NAME – Shipra Pal

REGISTRATION NO. – 2141016019

TASK: Create a tutorial for Git and GitHub with the Commands.

First Install Git

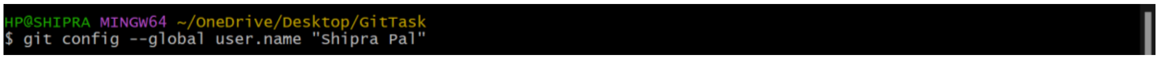
- **Step1** -Go to google and type Git and then click to the first link and download Git for mac, Linux or windows.
- **Step 2**-Then after that open Command shell to start the Git.
- **Step 3**- To know the Git version, we can type **$ git --version**.
- **Step 4** – Create a Git Folder and open the folder in Git Bash to go to the current directory

Now follow the following commands:

1. Git Setup

- Sets your Git username.
- **--global** makes it apply to all your repositories on your system.

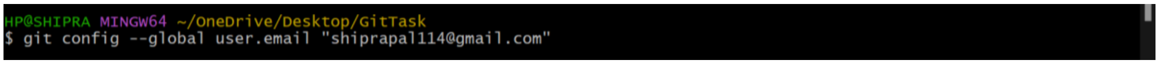
git config --global user.name "Name"



```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask
$ git config --global user.name "Shipra Pal"
```

- Sets your Git email globally. This is important for associating commits with your GitHub account.

git config --global user. Email “Email”

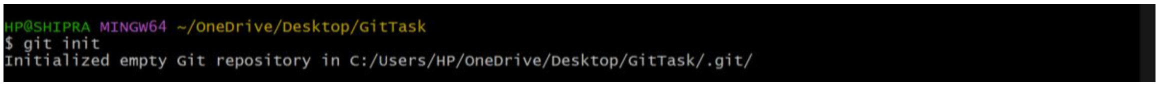


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask
$ git config --global user.email "shiprapal114@gmail.com"
```

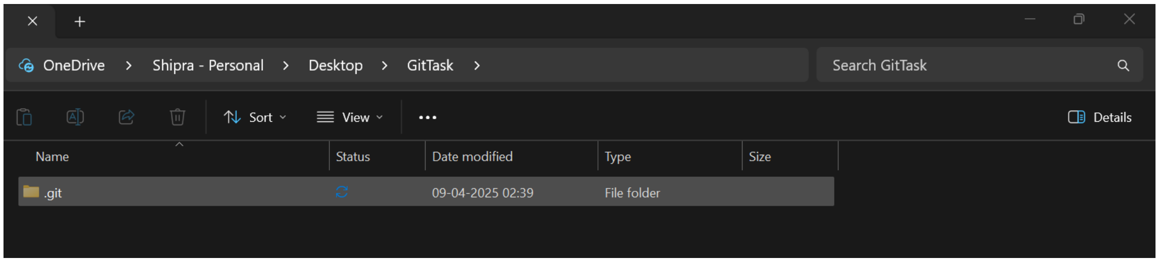
2. Initialize a Repository

- Initializes a new Git repository in the current directory, allowing version control to begin.
- Creates a **.git** folder to track changes:

git init



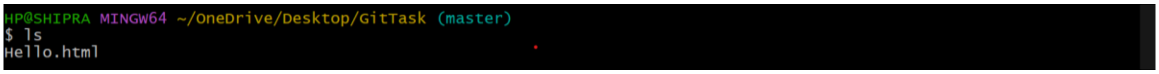
```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask
$ git init
Initialized empty Git repository in C:/Users/HP/OneDrive/Desktop/GitTask/.git/
```



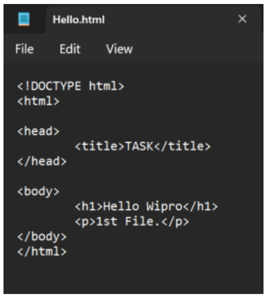
Name	Status	Date modified	Type	Size
.git		09-04-2025 02:39	File folder	

3. ls

- Check the files present in the directory.



```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ ls
Hello.html
```

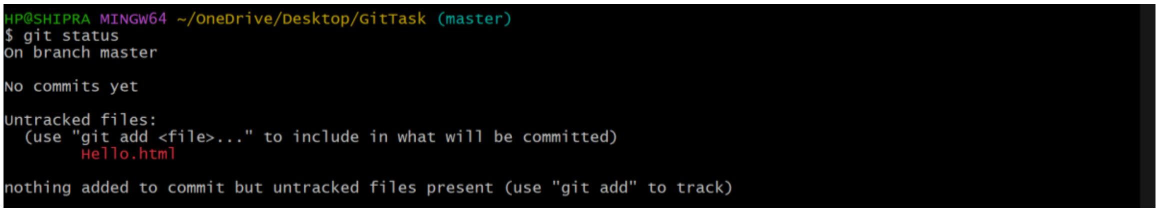


