Did you GIT it already?

Now that you've installed all the necessary requirements on your computer including the GIT bash and are curious on how to set up the GIT repositories, use the commands like a tech-savvy person, you've reached the right place to look for.In this blog we shall cover the basics.

Firstly, check for GIT on your computer. If you're a windows user check for the git bash and if you're a mac user look for the terminal and open them. Type the command "git version".The output will either tell you which version of Git is installed, or it will alert you that git is an unknown command.If it shows an unknown command read the blog on get started with GIT.

Before you jump into a plethora of applications, know that there are three different configuration levels. They are:

1. System:System-level covers an entire user, entire machine and all repos.
2. Global:Global level configuration is the level where editing in this level affects the user level.
3. Repository: Repository level is the level which is specific to the repository.

With the system being the highest level and the repository being the lowest among them.

Now that you know all the basics let's get started with the configuration.

1. Check version:

```
git --version
```

2. Setting git with your name:

```
git config --global user.name "<Your-Full-Name>"
```

3. Setting git with your email:

```
git config --global user.email "<your-email-address>"
```

4. Current configurations type:

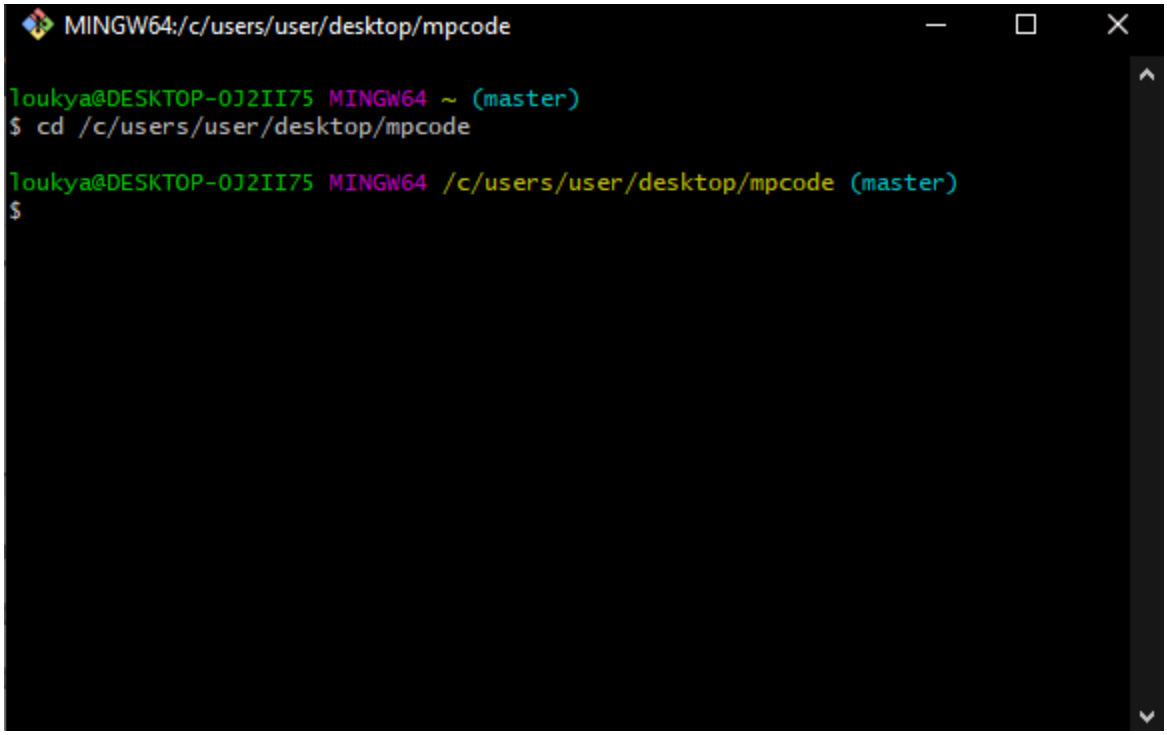
```
git config --list
```

Initializing a repository:

Let us begin with initializing a repository in an existing repository.

