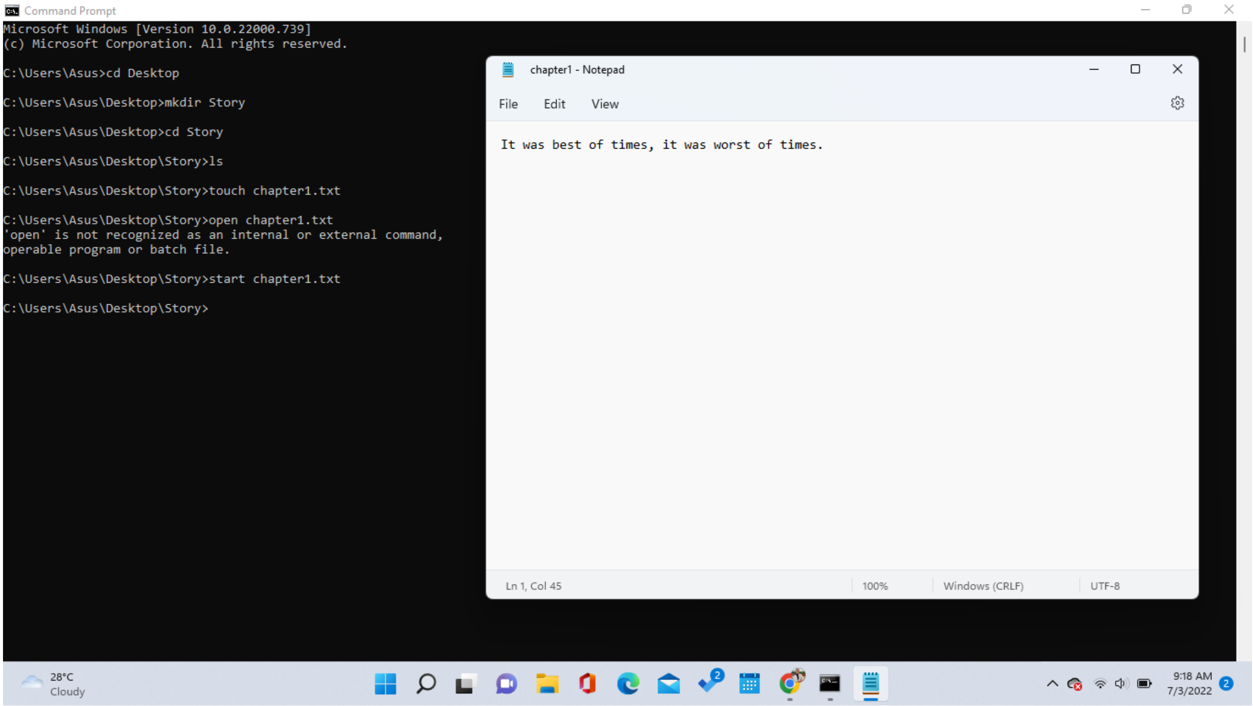
Version control using git and command line



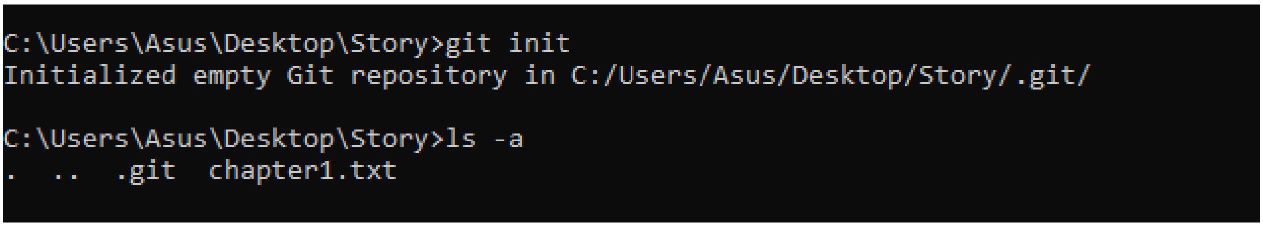
```
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>cd Desktop

C:\Users\Asus\Desktop>mkdir Story

C:\Users\Asus\Desktop>cd Story

C:\Users\Asus\Desktop\Story>ls

C:\Users\Asus\Desktop\Story>touch chapter1.txt

C:\Users\Asus\Desktop\Story>open chapter1.txt
'open' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Asus\Desktop\Story>start chapter1.txt

C:\Users\Asus\Desktop\Story>
```

```
It was best of times, it was worst of times.
```

Alt + f4 to force close app

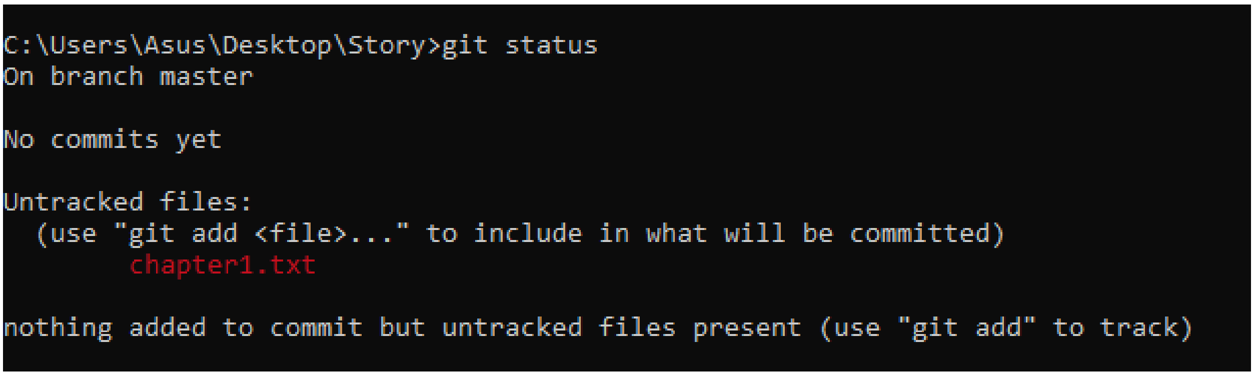
Initialize git - `git init`

Git is initialized as a hidden file.



```
C:\Users\Asus\Desktop\Story>git init
Initialized empty Git repository in C:/Users/Asus/Desktop/Story/.git/

C:\Users\Asus\Desktop\Story>ls -a
.   ..   .git   chapter1.txt
```

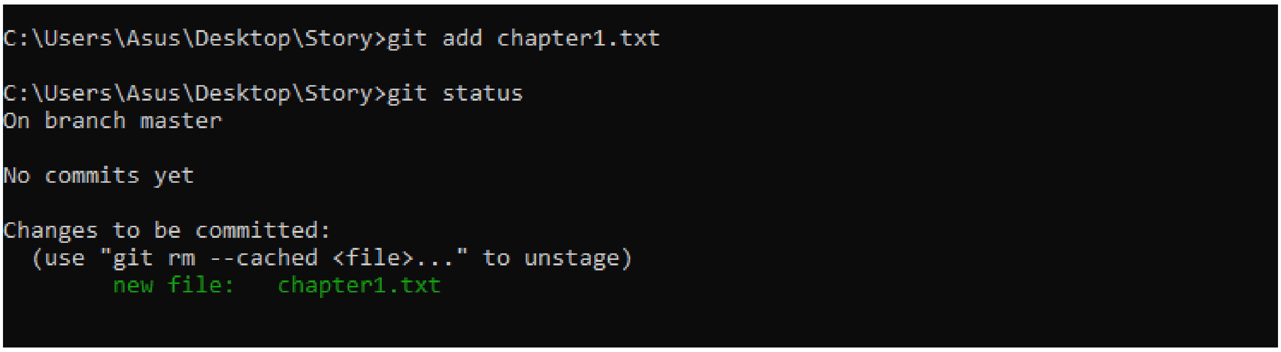
To see what's currently in the **staging area** – `git status`



```
C:\Users\Asus\Desktop\Story>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        chapter1.txt

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

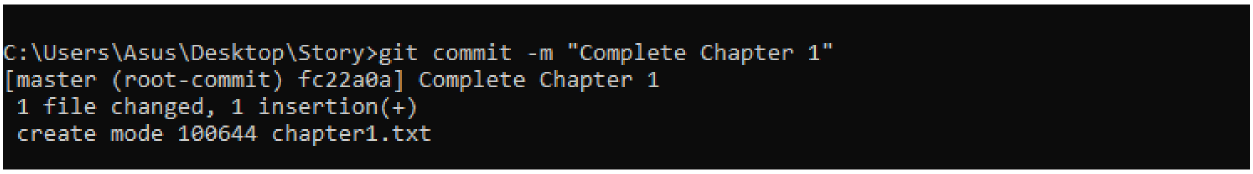
untracked files are shown in red.

to **add it to the staging area** and start tracking changes in it use – `git add`