```
<!DOCTYPE html>
<html>

<head>
        <title>TASK</title>
</head>

<body>
        <h1>Hello Wipro</h1>
        <p>1st File.</p>
</body>
</html>
```

4. Check Repository Status

- Displays the current status of the repository, showing which files are staged, modified, or untracked.

git status

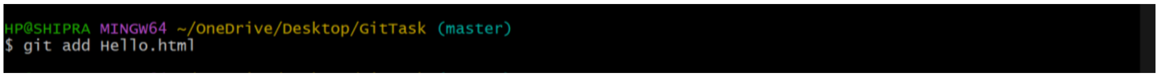


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git status
On branch master

No commits yet

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

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

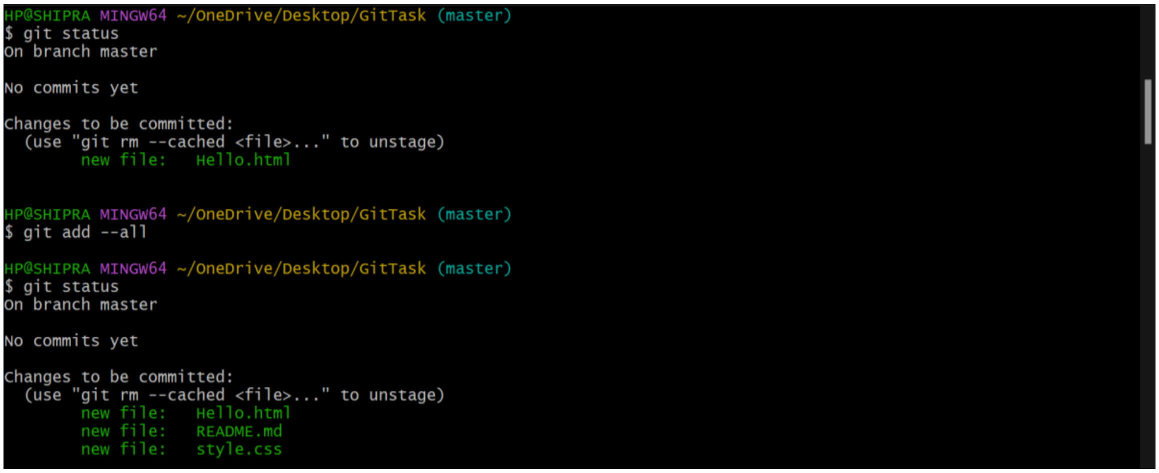
5. Add Files to Staging

- Stages a specific file to be committed. Only staged files are included in the next commit.

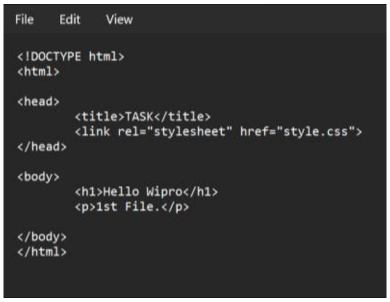
git add filename



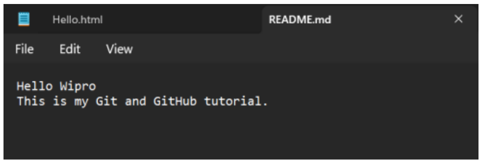
```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git add Hello.html
```



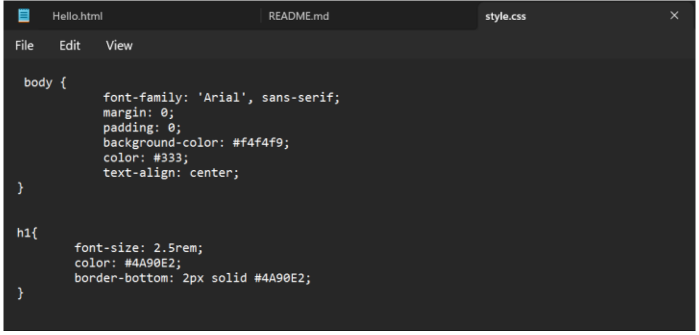
```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Hello.html

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git add --all

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Hello.html
        new file:   README.md
        new file:   style.css
```



```
<!DOCTYPE html>
<html>

<head>
        <title>TASK</title>
        <link rel="stylesheet" href="style.css">
</head>

<body>
        <h1>Hello Wipro</h1>
        <p>1st File.</p>

</body>
</html>
```



```
Hello Wipro
This is my Git and GitHub tutorial.
```

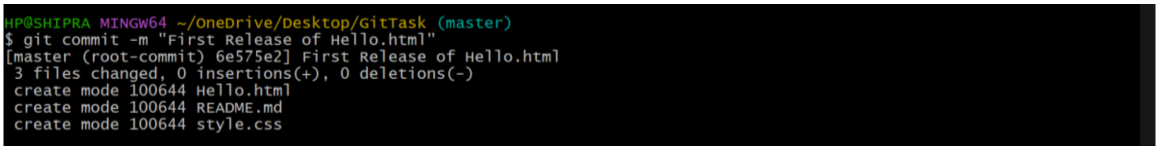


