Intro to GIT.

- Git is a distributed version control system.
- Its features are open source,scalable,speed and efficiency.
- **Distributed Version Control System**: Git has a remote repository which is stored in a server and a local repository which is stored in the computer of each developer.
- This means that the code is not just stored in a central server, but the full copy of the code is present in all the developers' computers.
- Git is a Distributed Version Control System since the code is present in every developer's computer.

mkdir (filename) - to create a directory.
cd (filename) - to change directory.

touch <filename> - to create a new file.

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ touch four.txt
```

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ ls
GitPractice/  four.txt  one.txt.txt  three.txt.txt  two.txt.txt
```

We can see that four.txt is added and displayed here.

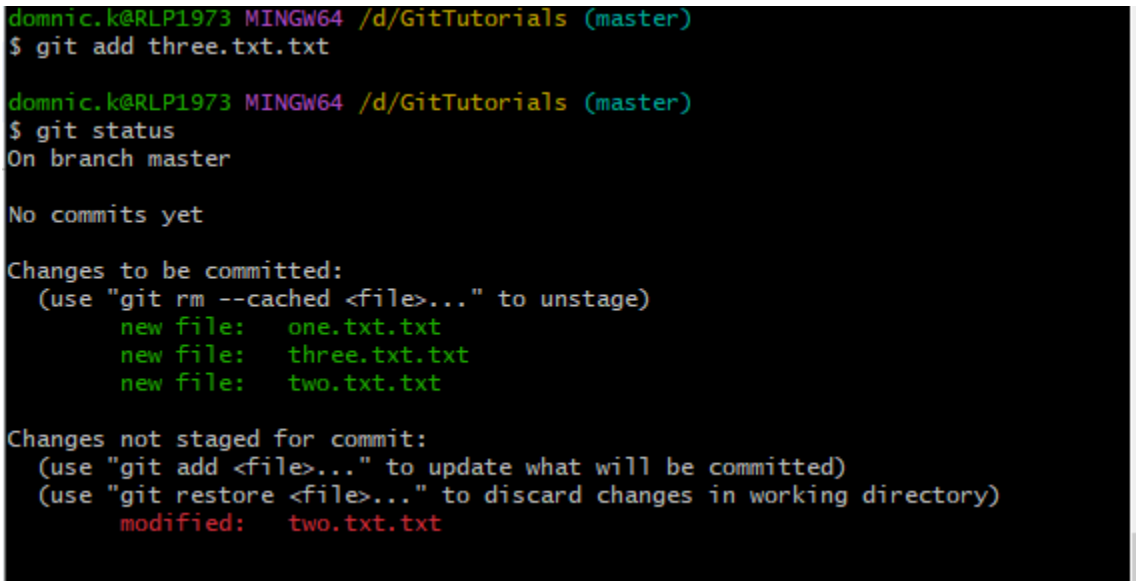
git init - it will create an empty git repository.

```
domnic.k@RLP1973 MINGW64 ~
$ pwd
/c/Users/domnic.k

domnic.k@RLP1973 MINGW64 ~
$ cd D:\initExample

domnic.k@RLP1973 MINGW64 /d/initExample
$ git init
Initialized empty Git repository in D:/initExample/.git/

domnic.k@RLP1973 MINGW64 /d/initExample (master)
```

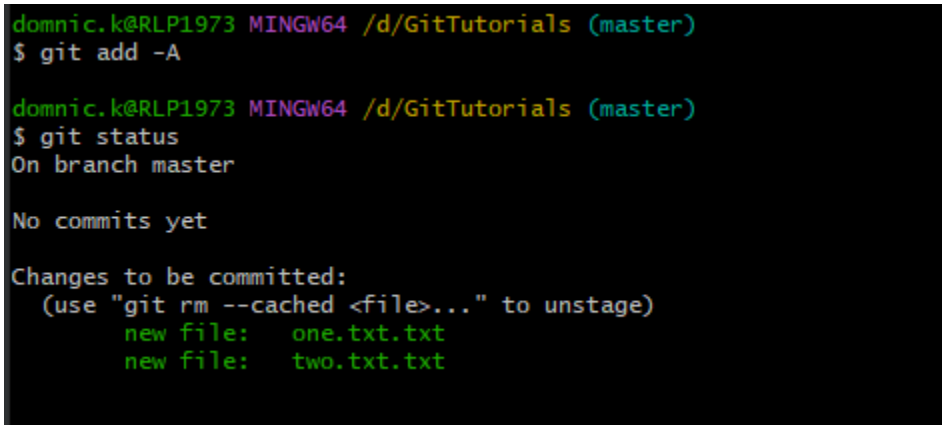
There are three stages in it.
1.Working directory
2.Staging area
3.Git repository.

