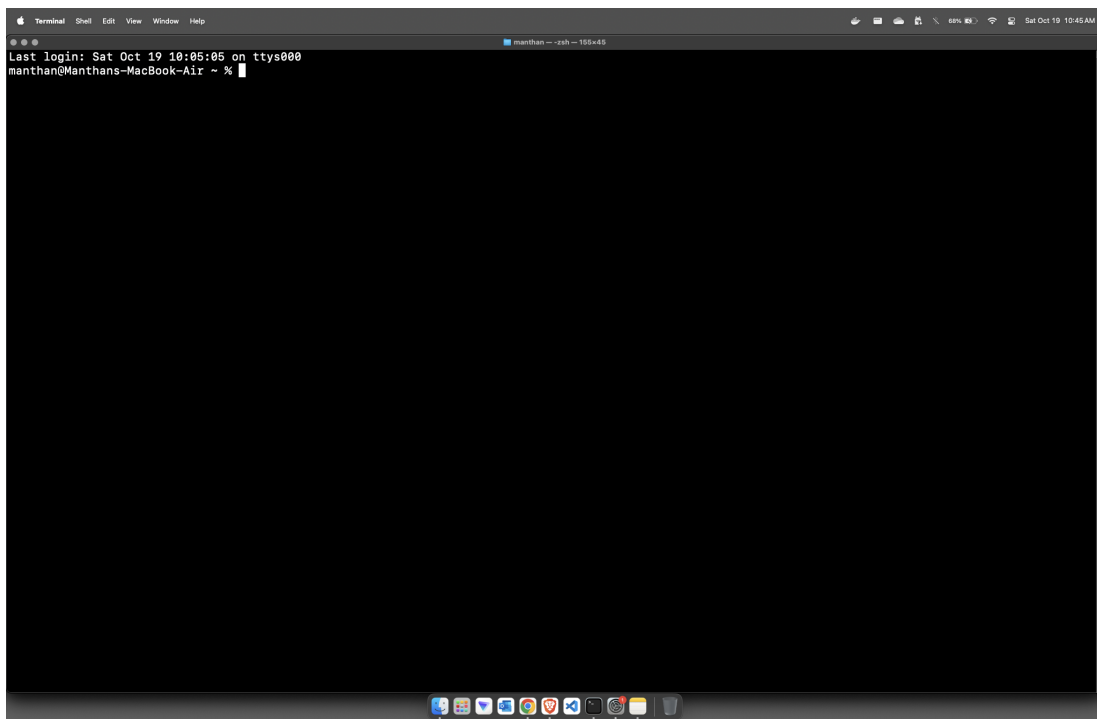


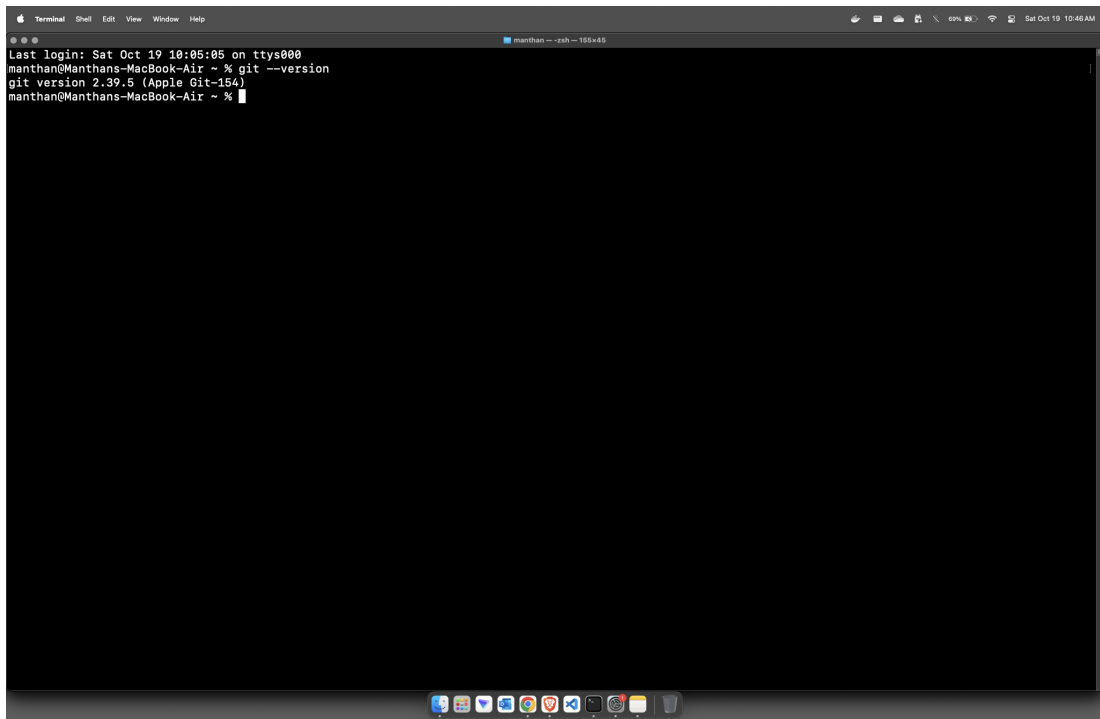
Git and Github - Lecture 3

1. Blank Terminal Screenshot

Step - Open Terminal

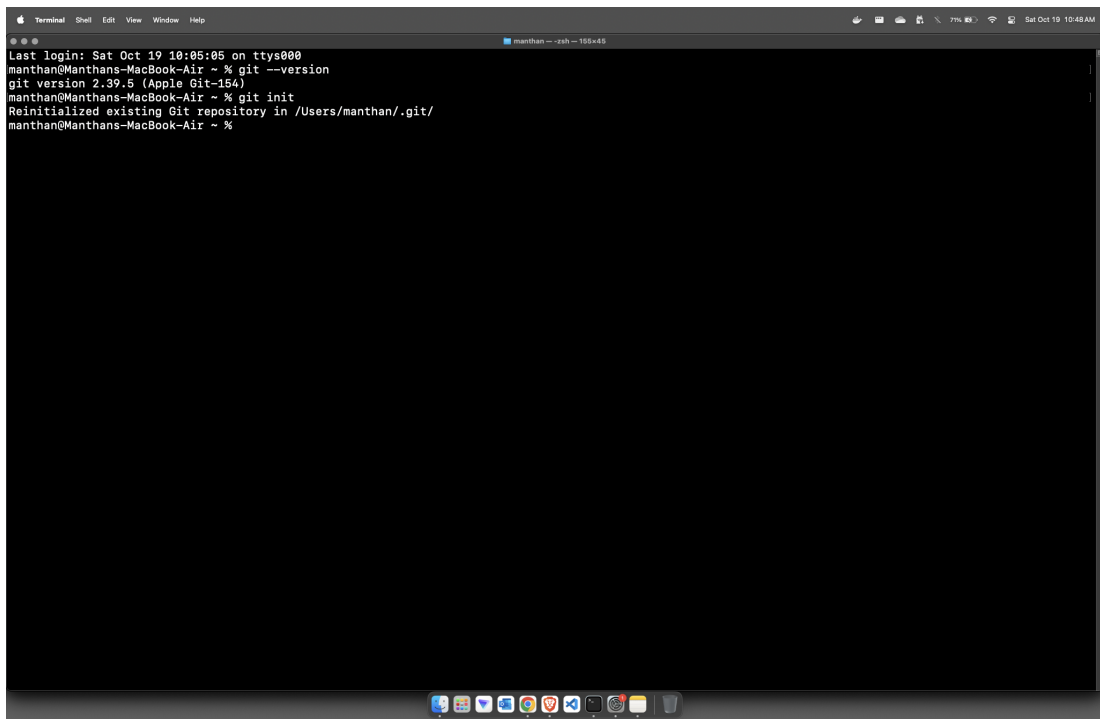


2. To check git installation, check with git —version

A screenshot of a macOS Terminal window. The window title is "Terminal" with menu options "Shell", "Edit", "View", "Window", and "Help". The status bar at the top right shows "manthan -- ssh -- 155x45", a battery icon at 69%, and the date/time "Sat Oct 19 10:46 AM". The terminal text shows a successful login on "Sat Oct 19 10:05:05 on ttys000". The user "manthan@Manthans-MacBook-Air" runs the command "git --version", which