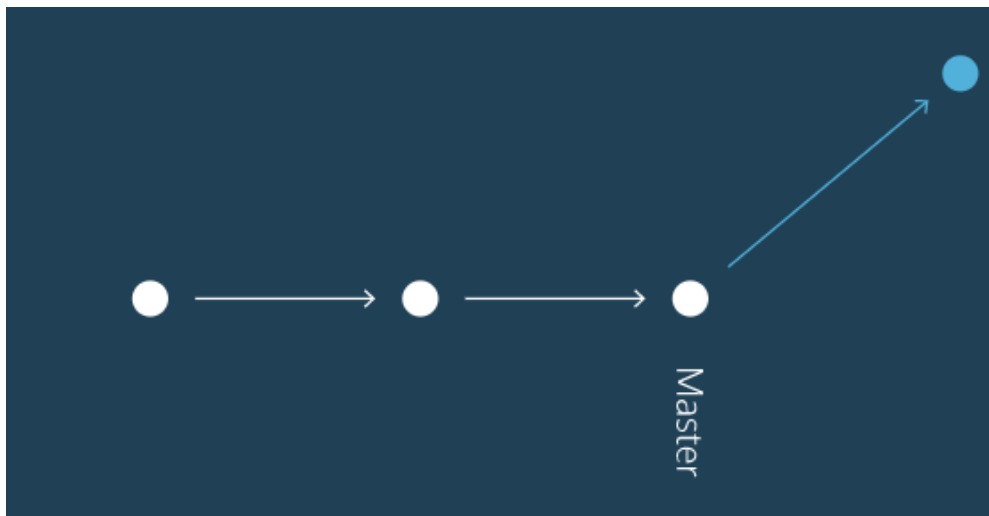
Warning: You must read this file before proceeding!

GIT BRANCHING I

- By default the branch name is called “master”

```
-92Q60TF4 MINGW64 /d
$ cd git_practice
-92Q60TF4 MINGW64 /d/git_practice (master)
$ |
```

- `git branch new_branch` create a new branch titled, “new_branch”



```
-92Q60TF4 MINGW64 /d/git_practice ((f50a3d9...))
$ git branch
* (HEAD detached at refs/heads/new_branch)
master
new_branch
```

GIT BRANCHING II

- `git checkout new_branch` switch to the new branch

```
-92Q60TF4 MINGW64 /d/git_practice (master)
$ git checkout new_branch
Switched to branch 'new_branch'
```

- `git log new_branch` shares the same commit history as “master”

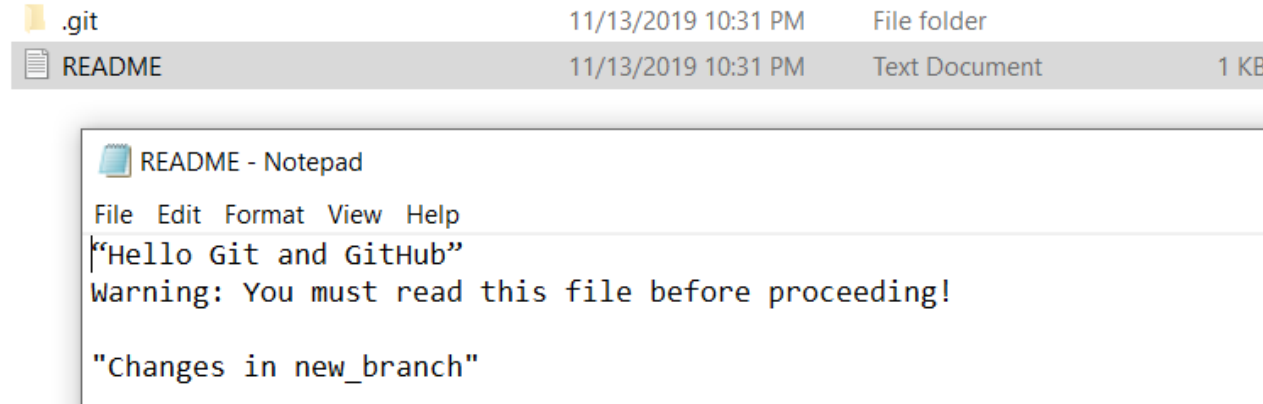
```
-92Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git log
commit f50a3d9868b56349eafaf12dfe546a63a2ce0f77 (HEAD -> new_branch, origin/master, origin/HEAD, master)
Author: Ruchika Verma <24606766+ruchikaverma-iitg@users.noreply.github.com>
Date:   Wed Nov 13 19:09:32 2019 -0500

    Warning added to readme file.

commit 2871c232e7aea6f05c54d5bc016fb60e16c9aef
Author: Ruchika Verma <24606766+ruchikaverma-iitg@users.noreply.github.com>
Date:   Wed Nov 13 19:05:50 2019 -0500

    First Commit
```