```
C:\Users\Asus\Desktop\Story>git add chapter1.txt

C:\Users\Asus\Desktop\Story>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   chapter1.txt
```

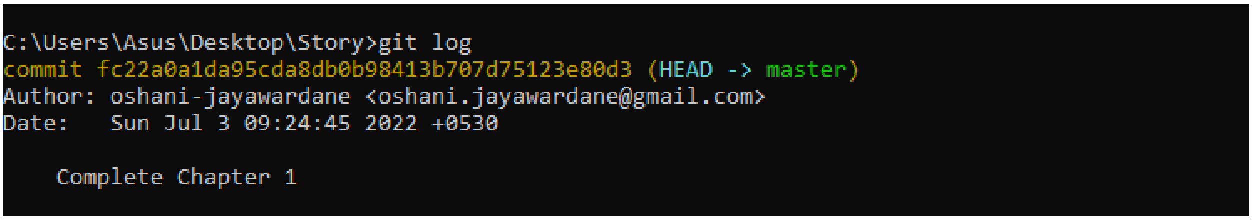
Commit the files – `git commit -m “commit message”`



```
C:\Users\Asus\Desktop\Story>git commit -m "Complete Chapter 1"
[master (root-commit) fc22a0a] Complete Chapter 1
 1 file changed, 1 insertion(+)
 create mode 100644 chapter1.txt
```

commit message by convention is written in present tense

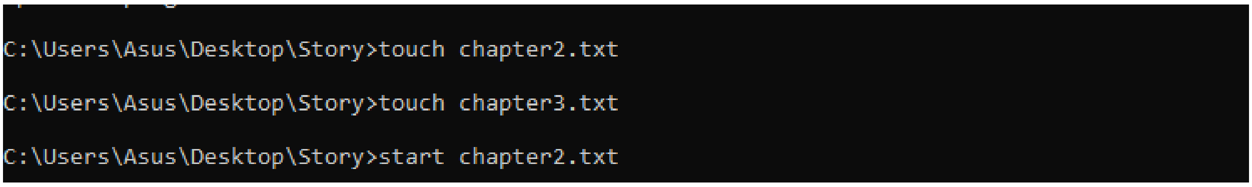
to **see details** of the commit – `git log`



```
C:\Users\Asus\Desktop\Story>git log
commit fc22a0a1da95cda8db0b98413b707d75123e80d3 (HEAD -> master)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:24:45 2022 +0530

    Complete Chapter 1
```

hash uniquely identifies the particular commit

creating and editing a couple more files afterwards

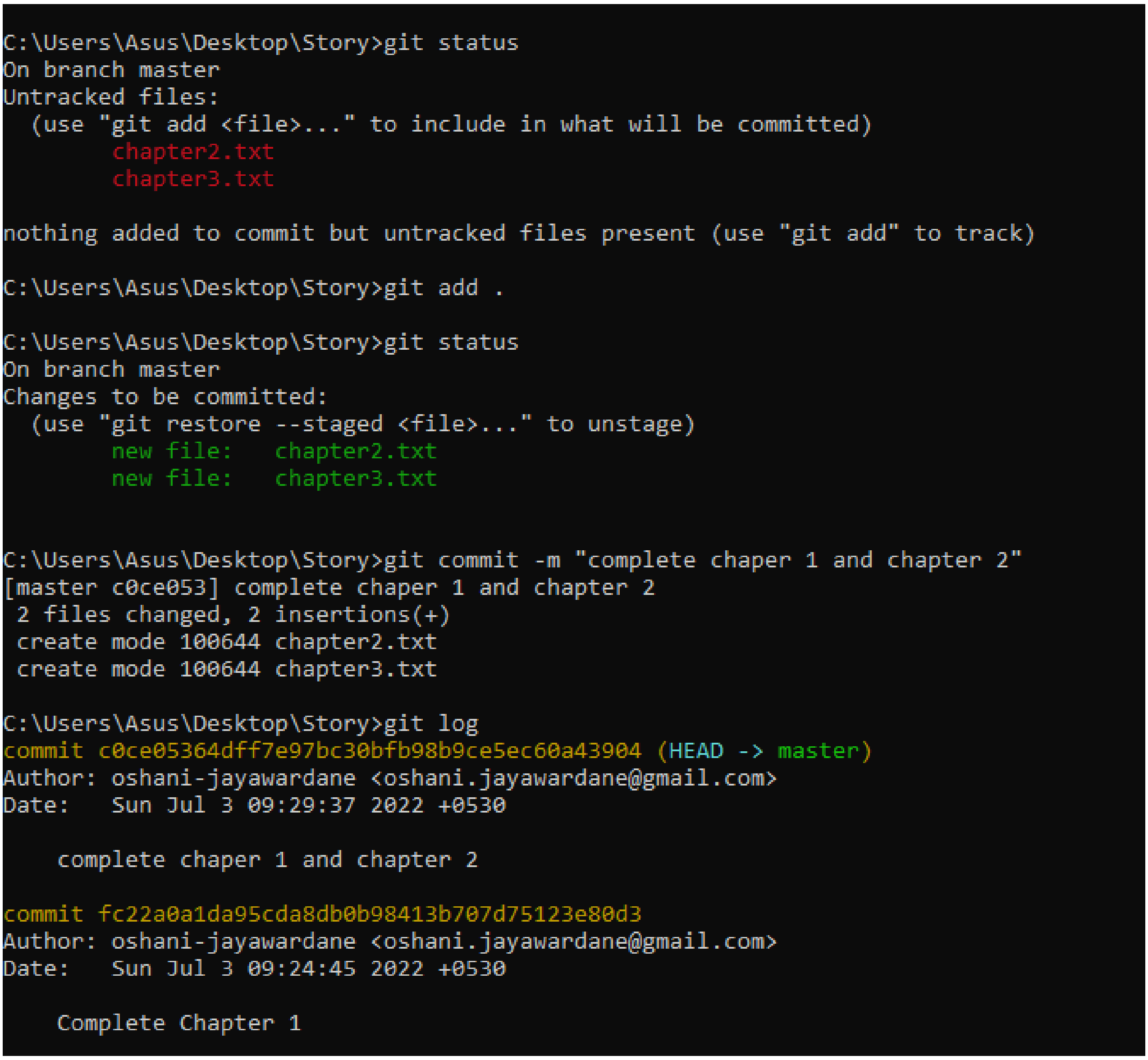


```
C:\Users\Asus\Desktop\Story>touch chapter2.txt

C:\Users\Asus\Desktop\Story>touch chapter3.txt

C:\Users\Asus\Desktop\Story>start chapter2.txt
```

Add both **(all) files** at once – `git add .`

(below commands have a mistake in the commit message – it should be:

git commit -m "complete chaper 2 and chapter 3")



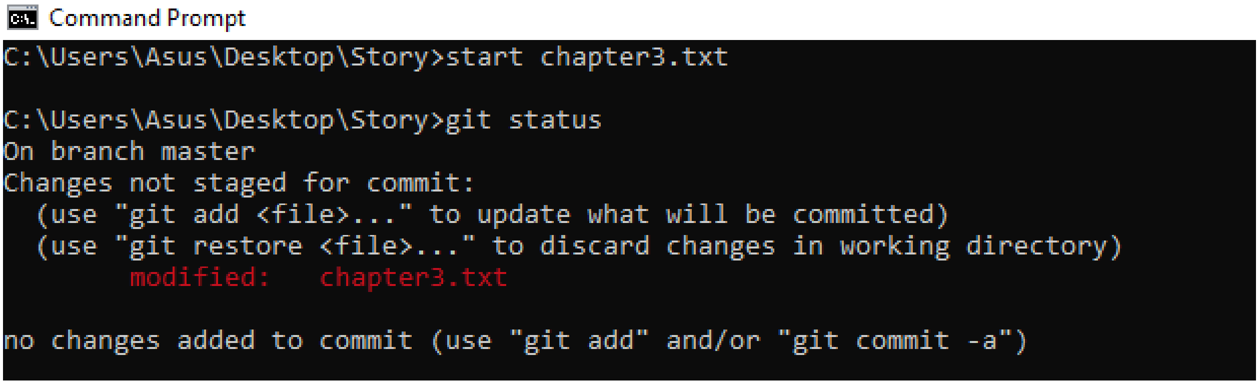
```
C:\Users\Asus\Desktop\Story>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        chapter2.txt
        chapter3.txt

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

C:\Users\Asus\Desktop\Story>git add .