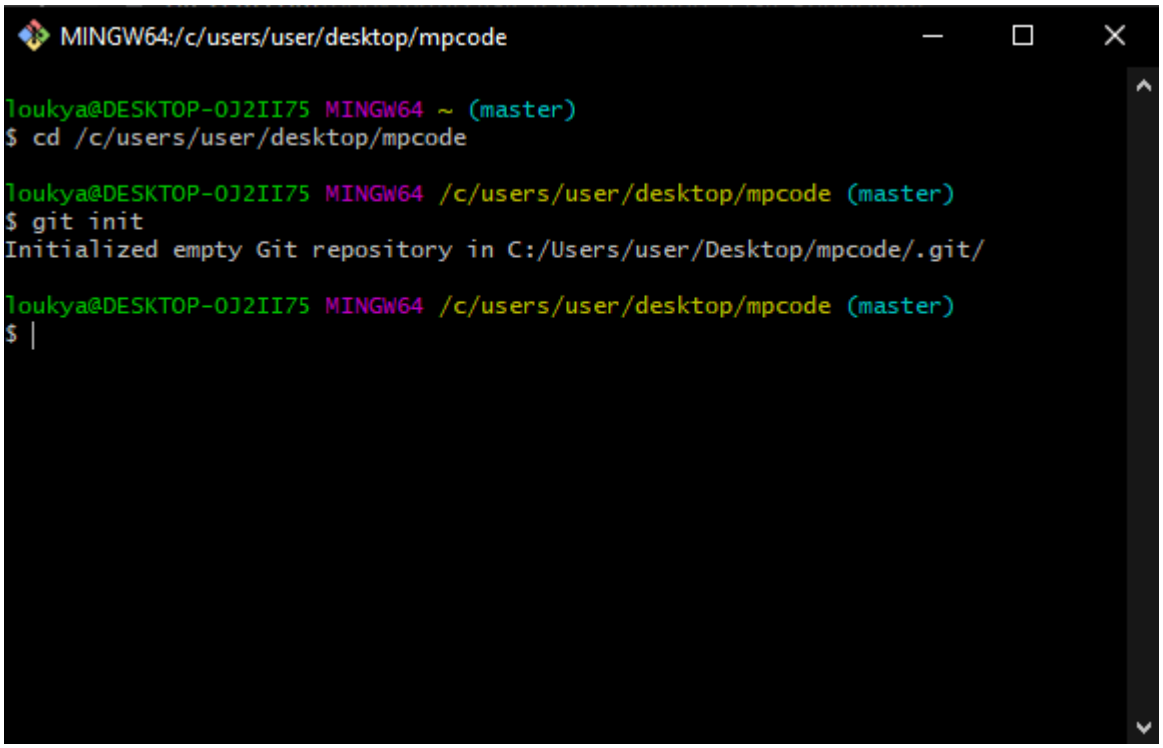
STEP 1:

Navigate to the directory using the command $ cd '/c/Users/User/Desktop/mpcode'



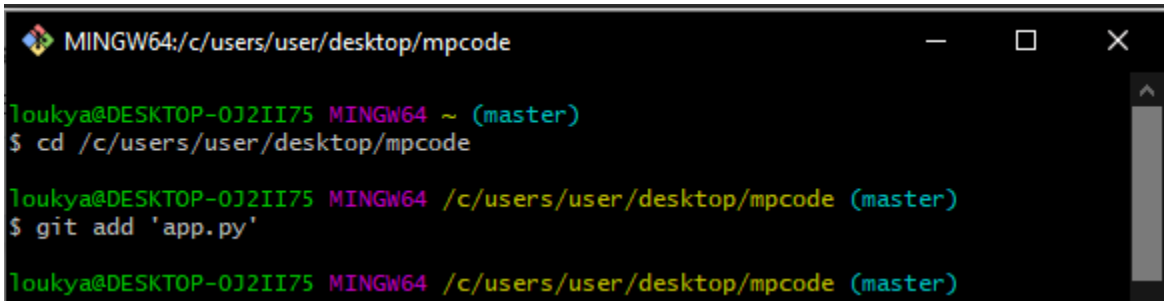
STEP 2:

Use the **git init** command to create a new Git repository.