GIT BRANCHING III



Check status : README.txt got modified in new_branch

Add changes to the staging area

Commit changes in new_branch

```
2Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git status
On branch new_branch
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.txt

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

Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git add README.txt

92Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git commit -m "Changes in new_branch"
[new_branch 8525376] Changes in new_branch
1 file changed, 2 insertions(+)
```

GIT BRANCHING IV

Changes in new_branch are visible
in its commit history

```
-92Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git log
commit 8525376dd40a7655369127a46ac600b550addf00 (HEAD -> new_branch)
Author: Ruchika Verma <24606766+ruchikaverma-iitg@users.noreply.github.com>
Date:   Wed Nov 13 22:26:47 2019 -0500

    Changes in new_branch

commit f50a3d9868b56349eafaf12dfe546a63a2ce0f77 (origin/master, origin/HEAD, master)
Author: Ruchika Verma <24606766+ruchikaverma-iitg@users.noreply.github.com>
Date:   Wed Nov 13 19:09:32 2019 -0500

    Warning added to readme file.

commit 2871c232e7aea6f05c54d5bc016fb60e16c9aeff
Author: Ruchika Verma <24606766+ruchikaverma-iitg@users.noreply.github.com>
Date:   Wed Nov 13 19:05:50 2019 -0500

    First Commit
```

Switch to master

Changes in new_branch are not
visible in “master” commit history

```
-92Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

-92Q60TF4 MINGW64 /d/git_practice (master)
$ git log
commit f50a3d9868b56349eafaf12dfe546a63a2ce0f77 (HEAD -> master, origin/master, origin/HEAD)
Author: Ruchika Verma <24606766+ruchikaverma-iitg@users.noreply.github.com>
Date:   Wed Nov 13 19:09:32 2019 -0500

    Warning added to readme file.

commit 2871c232e7aea6f05c54d5bc016fb60e16c9aeff
Author: Ruchika Verma <24606766+ruchikaverma-iitg@users.noreply.github.com>
Date:   Wed Nov 13 19:05:50 2019 -0500

    First Commit
```

GIT MERGE

- `git merge new_branch` update master with the changes made in new_branch

Switch to master

```
-92Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

Merge changes

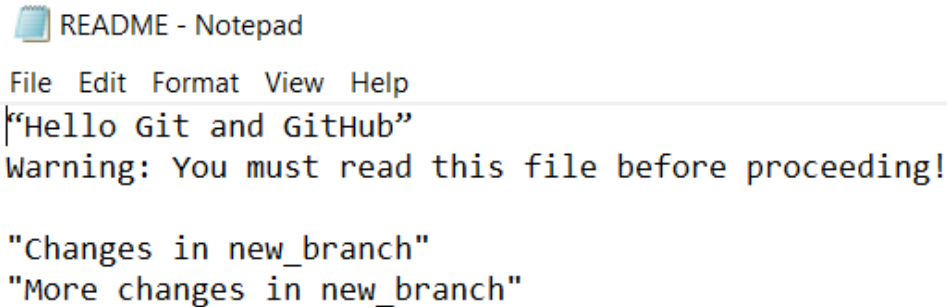
```
-92Q60TF4 MINGW64 /d/git_practice (master)
$ git merge new_branch
Updating f50a3d9..8525376
Fast-forward
 README.txt | 2 ++
 1 file changed, 2 insertions(+)
```

Merge was successful because master had not changed since we made a commit on new_branch

MERGE CONFLICT I

- What would happen if we made a commit on master before merging the updated new_branch?