```
 body {
           font-family: 'Arial', sans-serif;
           margin: 0;
           padding: 0;
           background-color: #f4f4f9;
           color: #333;
           text-align: center;
}


h1{
        font-size: 2.5rem;
        color: #4A90E2;
        border-bottom: 2px solid #4A90E2;
}
```

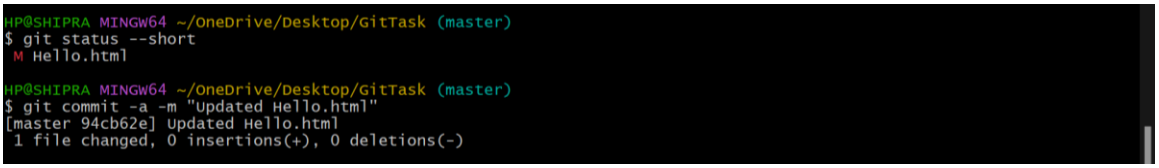
6. Commit Changes

- Creates a snapshot of the staged changes with a descriptive message.
- Save the staged changes to the repository:

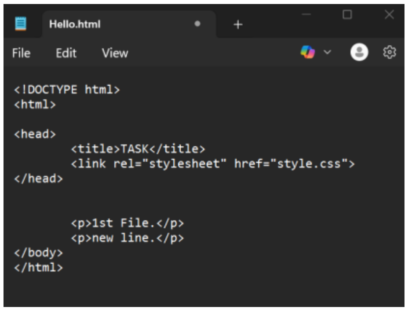
git commit -m "Your commit message"



```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git commit -m "First Release of Hello.html"
[master (root-commit) 6e575e2] First Release of Hello.html
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Hello.html
 create mode 100644 README.md
 create mode 100644 style.css
```



```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git status --short
 M Hello.html

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git commit -a -m "Updated Hello.html"
[master 94cb62e] Updated Hello.html
 1 file changed, 0 insertions(+), 0 deletions(-)
```

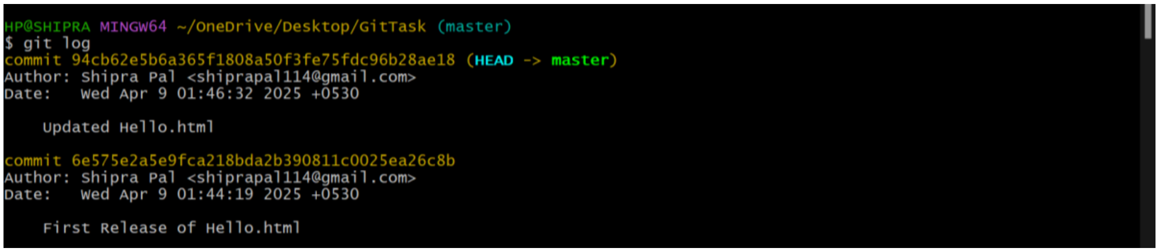


```
<!DOCTYPE html>
<html>

<head>
        <title>TASK</title>
        <link rel="stylesheet" href="style.css">
</head>


        <p>1st File.</p>
        <p>new line.</p>
</body>
</html>
```

7. View Commit History

- Displays a history of commits, including commit IDs, authors, dates, and messages.

git log

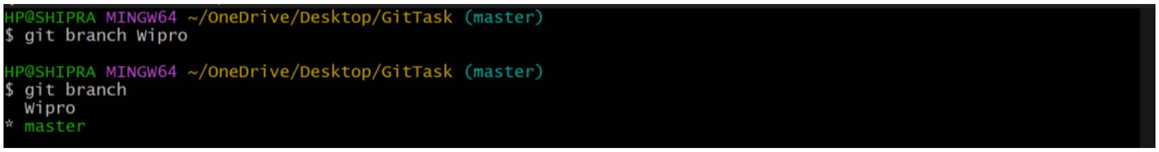


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git log
commit 94cb62e5b6a365f1808a50f3fe75fdc96b28ae18 (HEAD -> master)
Author: Shipra Pal <shiprapal114@gmail.com>
Date:   Wed Apr 9 01:46:32 2025 +0530

    Updated Hello.html

commit 6e575e2a5e9fca218bda2b390811c0025ea26c8b
Author: Shipra Pal <shiprapal114@gmail.com>
Date:   Wed Apr 9 01:44:19 2025 +0530

    First Release of Hello.html
```

8. Create a New Branch

- Creates a new branch from the current HEAD. Useful for feature development without affecting the main code.