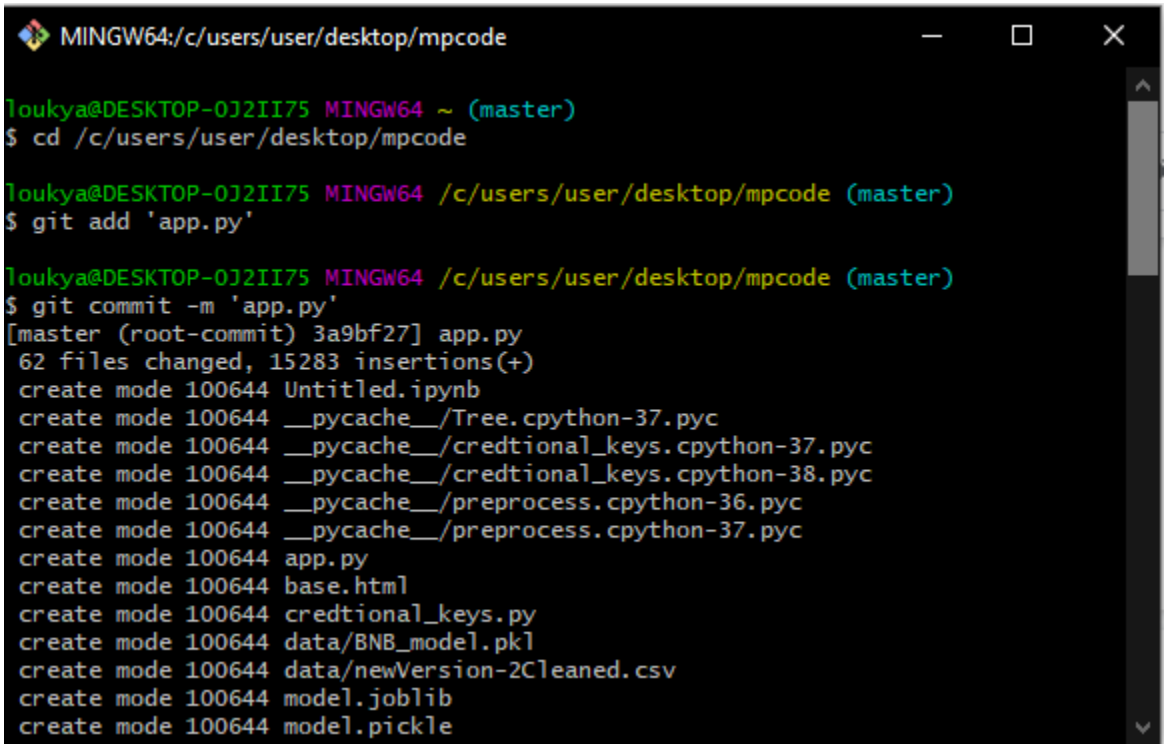
STEP 3:

Use the **git add command** to add a file to the Git staging area.