C:\Users\Asus\Desktop\Story>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   chapter2.txt
        new file:   chapter3.txt


C:\Users\Asus\Desktop\Story>git commit -m "complete chaper 1 and chapter 2"
[master c0ce053] complete chaper 1 and chapter 2
 2 files changed, 2 insertions(+)
 create mode 100644 chapter2.txt
 create mode 100644 chapter3.txt

C:\Users\Asus\Desktop\Story>git log
commit c0ce05364dff7e97bc30bfb98b9ce5ec60a43904 (HEAD -> master)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:29:37 2022 +0530

    complete chaper 1 and chapter 2

commit fc22a0a1da95cda8db0b98413b707d75123e80d3
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:24:45 2022 +0530

    Complete Chapter 1
```

two commits. with two different hashes

word **'HEAD'** represents the current position we are in

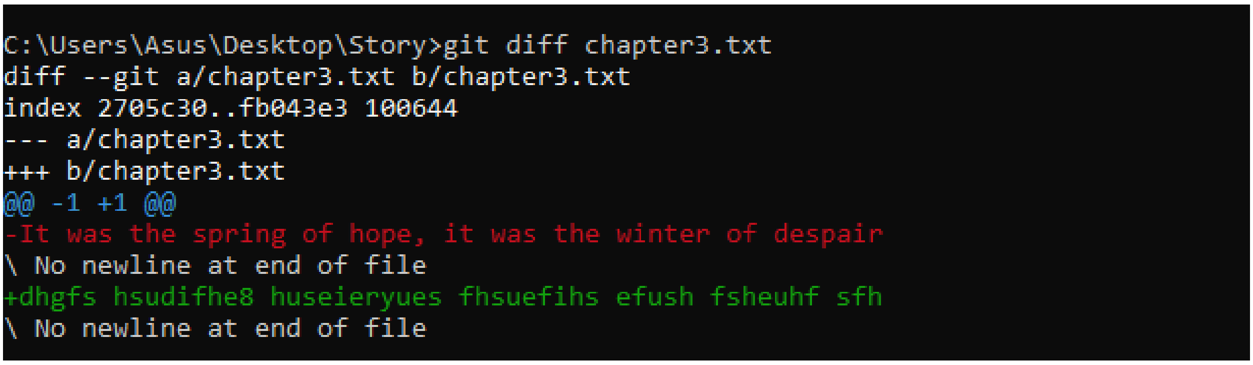
git add adds file to staging area. **why staging area?** It holds all files and let decide which files to commit. **git commit pushes selected files from staging area to git local repository**. Therefore, even if we mess up our file, we can revert files back to the working directory and use the last version under version control.

Ex: say we mess up chapter3.txt



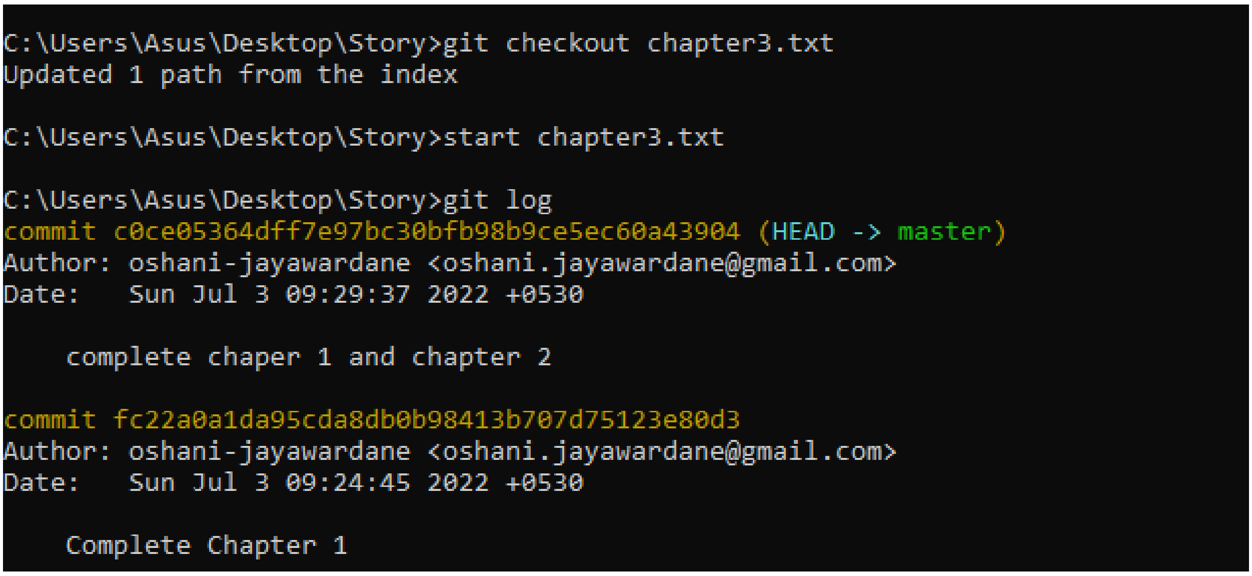
```
C:\Users\Asus\Desktop\Story>start chapter3.txt

C:\Users\Asus\Desktop\Story>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:   chapter3.txt

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

to **check the differences** between the files – `git diff`



```
C:\Users\Asus\Desktop\Story>git diff chapter3.txt
diff --git a/chapter3.txt b/chapter3.txt
index 2705c30..fb043e3 100644
--- a/chapter3.txt
+++ b/chapter3.txt
@@ -1 +1 @@
-It was the spring of hope, it was the winter of despair
\ No newline at end of file
+dhgfs hsudifhe8 huseieryues fhsuefihs efush fsheuhf sfh
\ No newline at end of file
```