git branch <new-branch>

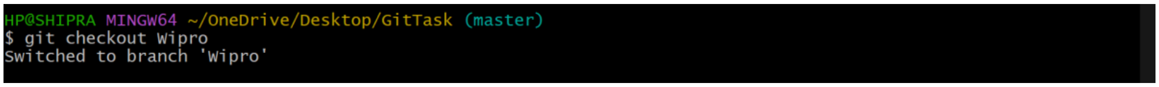


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git branch Wipro

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git branch
  Wipro
* master
```

9.Checkout

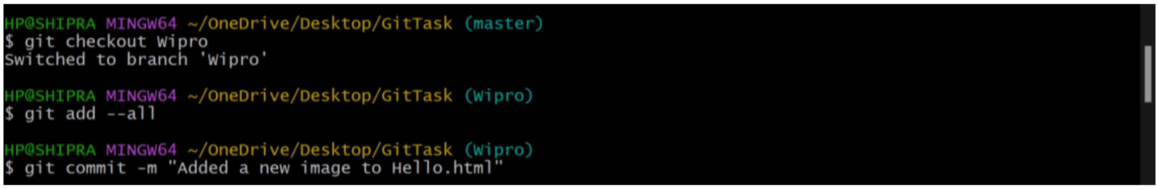
- Switches the working directory to the specified branch.

git checkout <new-branch>

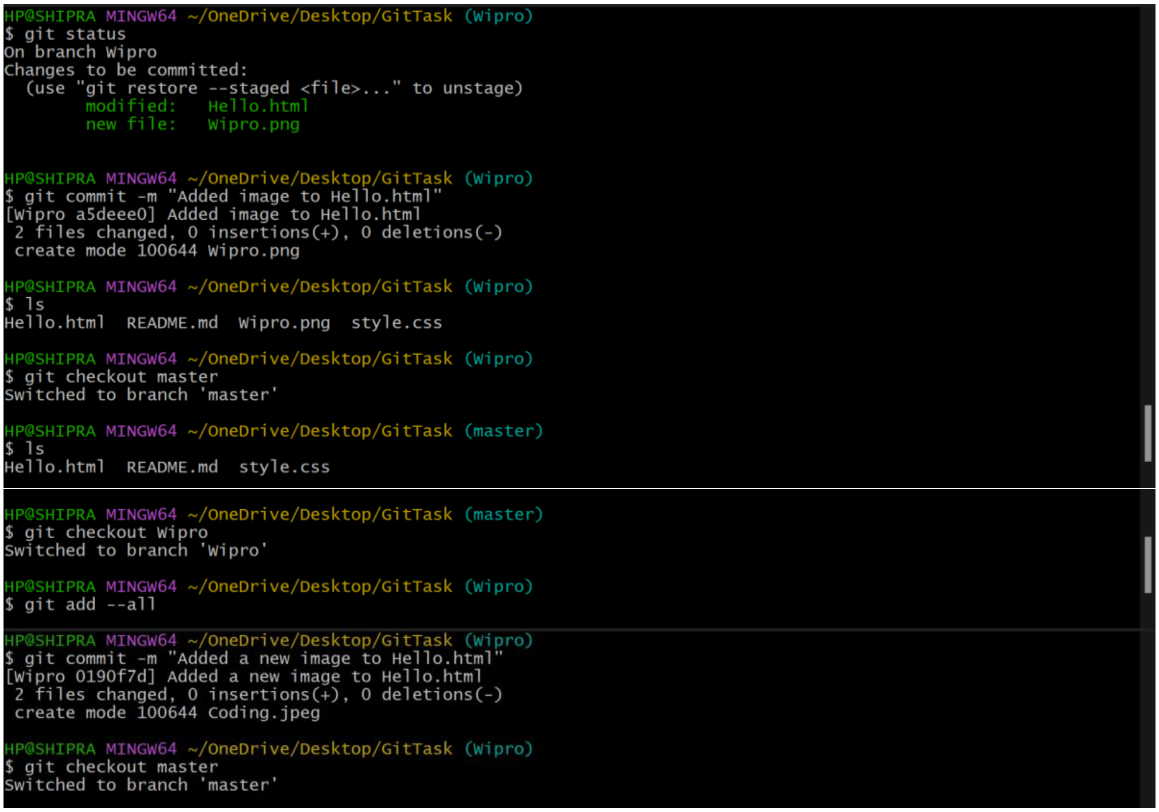


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git checkout Wipro
Switched to branch 'Wipro'
```