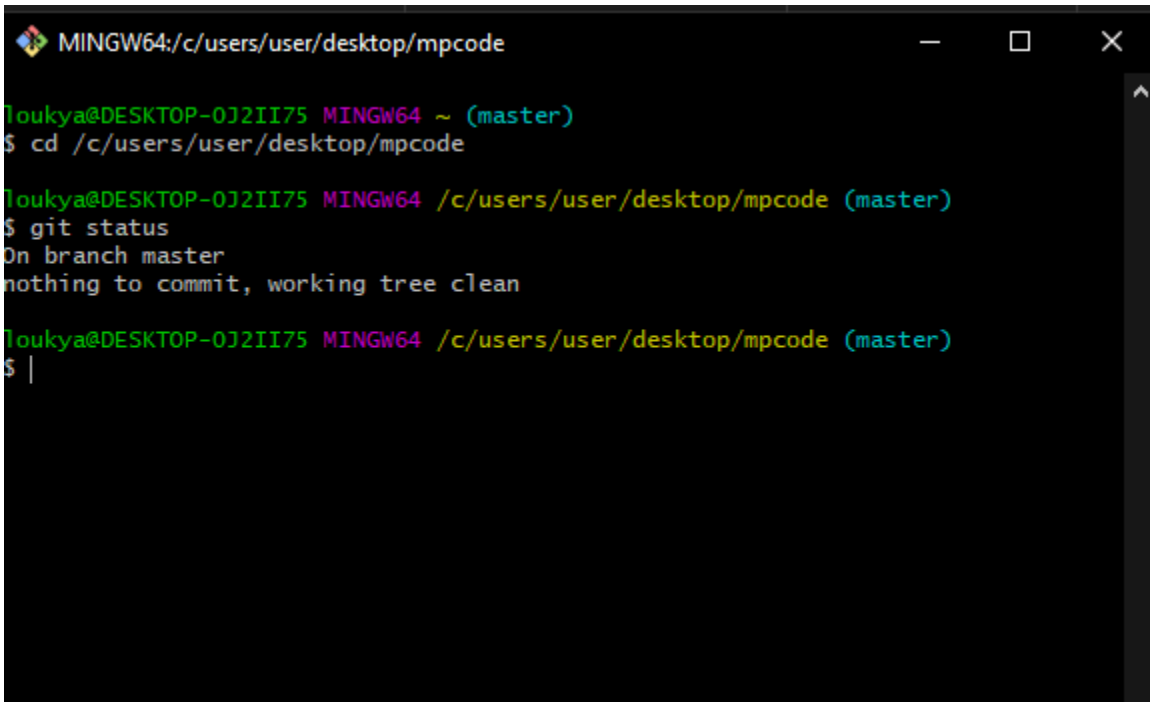
STEP 4:

Use the **git commit** command to save your changes to the local repository.



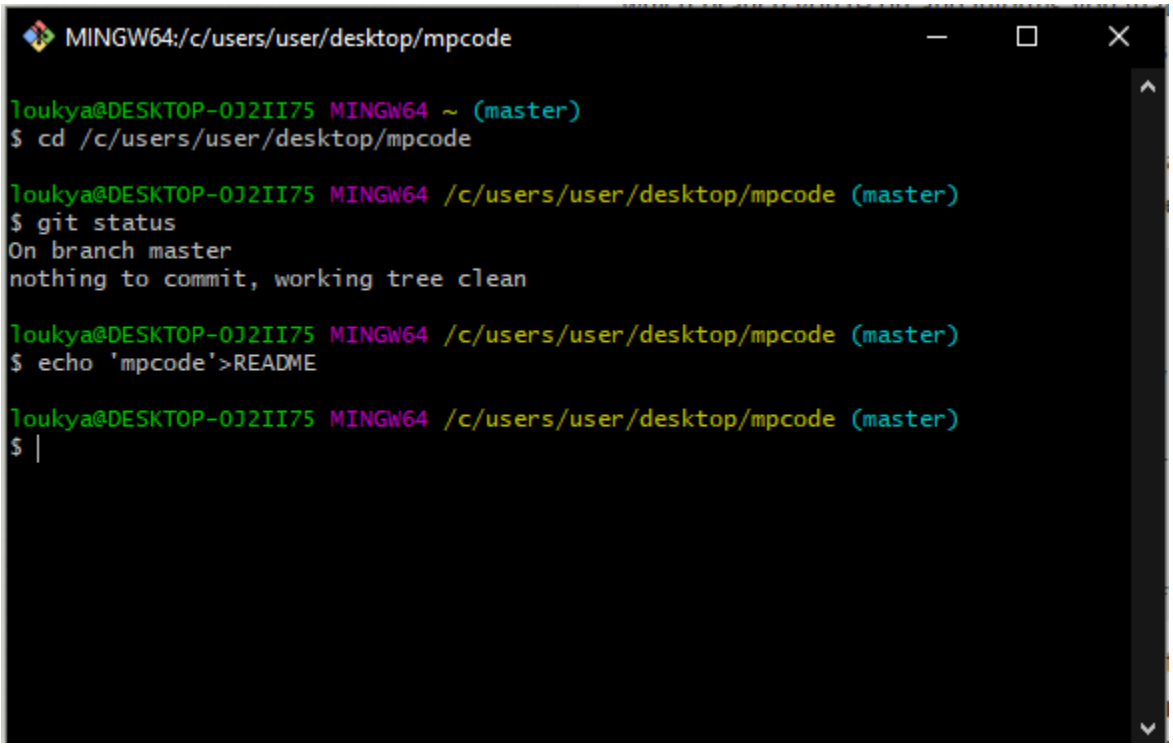
STEP 5:

Use the **git status** command to display the state of the working directory and the staging area.



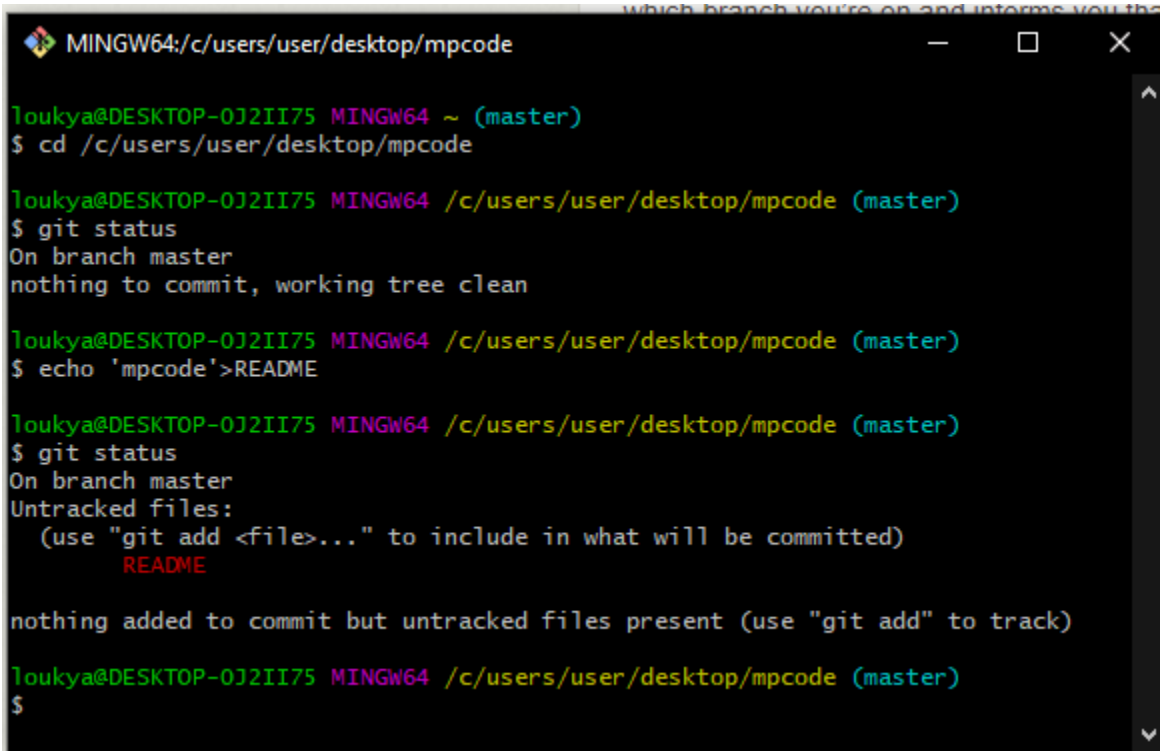
STEP 6:

Use the command **echo 'filename'>newfile** to add a new file.