Committing and staging:
Committing is the process of adding code in the local repository.before committing the code has to be in the **staging area.**
Staging area- It will track which are all the files that need to be committed.

git add (filename) - move the file from working directory to the staging area.



git add -A - it will add all the files including hidden files.



git add -A and **git add--all** both are same

git status - it will display the state of working directory and staging area.

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git status
On branch master

No commits yet

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

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

git commit -m"message here" - to move file from staging area to repository.

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git commit -m "Added 1 file"
[master (root-commit) 1566a7d] Added 1 file
 Committer: Domnic Dev Kulanthaisamy <domnic.k@itp.objectfrontier.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 3 files changed, 2 insertions(+)
 create mode 100644 one.txt.txt
 create mode 100644 three.txt.txt
 create mode 100644 two.txt.txt
```

git log - to show the commits which have been made until now.

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git log
commit 0d2d68ba5d6216244672326d92e5206273a7358f (HEAD -> master)
Author: Domnic Dev Kulanthaisamy <domnic.k@itp.objectfrontier.com>
Date:   Wed Aug 9 17:28:21 2023 +0530

    Add one file

commit 1566a7d12ceae203c238bdc18ce95fff35d275f1
Author: Domnic Dev Kulanthaisamy <domnic.k@itp.objectfrontier.com>
Date:   Wed Aug 9 16:38:24 2023 +0530

    Added 1 file
```

git rm <filename> - to delete the file.

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git rm two.txt.txt
rm 'two.txt.txt'

domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    two.txt.txt
```

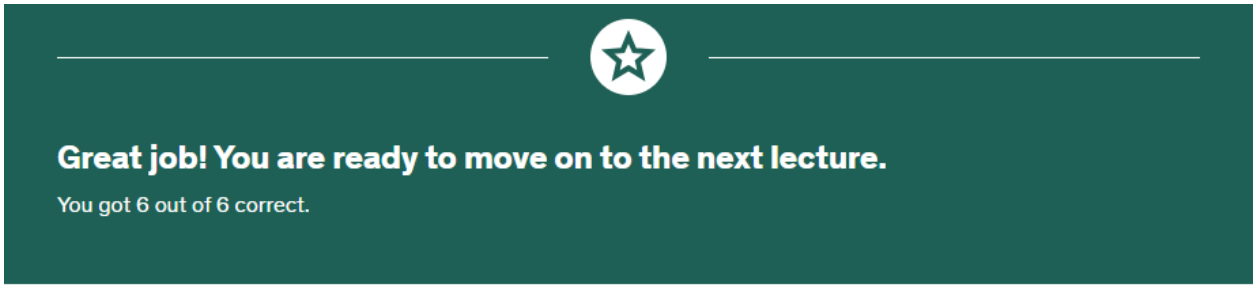
git diff <filename> - we can see what was changed in the recent file

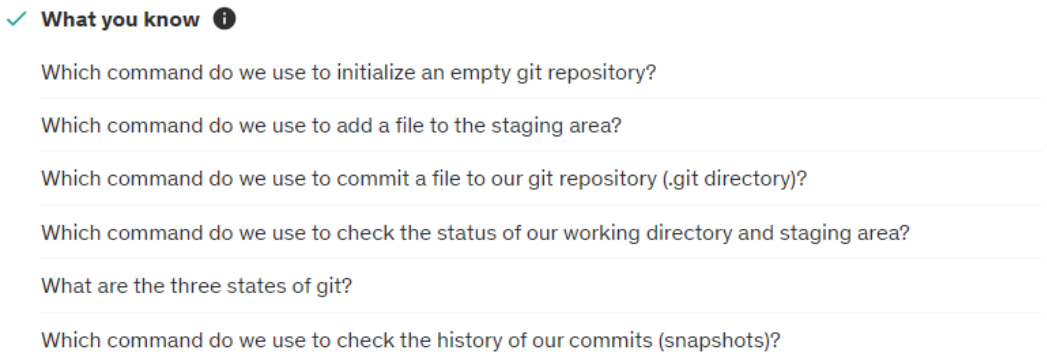
```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git diff two.txt.txt
diff --git a/two.txt.txt b/two.txt.txt
index b7a3064..f58b60e 100644
--- a/two.txt.txt
+++ b/two.txt.txt
@@ -1 +1,5 @@
-Hello Good morning
\ No newline at end of file
+Hello Good morning
+
+Lets begin
+
+Good Evening
\ No newline at end of file
```

The previous text will be displayed in Red color,And the changed text will be green color.

git reset --hard - to undo all the changes.

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git reset --hard
HEAD is now at 1566a7d Added 1 file
```