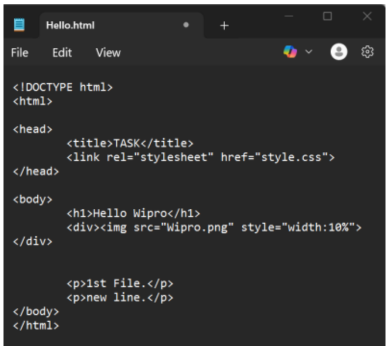
10.Commiting Changes



```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git checkout Wipro
Switched to branch 'Wipro'

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ git add --all

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ git commit -m "Added a new image to Hello.html"
```



```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ git status
On branch Wipro
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Hello.html
        new file:   Wipro.png


HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ git commit -m "Added image to Hello.html"
[Wipro a5deee0] Added image to Hello.html
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Wipro.png

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ ls
Hello.html  README.md  Wipro.png  style.css

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ git checkout master
Switched to branch 'master'

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ ls
Hello.html  README.md  style.css

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git checkout Wipro
Switched to branch 'Wipro'

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ git add --all

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ git commit -m "Added a new image to Hello.html"
[Wipro 0190f7d] Added a new image to Hello.html
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Coding.jpeg

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (Wipro)
$ git checkout master
Switched to branch 'master'
```

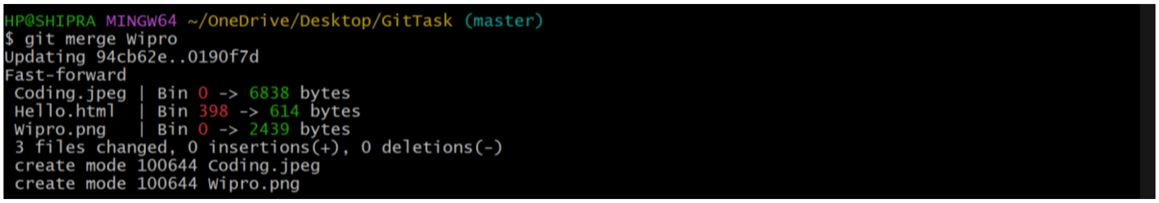


```
<!DOCTYPE html>
<html>

<head>
        <title>TASK</title>
        <link rel="stylesheet" href="style.css">
</head>

<body>
        <h1>Hello Wipro</h1>
        <div><img src="Wipro.png" style="width:10%">
</div>


        <p>1st File.</p>
        <p>new line.</p>
</body>
</html>
```

11. Merge Branches

- Combines the specified branch into the current branch, integrating changes made in the other branch.