To **revert back** to last checkpoint – `git checkout`

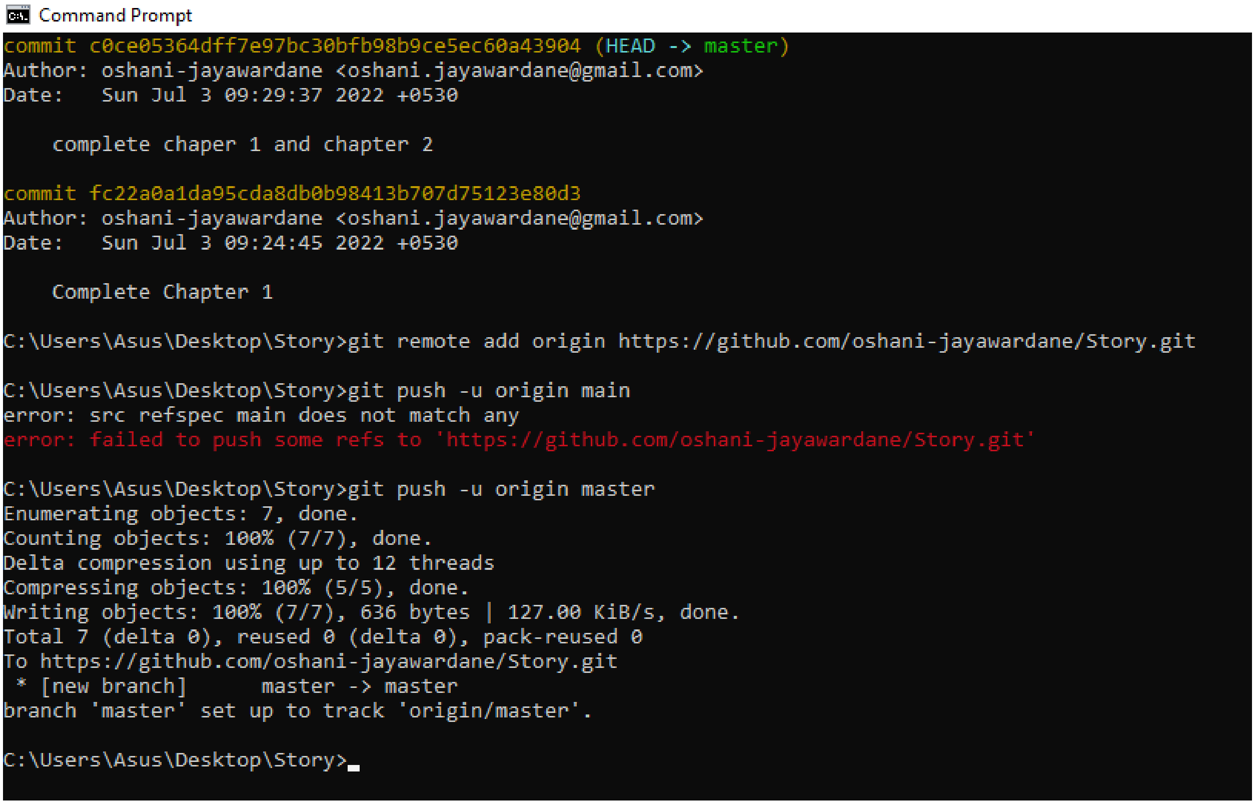


```
C:\Users\Asus\Desktop\Story>git checkout chapter3.txt
Updated 1 path from the index

C:\Users\Asus\Desktop\Story>start chapter3.txt

C:\Users\Asus\Desktop\Story>git log
commit c0ce05364dff7e97bc30bfb98b9ce5ec60a43904 (HEAD -> master)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:29:37 2022 +0530

    complete chaper 1 and chapter 2

commit fc22a0a1da95cda8db0b98413b707d75123e80d3
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:24:45 2022 +0530

    Complete Chapter 1
```

GitHub

Create a new repository in GitHub account – Story

(Commands here are different from the new version – check and follow instructions in the GitHub page when creating repo)



```
commit c0ce05364dff7e97bc30bfb98b9ce5ec60a43904 (HEAD -> master)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:29:37 2022 +0530

    complete chaper 1 and chapter 2

commit fc22a0a1da95cda8db0b98413b707d75123e80d3
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:24:45 2022 +0530

    Complete Chapter 1

C:\Users\Asus\Desktop\Story>git remote add origin https://github.com/oshani-jayawardane/Story.git

C:\Users\Asus\Desktop\Story>git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/oshani-jayawardane/Story.git'

C:\Users\Asus\Desktop\Story>git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 636 bytes | 127.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/oshani-jayawardane/Story.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

C:\Users\Asus\Desktop\Story>
```

Go to -> insights -> network -> and check the graph of **commit history**

Gitignore

Prevent committing sensitive information (passwords, api keys etc.) in public commits

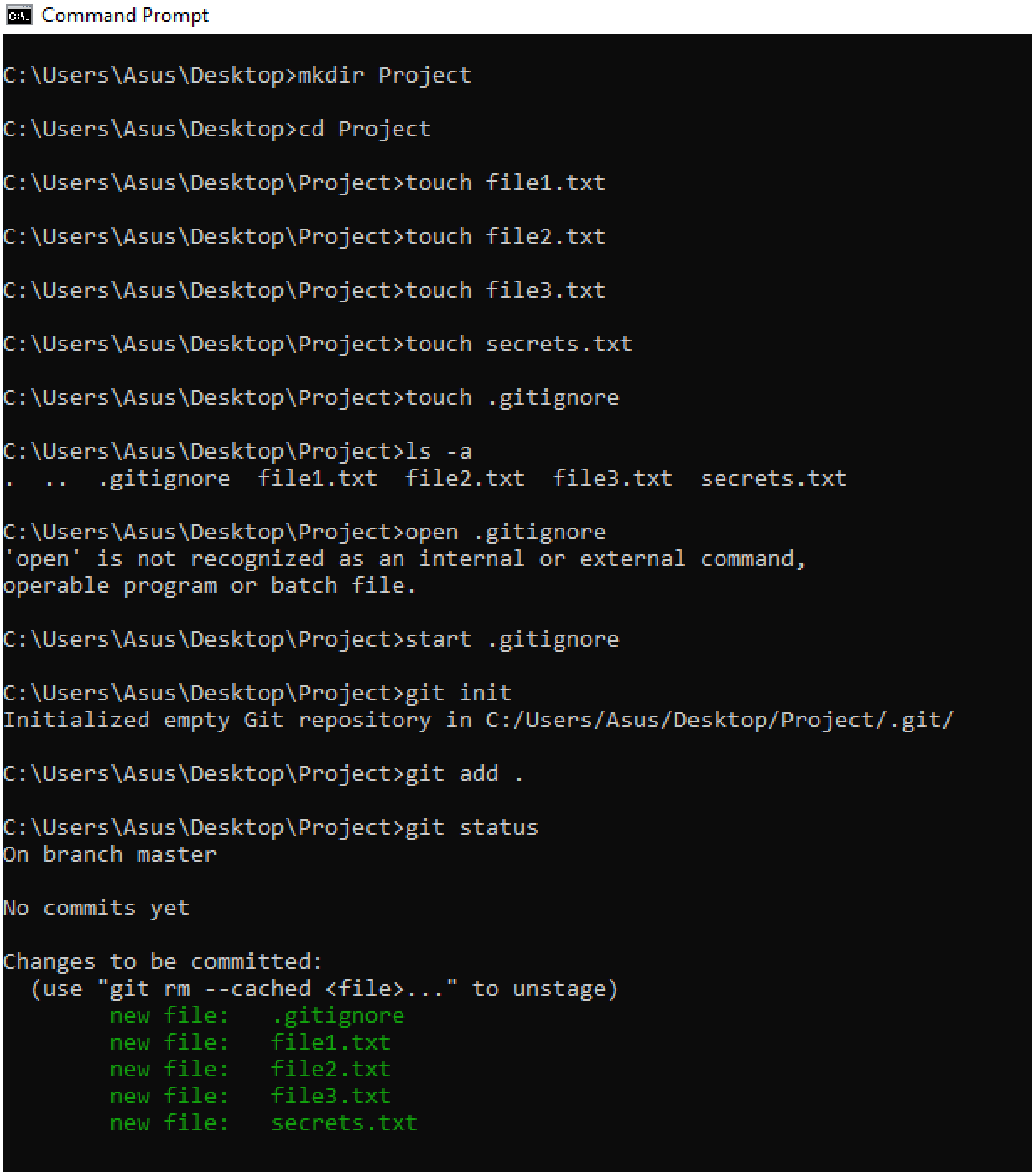
Command – `.gitignore`

(a hidden file). Inside the gitignore file right the name of the file you want to ignore:

ex: secrets.txt

usually node modules are ignored when publishing to an online server

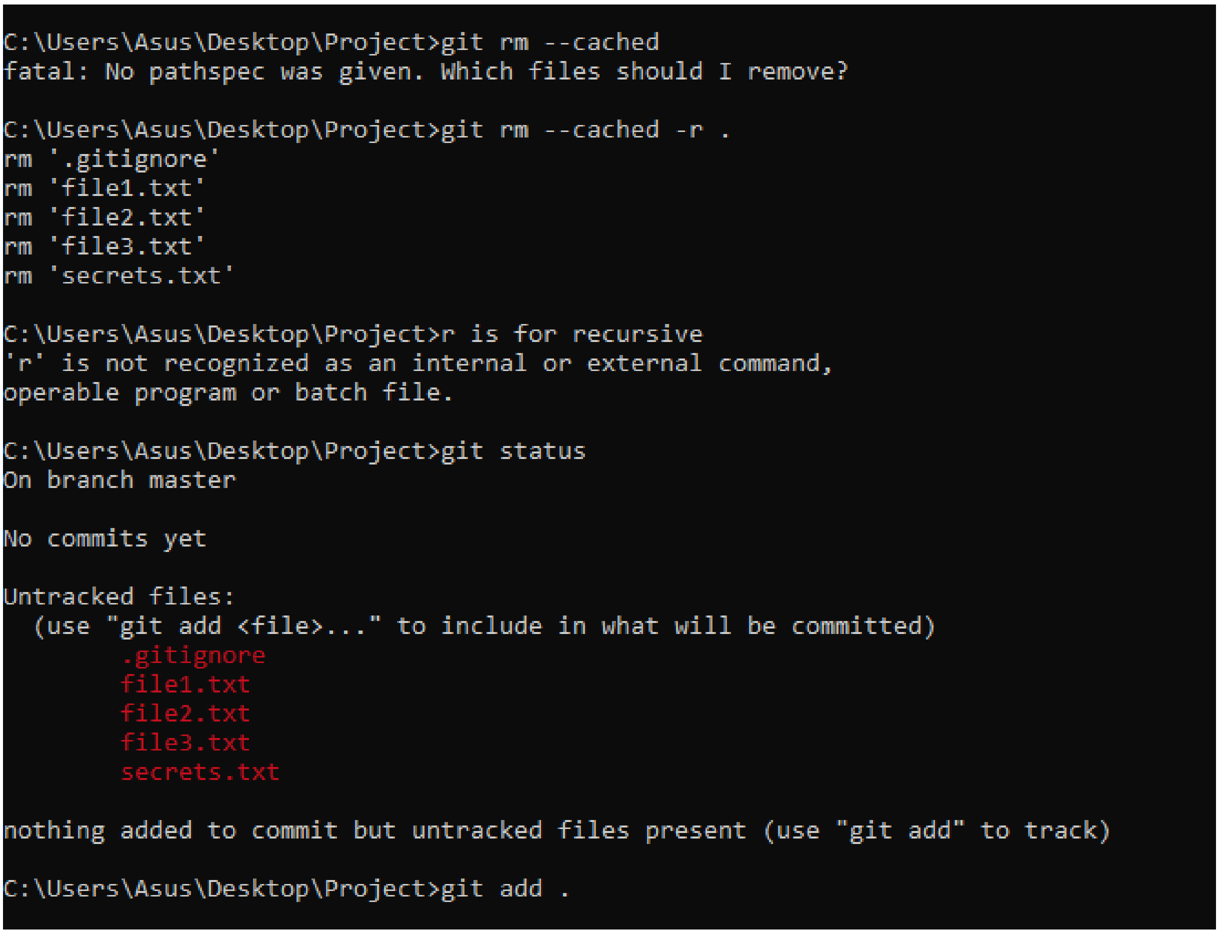
use # to add comments



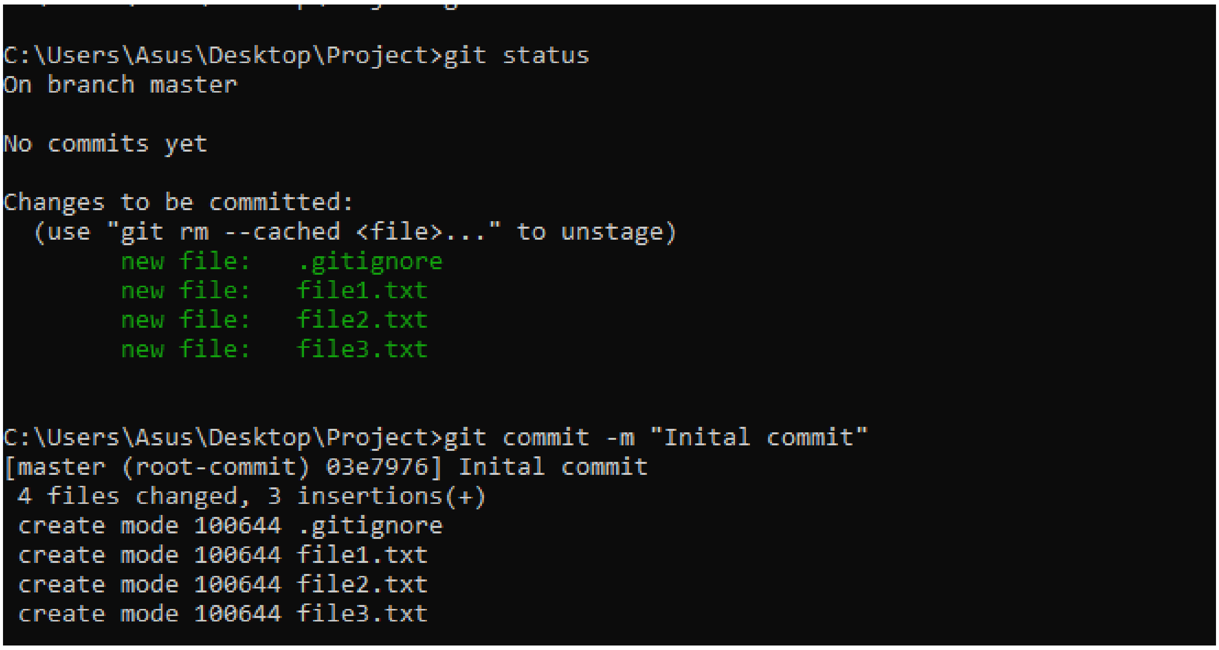
```
C:\Users\Asus\Desktop>mkdir Project

C:\Users\Asus\Desktop>cd Project

C:\Users\Asus\Desktop\Project>touch file1.txt

C:\Users\Asus\Desktop\Project>touch file2.txt

C:\Users\Asus\Desktop\Project>touch file3.txt

C:\Users\Asus\Desktop\Project>touch secrets.txt

C:\Users\Asus\Desktop\Project>touch .gitignore

C:\Users\Asus\Desktop\Project>ls -a
.  ..  .gitignore  file1.txt  file2.txt  file3.txt  secrets.txt

C:\Users\Asus\Desktop\Project>open .gitignore
'open' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Asus\Desktop\Project>start .gitignore

C:\Users\Asus\Desktop\Project>git init
Initialized empty Git repository in C:/Users/Asus/Desktop/Project/.git/

C:\Users\Asus\Desktop\Project>git add .

C:\Users\Asus\Desktop\Project>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt
        new file:   secrets.txt
```

to **undo the last step** in git – `git rm -cached -r file_name`

r is for recursive



```
C:\Users\Asus\Desktop\Project>git rm --cached
fatal: No pathspec was given. Which files should I remove?

C:\Users\Asus\Desktop\Project>git rm --cached -r .
rm '.gitignore'
rm 'file1.txt'
rm 'file2.txt'
rm 'file3.txt'
rm 'secrets.txt'

C:\Users\Asus\Desktop\Project>r is for recursive
'r' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Asus\Desktop\Project>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        file1.txt
        file2.txt
        file3.txt
        secrets.txt

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

C:\Users\Asus\Desktop\Project>git add .
```



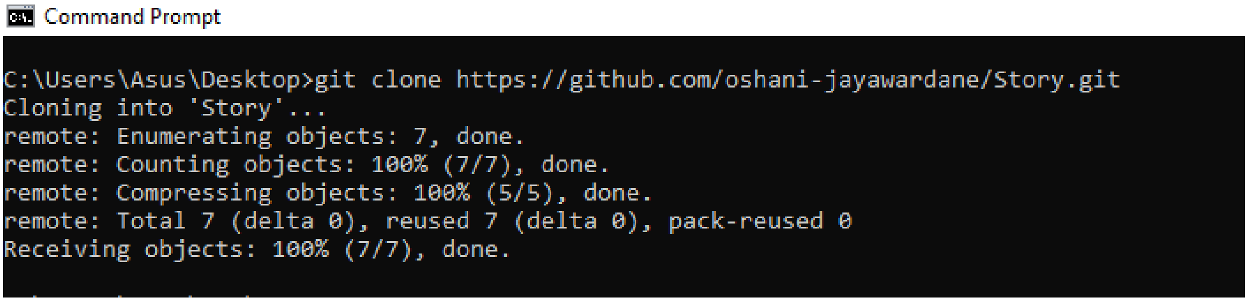
```
C:\Users\Asus\Desktop\Project>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   .gitignore
        new file:   file1.txt
        new file:   file2.txt
        new file:   file3.txt


C:\Users\Asus\Desktop\Project>git commit -m "Inital commit"
[master (root-commit) 03e7976] Inital commit
 4 files changed, 3 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 file1.txt
 create mode 100644 file2.txt
 create mode 100644 file3.txt
```