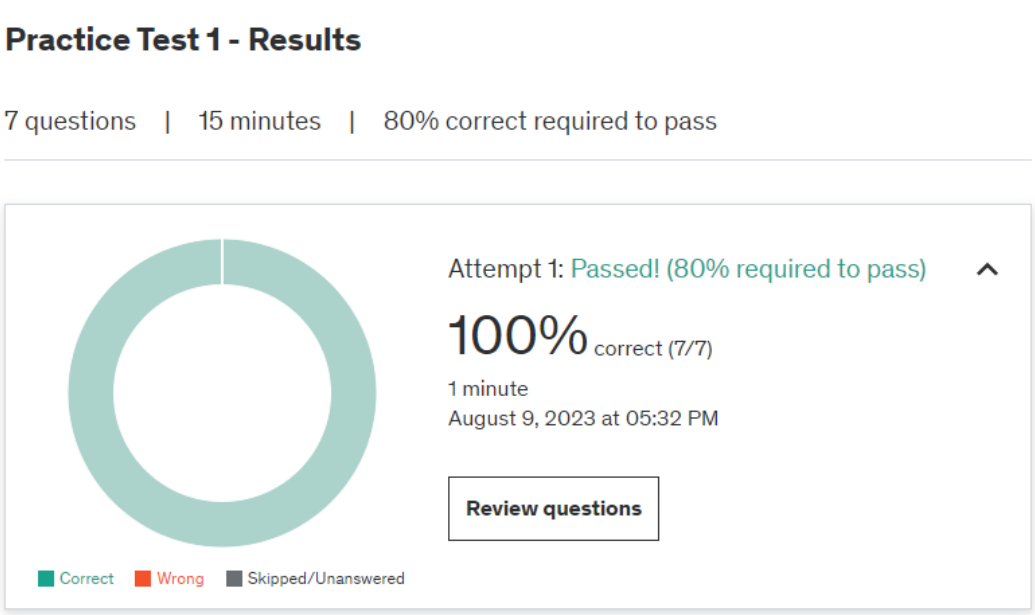
git reset Head - reset to staging area from local repository.