git merge new-branch

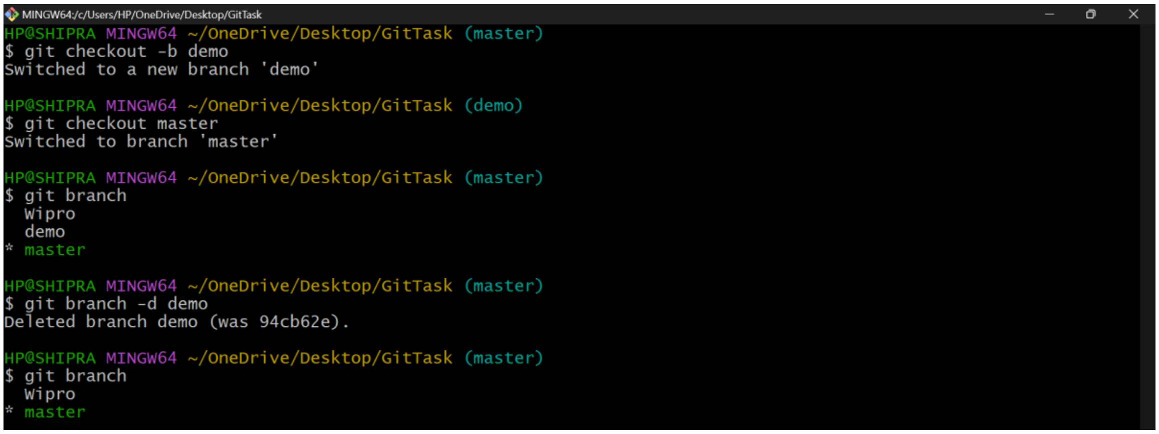


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git merge Wipro
Updating 94cb62e..0190f7d
Fast-forward
 Coding.jpeg | Bin 0 -> 6838 bytes
 Hello.html  | Bin 398 -> 614 bytes
 Wipro.png   | Bin 0 -> 2439 bytes
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Coding.jpeg
 create mode 100644 Wipro.png
```

12. Delete branches

- To delete a branch we use a command :

git branch -d <branch name>

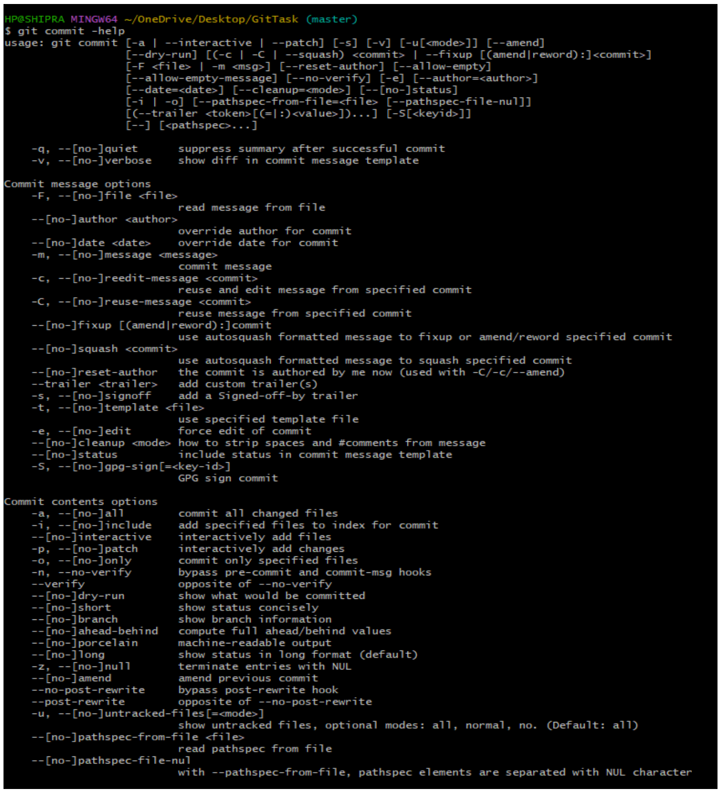


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git checkout -b demo
Switched to a new branch 'demo'

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (demo)
$ git checkout master
Switched to branch 'master'

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git branch
  Wipro
  demo
* master

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git branch -d demo
Deleted branch demo (was 94cb62e).

HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git branch
  Wipro
* master
```

13.Help

- **git commit -help** See all the available options for the specific command

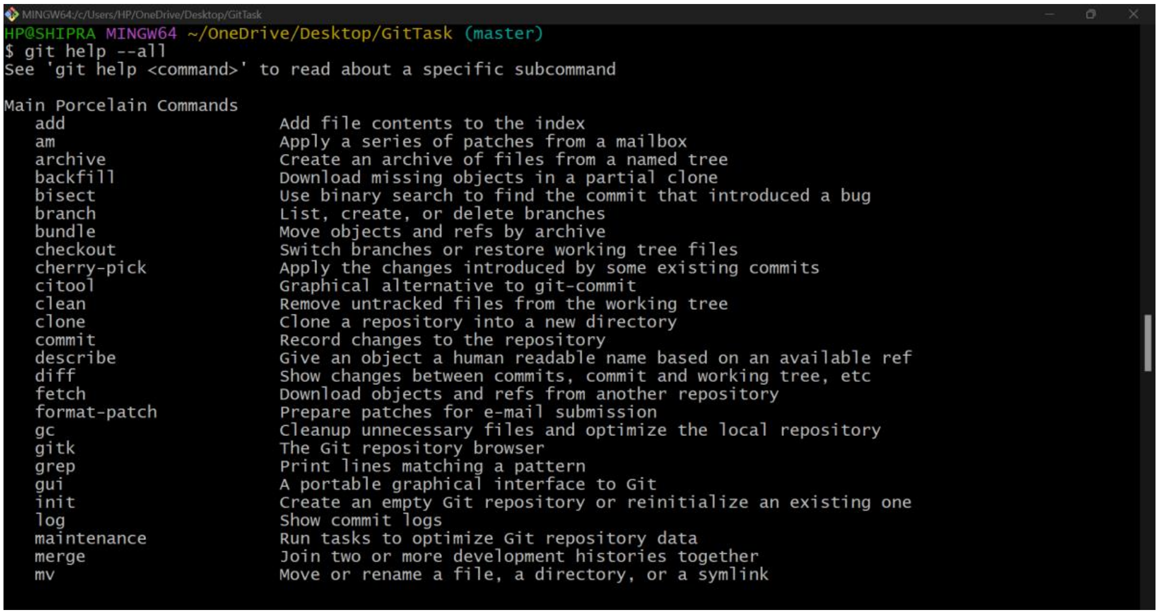
git commit -help



```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git commit -help
usage: git commit [-a | --interactive | --patch] [-s] [-v] [-u[<mode>]] [--amend]
                  [--dry-run] [(-c | -C | --squash) <commit> | --fixup [(amend|reword):]<commit>]
                  [-F <file> | -m <msg>] [--reset-author] [--allow-empty]
                  [--allow-empty-message] [--no-verify] [-e] [--author=<author>]
                  [--date=<date>] [--cleanup=<mode>] [--[no-]status]
                  [-i | -o] [--pathspec-from-file=<file> [--pathspec-file-nul]]
                  [(--trailer <token>[(=|:)<value>])...] [-S[<keyid>]]
                  [--] [<pathspec>...]

    -q, --[no-]quiet      suppress summary after successful commit
    -v, --[no-]verbose    show diff in commit message template

Commit message options
    -F, --[no-]file <file>
                          read message from file
    --[no-]author <author>
                          override author for commit
    --[no-]date <date>    override date for commit
    -m, --[no-]message <message>
                          commit message
    -c, --[no-]reedit-message <commit>
                          reuse and edit message from specified commit
    -C, --[no-]reuse-message <commit>
                          reuse message from specified commit
    --[no-]fixup [(amend|reword):]commit
                          use autosquash formatted message to fixup or amend/reword specified commit
    --[no-]squash <commit>
                          use autosquash formatted message to squash specified commit
    --[no-]reset-author   the commit is authored by me now (used with -C/-c/--amend)
    --trailer <trailer>   add custom trailer(s)
    -s, --[no-]signoff    add a Signed-off-by trailer
    -t, --[no-]template <file>
                          use specified template file
    -e, --[no-]edit       force edit of commit
    --[no-]cleanup <mode> how to strip spaces and #comments from message
    --[no-]status         include status in commit message template
    -S, --[no-]gpg-sign[=<key-id>]
                          GPG sign commit

Commit contents options
    -a, --[no-]all        commit all changed files
    -i, --[no-]include    add specified files to index for commit
    --[no-]interactive    interactively add files
    -p, --[no-]patch      interactively add changes
    -o, --[no-]only       commit only specified files
    -n, --no-verify       bypass pre-commit and commit-msg hooks
    --verify              opposite of --no-verify
    --[no-]dry-run        show what would be committed
    --[no-]short          show status concisely
    --[no-]branch         show branch information
    --[no-]ahead-behind   compute full ahead/behind values
    --[no-]porcelain      machine-readable output
    --[no-]long           show status in long format (default)
    -z, --[no-]null       terminate entries with NUL
    --[no-]amend          amend previous commit
    --no-post-rewrite     bypass post-rewrite hook
    --post-rewrite        opposite of --no-post-rewrite
    -u, --[no-]untracked-files[=<mode>]
                          show untracked files, optional modes: all, normal, no. (Default: all)
    --[no-]pathspec-from-file <file>
                          read pathspec from file
    --[no-]pathspec-file-nul
                          with --pathspec-from-file, pathspec elements are separated with NUL character
```