Cloning

Clone files from remote repository to working directory

In Command Prompt:

- Go to the directory you want to clone the file in
- `git clone (remote rep url)`

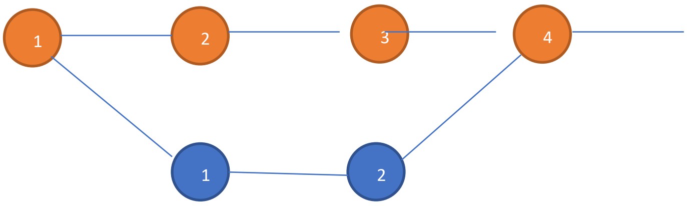


```
C:\Users\Asus\Desktop>git clone https://github.com/oshani-jayawardane/Story.git
Cloning into 'Story'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.
```

Branching and Merging

Developing experimental branches parallelly

- start with the main branch
- develop experimental branches parallelly if needed
- if the experimental branch seems to be fruitful, merge the branches later on



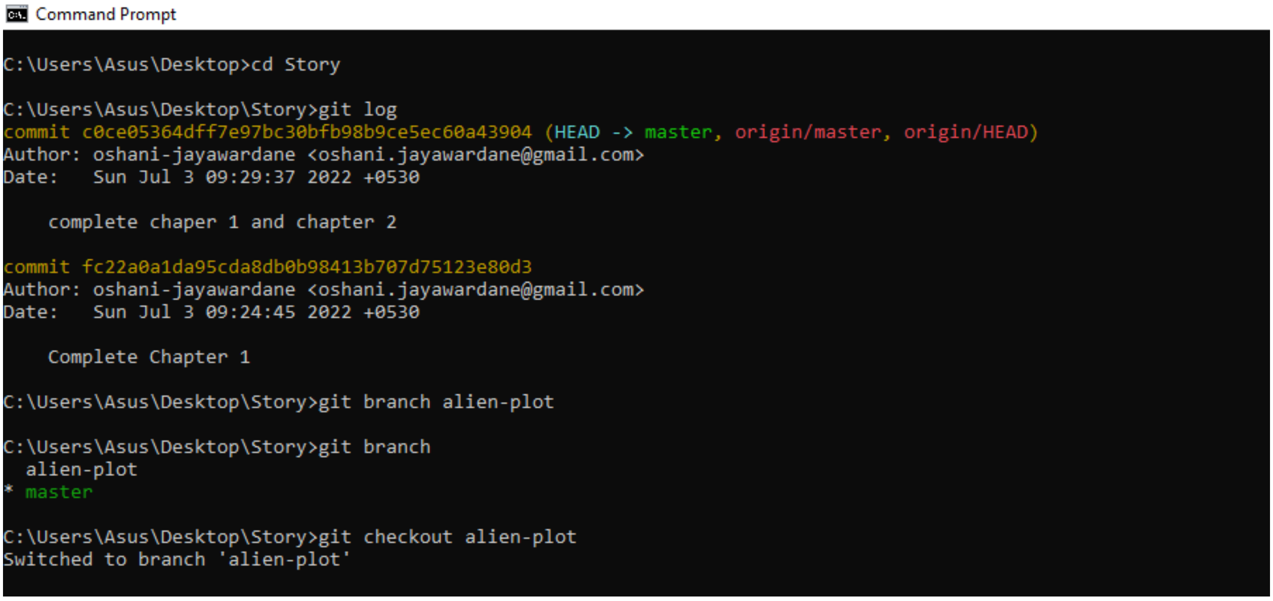
Ex: imagine **creating a new branch** to our story example as alien-branch

Command – `git branch (new_file_for_branch)`

Ex: `git branch alien-plot` to create a separate plot for the chapters in our stories project

View all branches – `git branch`

Branch with the * is the **current branch**

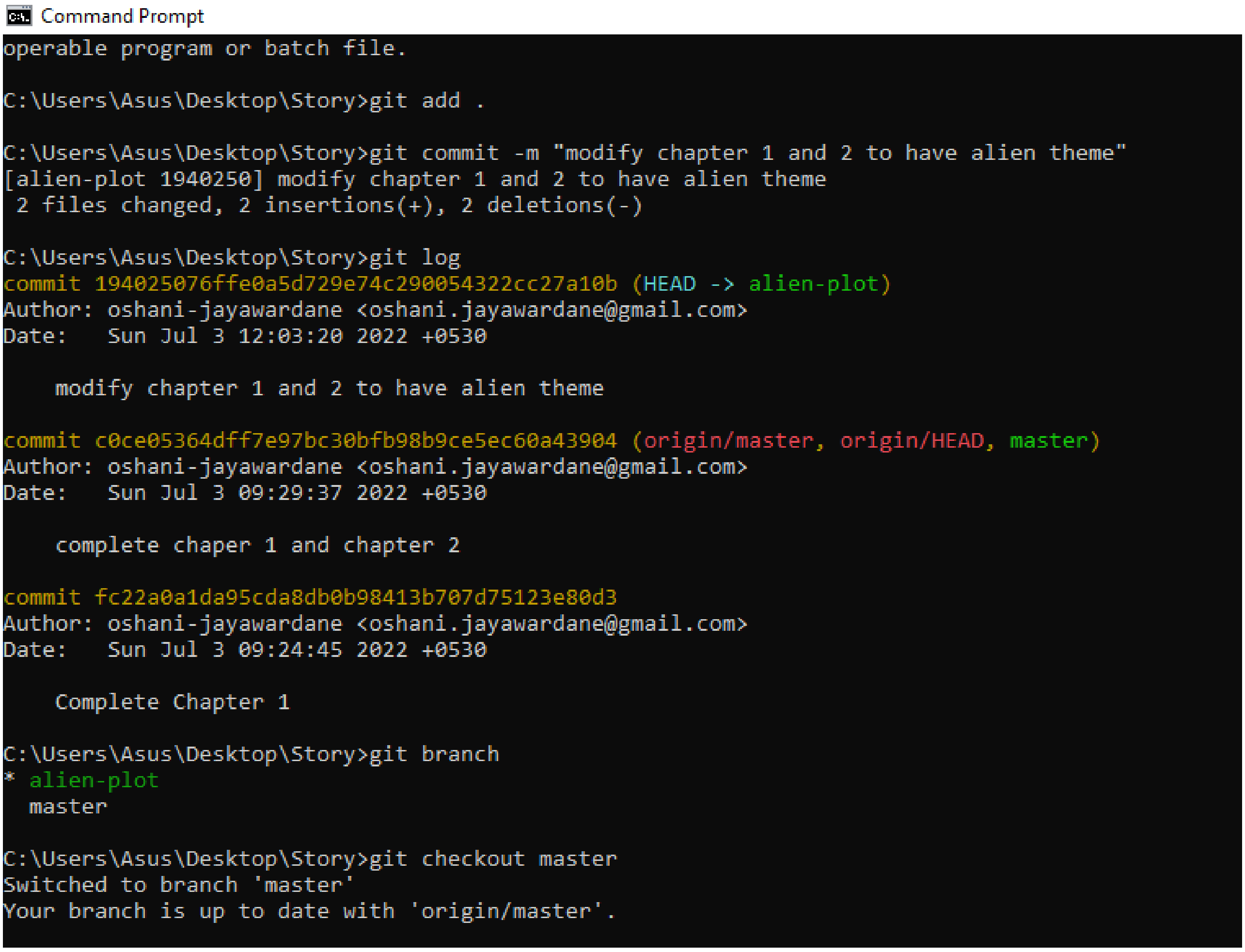


```
C:\Users\Asus\Desktop>cd Story

C:\Users\Asus\Desktop\Story>git log
commit c0ce05364dff7e97bc30bfb98b9ce5ec60a43904 (HEAD -> master, origin/master, origin/HEAD)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:29:37 2022 +0530

    complete chaper 1 and chapter 2

commit fc22a0a1da95cda8db0b98413b707d75123e80d3
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:24:45 2022 +0530

    Complete Chapter 1

C:\Users\Asus\Desktop\Story>git branch alien-plot

C:\Users\Asus\Desktop\Story>git branch
  alien-plot
* master

C:\Users\Asus\Desktop\Story>git checkout alien-plot
Switched to branch 'alien-plot'
```