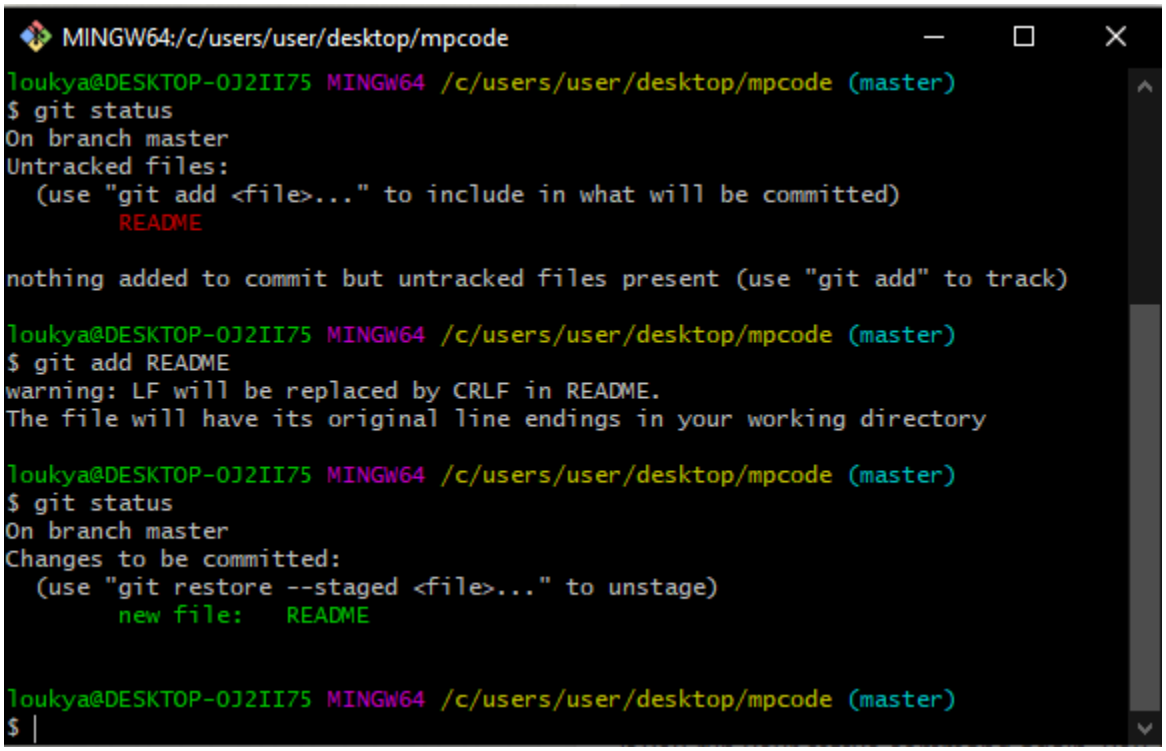
STEP 7:

Now check the status again. You can observe that there are some untracked files.