- **git help --all** See all possible command

git help --all

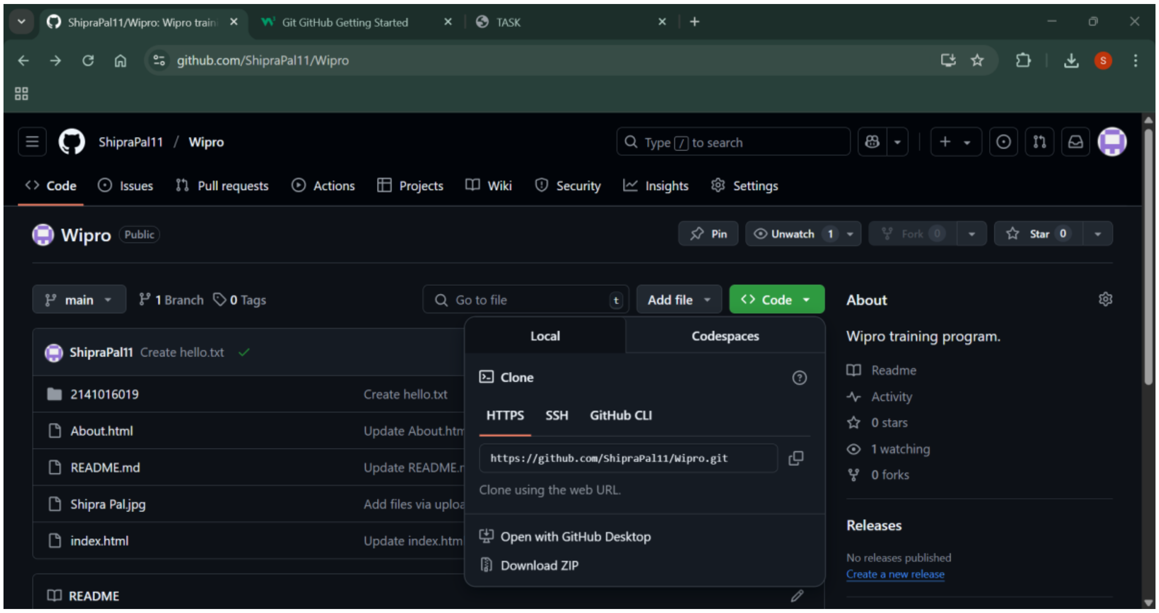


```
MINGW64:/c/Users/HP/OneDrive/Desktop/GitTask
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git help --all
See 'git help <command>' to read about a specific subcommand

Main Porcelain Commands
   add                  Add file contents to the index
   am                   Apply a series of patches from a mailbox
   archive              Create an archive of files from a named tree
   backfill             Download missing objects in a partial clone
   bisect               Use binary search to find the commit that introduced a bug
   branch               List, create, or delete branches
   bundle               Move objects and refs by archive
   checkout             Switch branches or restore working tree files
   cherry-pick          Apply the changes introduced by some existing commits
   citool               Graphical alternative to git-commit
   clean                Remove untracked files from the working tree
   clone                Clone a repository into a new directory
   commit               Record changes to the repository
   describe             Give an object a human readable name based on an available ref
   diff                 Show changes between commits, commit and working tree, etc
   fetch                Download objects and refs from another repository
   format-patch         Prepare patches for e-mail submission
   gc                   Cleanup unnecessary files and optimize the local repository
   gitk                 The Git repository browser
   grep                 Print lines matching a pattern
   gui                  A portable graphical interface to Git
   init                 Create an empty Git repository or reinitialize an existing one
   log                  Show commit logs
   maintenance          Run tasks to optimize Git repository data
   merge                Join two or more development histories together
   mv                   Move or rename a file, a directory, or a symlink
```