```
-92Q60TF4 MINGW64 /d/git_practice (master)
$ git checkout new_branch
Switched to branch 'new_branch'
```



README - Notepad

File Edit Format View Help

"Hello Git and GitHub"

Warning: You must read this file before proceeding!

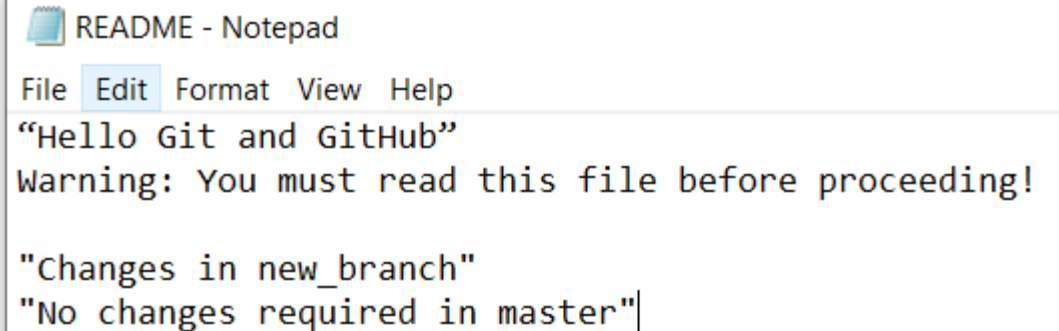
"Changes in new_branch"

"More changes in new_branch"

```
-92Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git add README.txt

-92Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git commit -m "Committing more changes in new_branch"
[new_branch 6c4f558] Committing more changes in new_branch
1 file changed, 1 insertion(+)
```

```
-92Q60TF4 MINGW64 /d/git_practice (new_branch)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
```



README - Notepad

File Edit Format View Help

"Hello Git and GitHub"

Warning: You must read this file before proceeding!

"Changes in new_branch"

"No changes required in master"

```
-92Q60TF4 MINGW64 /d/git_practice (master)
$ git add README.txt

-92Q60TF4 MINGW64 /d/git_practice (master)
$ git commit -m "Changes in master"
[master 63c3eb8] Changes in master
1 file changed, 1 insertion(+)
```

MERGE CONFLICT II

```
-92Q60TF4 MINGW64 /d/git_practice (master)
$ git merge new_branch
Auto-merging README.txt
CONFLICT (content): Merge conflict in README.txt
Automatic merge failed; fix conflicts and then commit the result.
```

README - Notepad

File Edit Format View Help

"Hello Git and GitHub"

Warning: You must read this file before proceeding!

"Changes in new_branch"

<<<<<< HEAD

"No changes required in master"

=====

"More changes in new_branch"

>>>>>> new_branch

Merge conflict

README - Notepad

File Edit Format View Help

"Hello Git and GitHub"

Warning: You must read this file before proceeding!

"Changes in new_branch"

"More changes in new_branch"

```
92Q60TF4 MINGW64 /d/git_practice (master|MERGING)
$ git add README.txt

92Q60TF4 MINGW64 /d/git_practice (master|MERGING)
$ git commit -m "Final Commit"
[master be62272] Final Commit
```

DELETE BRANCH

- `git branch -d branch_name` (to be deleted)

```
PS-92Q60TF4 MINGW64 /d/git_practice (master)
$ git branch -d new_branch
Deleted branch new_branch (was 6c4f558).