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git reset Head
Unstaged changes after reset:
M       two.txt.txt
```

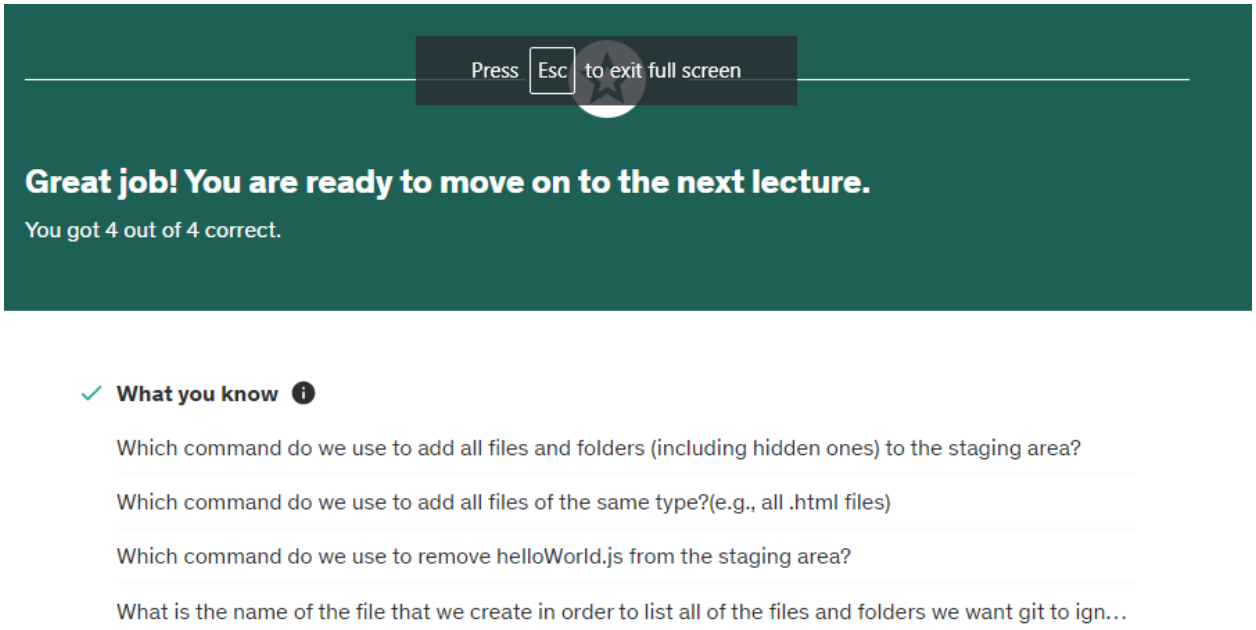
git reset <filename> - it will undo the changes that were made in the repository.

```
domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git reset

domnic.k@RLP1973 MINGW64 /d/GitTutorials (master)
$ git status
On branch master

No commits yet

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

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

git add*.html - to add all files of the same type.



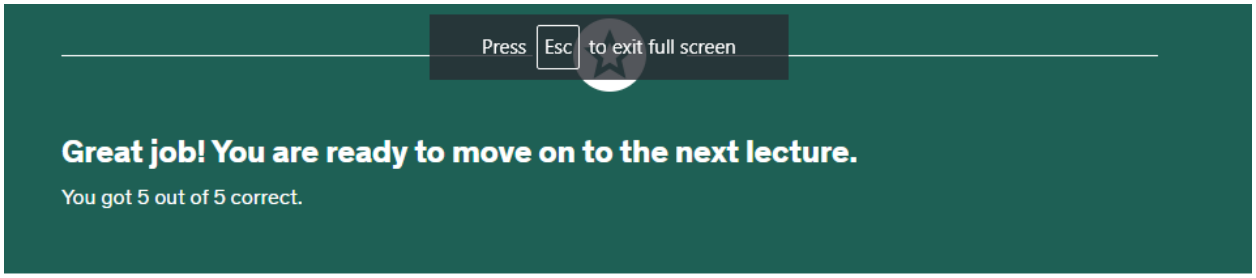
For Ignoring files - By creating a hidden file called **.gitignore** and adding the files need to be ignored in .gitignore.



GIT Branches:

- A branch represents an independent line of development

git branch - to display all the branches.
git checkout -b<branchname> - it will add a new branch.
git checkout<branch name> - to go to the previous branch(navigate between branches).
git merge <branch name> - to merge two different branches.
git branch -d<branch name> - to remove the branch.



✓ **What you know**

Which command do we use to list our branches?

Which command do we use to add and checkout a new branch, simultaneously?

Which command do we use to switch to another branch?

What are the steps for merging a feature branch into master?

Which command do we use to remove an unwanted branch?

Well done!

You've successfully completed the Intro to Git course.