14. Clone a Repository

- Creates a local copy of a remote repository by downloading its contents and history

git clone https://github.com/username/repo-name.git



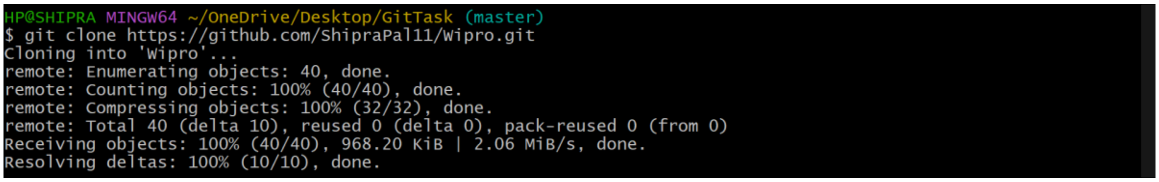
ShipraPal11 Create hello.txt

2141016019	Create hello.txt
About.html	Update About.htm
README.md	Update README.r
Shipra Pal.jpg	Add files via uploa
index.html	Update index.htm

https://github.com/ShipraPal11/Wipro.git

Clone using the web URL.

Wipro training program.

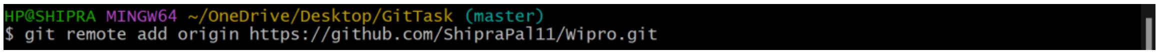


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git clone https://github.com/ShipraPal11/Wipro.git
Cloning into 'Wipro'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 40 (delta 10), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (40/40), 968.20 KiB | 2.06 MiB/s, done.
Resolving deltas: 100% (10/10), done.
```

15. Connect to GitHub

- Links the local repository to a remote one, usually on GitHub, for pushing and pulling code.

git remote add origin ***https://github.com/username/repo-name.git***

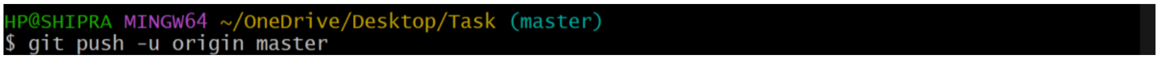


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/GitTask (master)
$ git remote add origin https://github.com/ShipraPal11/Wipro.git
```

16. Push Code to GitHub

- Pushes local commits to the remote master branch and sets the upstream tracking reference
- **-u** sets upstream tracking so future **git push** commands can be used without specifying the branch.

git push -u origin master

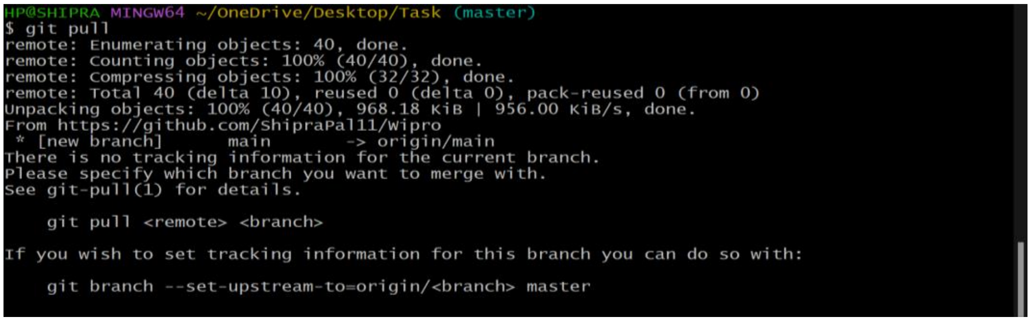


```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/Task (master)
$ git push -u origin master
```

17. Pull Changes

- Fetches and integrates changes from the remote master branch into the local branch.

git pull origin master



```
HP@SHIPRA MINGW64 ~/OneDrive/Desktop/Task (master)
$ git pull
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 40 (delta 10), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (40/40), 968.18 KiB | 956.00 KiB/s, done.
From https://github.com/ShipraPal11/Wipro
 * [new branch]      main       -> origin/main
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> master
```