PS-92Q60TF4 MINGW64 /d/git_practice (master)
$ git branch
* master
```


GIT PUSH LOCAL TO GIT

```
~-92Q60TF4 MINGW64 /d/git_practice (master)
$ git push
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1.11 KiB | 377.00 KiB/s, done.
Total 12 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/ruchikaverma-iitg/git_practice
f50a3d9..be62272 master -> master
```



ruchikaverma-iitg / git_practice

[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#)


Description

GitHub short course

Manage topics

6 commits 1 branch 0 packages

Branch: master [New pull request](#)

 ruchikaverma-iitg Final Commit

[README.txt](#) Final Commit

README.txt

"Hello Git and GitHub"

Warning: You must read this file before proceeding!

"Changes in new_branch"

"More changes in new_branch"

GIT CLONE

```
PS C:\Users\r92Q60TF4 MINGW64 /d/git_practice (master)
$ git clone https://github.com/ruchikaverma-iitg/git_practice quiz
Cloning into 'quiz'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 18 (delta 6), reused 17 (delta 5), pack-reused 0
Unpacking objects: 100% (18/18), done.
```

Clone remote repository to local directory
named quiz




This PC > DATA (D:) > git_practice

Name	Date modified	Type	Size
.git	11/13/2019 11:30 PM	File folder	
git_practice	11/14/2019 12:11 AM	File folder	
quiz	11/14/2019 12:14 AM	File folder	
README	11/13/2019 11:28 PM	Text Document	1 KB

GIT FETCH I

```
-92Q60TF4 MINGW64 /d/git_practice (master)
$ git remote -v
origin  https://github.com/ruchikaverma-iitg/git_practice (fetch)
origin  https://github.com/ruchikaverma-iitg/git_practice (push)
```



 **ruchikaverma-iitg** Update README.txt

1 contributor

9 lines (5 sloc) | 182 Bytes

```
1  "Hello Git and GitHub"
2  Warning: You must read this file before proceeding!
3
4  "Changes in new_branch"
5  "More changes in new_branch"
6
7  "A few more changes to finish the last topic!"
8
```

Remote repository from where the cloning was done called “remote” for reference

- New changes in the remote repository
- Clone (in quiz) is not up-to-date on the local machine

GIT FETCH II

```
2Q60TF4 MINGW64 /d/git_practice/quiz (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/ruchikaverma-iitg/git_practice
   be62272..20b00d2  master    -> origin/master
```

- Merge changes from remote into the local repository (quiz)
- Bring new changes in the origin/master (different from “master”)

- New changes are in the repository called “origin/master”

GIT MERGE ORIGIN/MASTER

README - Notepad

File Edit Format View Help

"Hello Git and GitHub"

Warning: You must read this file before proceeding!

"Changes in new_branch"

"More changes in new_branch"

README - Notepad

File Edit Format View Help



"Hello Git and GitHub"

Warning: You must read this file before proceeding!

"Changes in new_branch"

"More changes in new_branch"

"A few more changes to finish the last topic!"



```
12Q60TF4 MINGW64 /d/git_practice/quiz (master)
$ git merge origin/master
Updating be62272..20b00d2
Fast-forward
 README.txt | 1 +
 1 file changed, 1 insertion(+)
```

- Merge changes in the "master"
- Complete your assignment/quiz

GIT PUSH

```
92Q60TF4 MINGW64 /d/git_practice/quiz (master)
$ git add README.txt

92Q60TF4 MINGW64 /d/git_practice/quiz (master)
$ git commit -m "Finishing!"
[master ac32692] Finishing!
1 file changed, 1 insertion(+)

92Q60TF4 MINGW64 /d/git_practice/quiz (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 320 bytes | 106.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ruchikaverma-iitg/git_practice
20b00d2..ac32692 master -> master
```



- Add and commit new changes
- Push master to the remote (GITHUB) for evaluation

README - Notepad

File Edit Format View Help

"Hello Git and GitHub"

Warning: You must read this file before proceeding!

"Changes in new_branch"

"More changes in new_branch"

"A few more changes to finish the last topic!"

"Quiz completed :)"



ruchikaverma-iitg Finishing!



[README.txt](#)



README.txt

"Hello Git and GitHub"

Warning: You must read this file before proceeding!

"Changes in new_branch"

"More changes in new_branch"

"A few more changes to finish the last topic!"

"Quiz completed :)"

THANK YOU!