Switch to required branch (alien-plot) – `git checkout alien-plot`

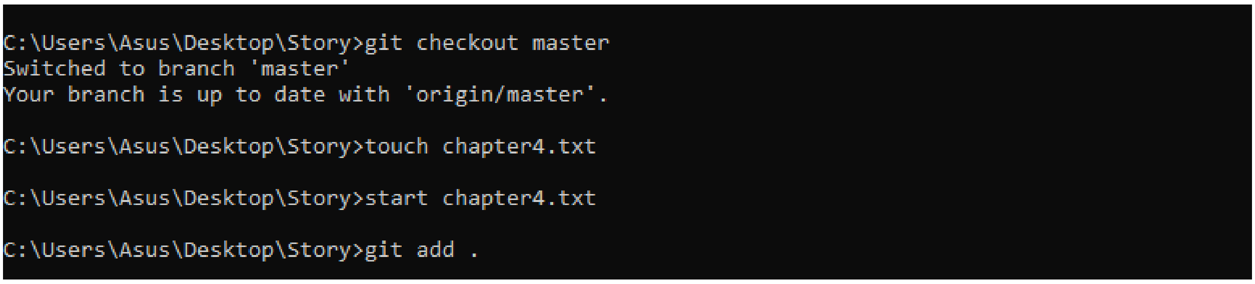
here we change all codes/text of files chapter1.txt, chapter2.txt, chapter3.txt

however, original scripts are intact. When we switch back branches and view the main branch the main branches content for chapter1.txt, chapter2.txt, chapter3.txt will be different from those of alien-plot branch

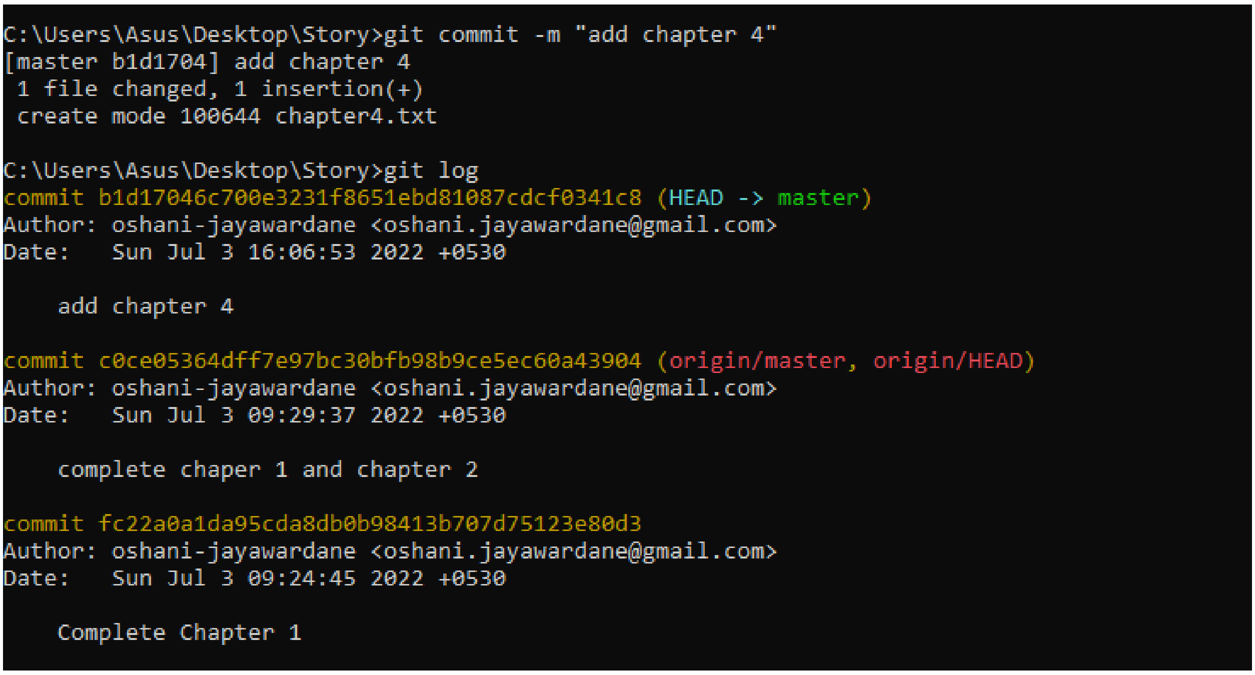


```
operable program or batch file.

C:\Users\Asus\Desktop\Story>git add .

C:\Users\Asus\Desktop\Story>git commit -m "modify chapter 1 and 2 to have alien theme"
[alien-plot 1940250] modify chapter 1 and 2 to have alien theme
 2 files changed, 2 insertions(+), 2 deletions(-)

C:\Users\Asus\Desktop\Story>git log
commit 194025076ffe0a5d729e74c290054322cc27a10b (HEAD -> alien-plot)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 12:03:20 2022 +0530

    modify chapter 1 and 2 to have alien theme

commit c0ce05364dff7e97bc30bfb98b9ce5ec60a43904 (origin/master, origin/HEAD, master)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:29:37 2022 +0530

    complete chaper 1 and chapter 2

commit fc22a0a1da95cda8db0b98413b707d75123e80d3
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:24:45 2022 +0530

    Complete Chapter 1

C:\Users\Asus\Desktop\Story>git branch
* alien-plot
  master

C:\Users\Asus\Desktop\Story>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

Add a chapter4.txt to master branch



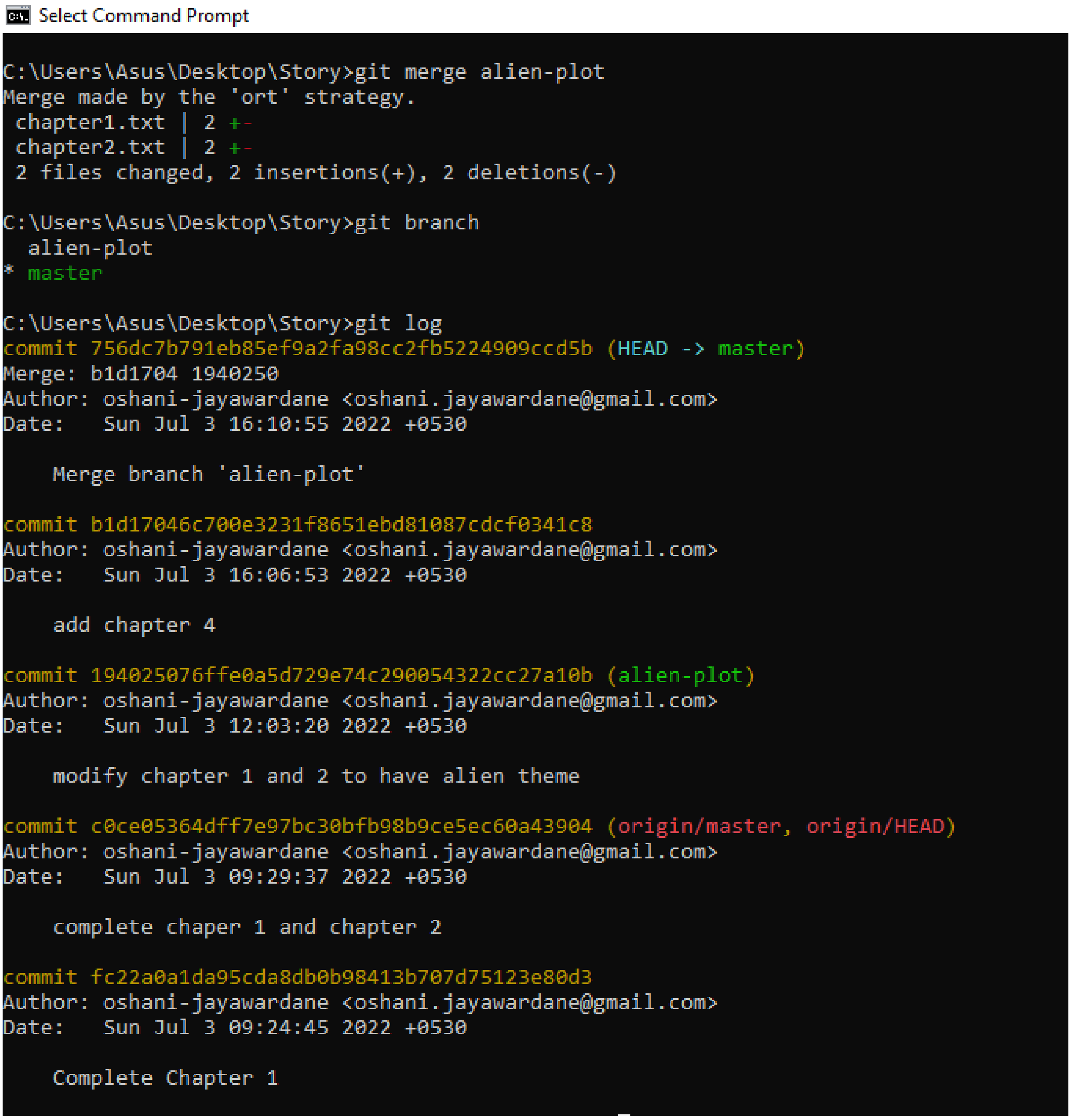
```
C:\Users\Asus\Desktop\Story>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