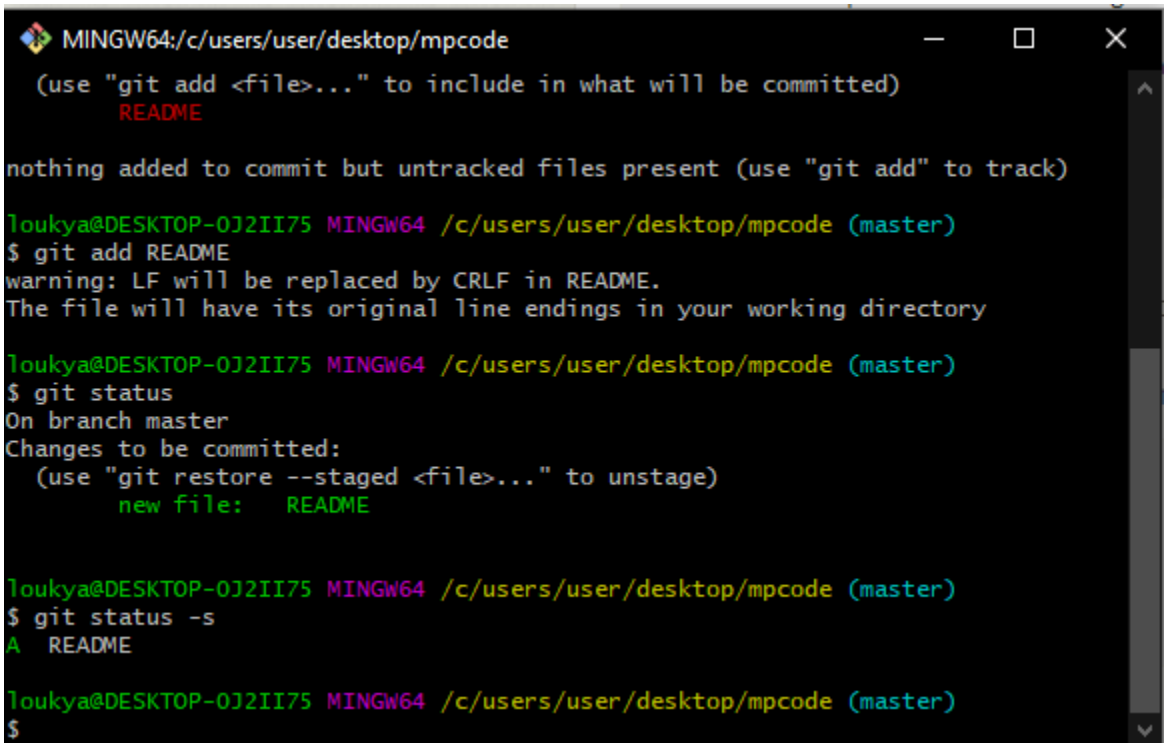
STEP 8:

Now add the previous file to the git staging area.


MINGW64:/c/users/user/desktop/mpcode
loukya@DESKTOP-0J2II75 MINGW64 /c/users/user/desktop/mpcode (master)
$ git status
On branch master
Untracked files:
(use "git add <file>..." to include in what will be committed)
README
nothing added to commit but untracked files present (use "git add" to track)
loukya@DESKTOP-0J2II75 MINGW64 /c/users/user/desktop/mpcode (master)
$ git add README
warning: LF will be replaced by CRLF in README.
The file will have its original line endings in your working directory
loukya@DESKTOP-0J2II75 MINGW64 /c/users/user/desktop/mpcode (master)
$ git status
On branch master
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
new file: README
loukya@DESKTOP-0J2II75 MINGW64 /c/users/user/desktop/mpcode (master)
$

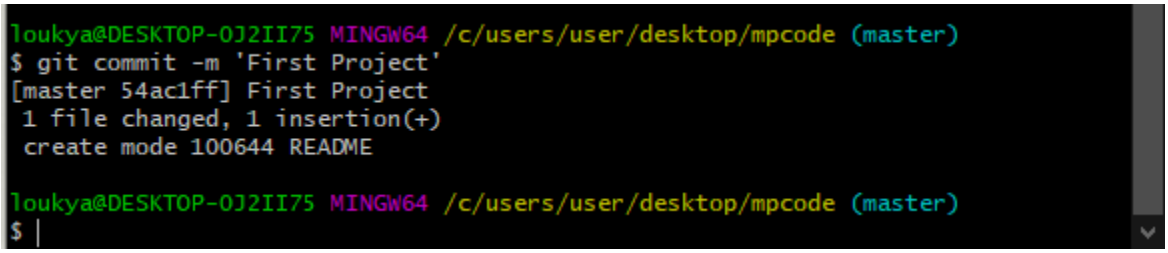
STEP 9:

Check the status again to see changes. You can observe that you can now see that the file has been added to the staging area and is ready to be committed.

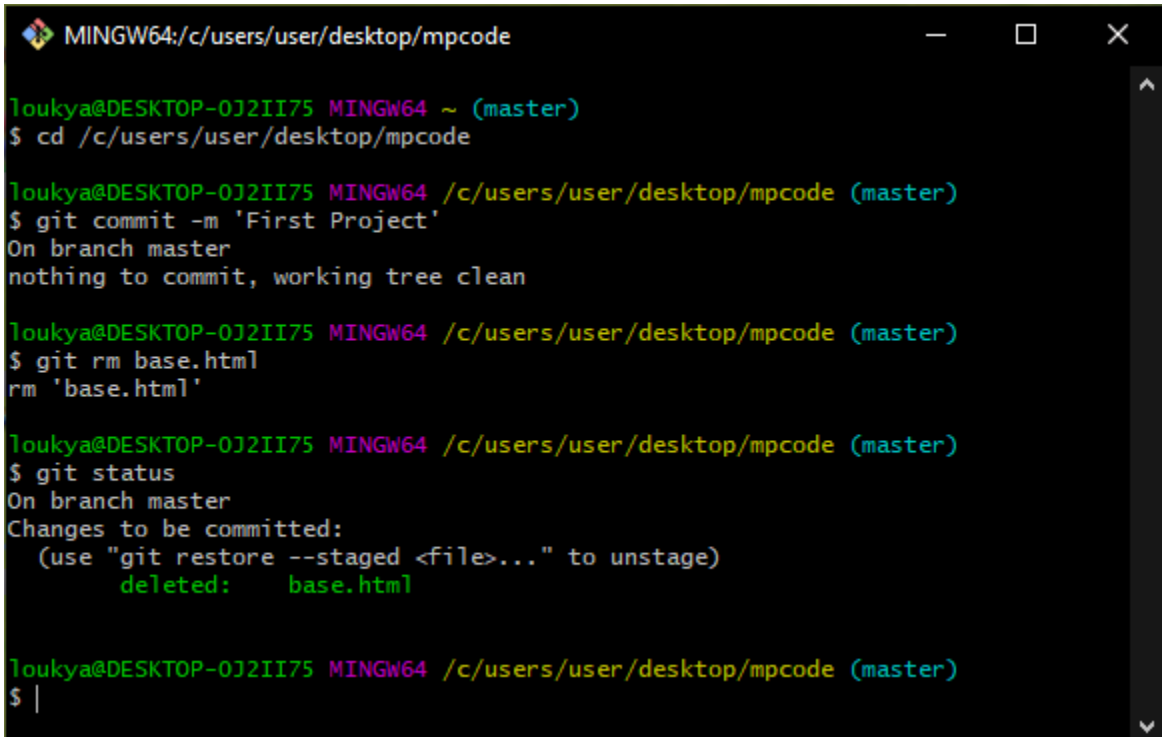


STEP 10:

Now use the **git commit -m 'message'** command to commit with a message.



STEP 11:
If you want to remove a file from the repository use the command **git rm 'filename'**



In this blog we have covered:

- Checking version
- Setting git with your name
- Setting git with your email
- Checking current configuration type
- Initializing a repository
- Using various commands to configure the repository.

To know more about the push and pull requests please refer to our other blog [GIT push: Backup from local to cloud](#).