C:\Users\Asus\Desktop\Story>touch chapter4.txt

C:\Users\Asus\Desktop\Story>start chapter4.txt

C:\Users\Asus\Desktop\Story>git add .
```



```
C:\Users\Asus\Desktop\Story>git commit -m "add chapter 4"
[master b1d1704] add chapter 4
 1 file changed, 1 insertion(+)
 create mode 100644 chapter4.txt

C:\Users\Asus\Desktop\Story>git log
commit b1d17046c700e3231f8651ebd81087cdcf0341c8 (HEAD -> master)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 16:06:53 2022 +0530

    add chapter 4

commit c0ce05364dff7e97bc30bfb98b9ce5ec60a43904 (origin/master, origin/HEAD)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:29:37 2022 +0530

    complete chaper 1 and chapter 2

commit fc22a0a1da95cda8db0b98413b707d75123e80d3
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:24:45 2022 +0530

    Complete Chapter 1
```

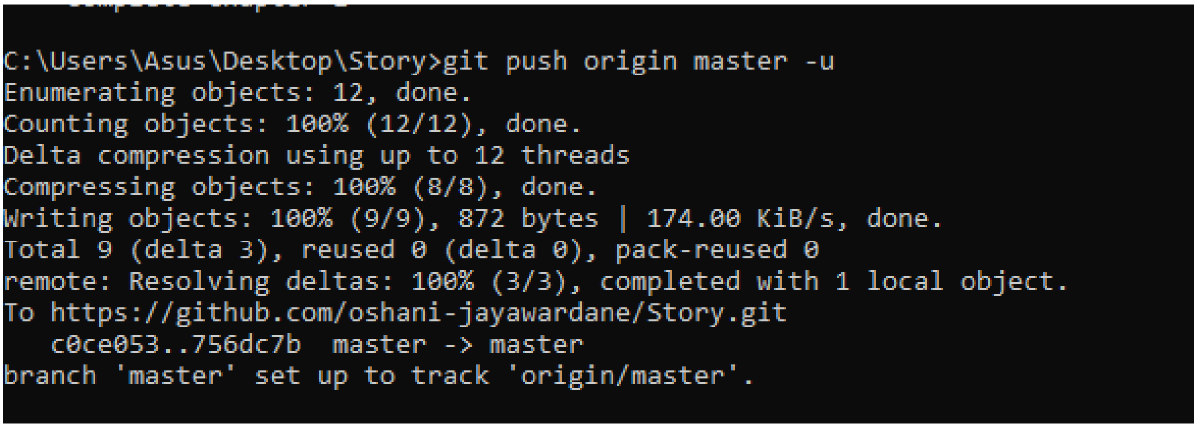
merging the experimented branch ‘alien-plot’ with the master branch, cause the results were satisfactory.

Before merging we have to make sure we are in the master branch



```
C:\Users\Asus\Desktop\Story>git merge alien-plot
Merge made by the 'ort' strategy.
 chapter1.txt | 2 +-
 chapter2.txt | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)

C:\Users\Asus\Desktop\Story>git branch
  alien-plot
* master

C:\Users\Asus\Desktop\Story>git log
commit 756dc7b791eb85ef9a2fa98cc2fb5224909ccd5b (HEAD -> master)
Merge: b1d1704 1940250
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 16:10:55 2022 +0530

    Merge branch 'alien-plot'

commit b1d17046c700e3231f8651ebd81087cdcf0341c8
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 16:06:53 2022 +0530

    add chapter 4

commit 194025076ffe0a5d729e74c290054322cc27a10b (alien-plot)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 12:03:20 2022 +0530

    modify chapter 1 and 2 to have alien theme

commit c0ce05364dff7e97bc30bfb98b9ce5ec60a43904 (origin/master, origin/HEAD)
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:29:37 2022 +0530

    complete chaper 1 and chapter 2

commit fc22a0a1da95cda8db0b98413b707d75123e80d3
Author: oshani-jayawardane <oshani.jayawardane@gmail.com>
Date:   Sun Jul 3 09:24:45 2022 +0530

    Complete Chapter 1
```

Pushing changes to GitHub



```
C:\Users\Asus\Desktop\Story>git push origin master -u
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 872 bytes | 174.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/oshani-jayawardane/Story.git
   c0ce053..756dc7b  master -> master
branch 'master' set up to track 'origin/master'.
```

Now there are new commits on GitHub.

Go to insights -> network

It indicates the new process with the branch and the merge

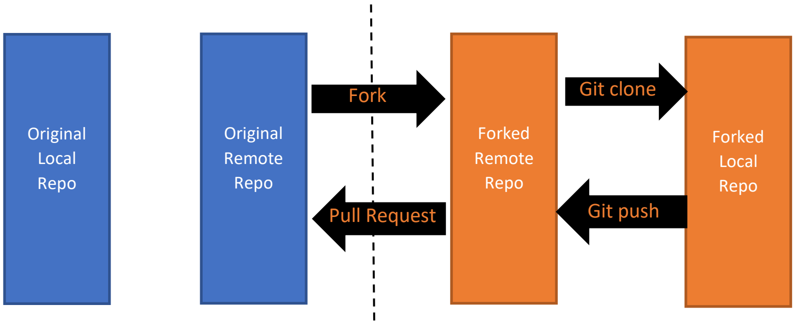
This process can also be done inside GitHub itself

- create a repository
- add files using create a new file and code in itself/ upload
- Create a new branch 'experimental' from master
- This creates a copy of the master in experimental
- Now we can edit experimental separately without effecting the master
- Check network insights
- To merge experimental to master branch go back to master branch page. Click new pull request
- Compare master branch with experimental (red-erasure, plus and green - addition)
- Once done create a pull request
- Merge
- Check the network graph

Forking and Pull Requests

Forking – create a copy of the remote repository in someone else's own working directory. Now that someone has complete ownership of the copy. (Different from cloning). Only reading

Cloning – both read and write



Pull request is like a suggestion, and permission to corporate

If the other person's edit seems better, we can accept the pull request and merge the files to our repo.

This can be done inside GitHub

Quick Notes

When cloning repos from GitHub, they usually don’t have the node_modules folder. But they have specified dependencies in the packages.json.

In the cmd, locate to project folder and type `npm install`. This will install all the dependencies at once