outputs "git version 2.39.5 (Apple Git-154)". The prompt "manthan@Manthans-MacBook-Air ~ %" is followed by a cursor. The dock at the bottom contains icons for Finder, Launchpad, Safari, Mail, Messages, Photos, App Store, and Trash.

```
Terminal Shell Edit View Window Help
manthan -- ssh -- 155x45
Last login: Sat Oct 19 10:05:05 on ttys000
manthan@Manthans-MacBook-Air ~ % git --version
git version 2.39.5 (Apple Git-154)
manthan@Manthans-MacBook-Air ~ %
```

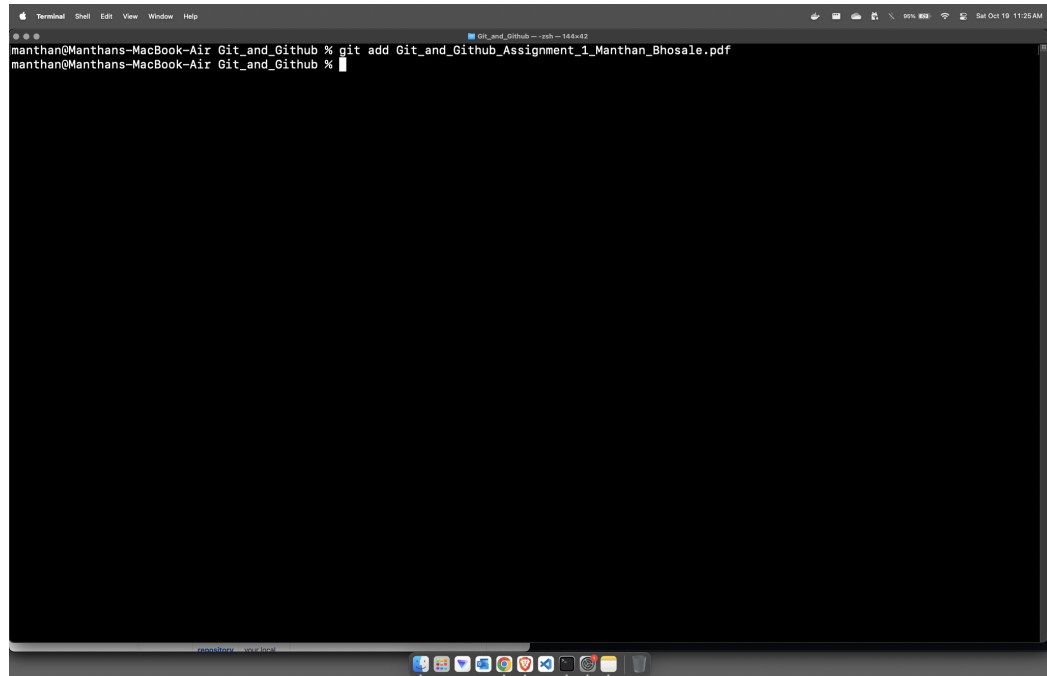
3. Initialization

A screenshot of a macOS Terminal window, similar to the one above. The window title is "Terminal" with menu options "Shell", "Edit", "View", "Window", and "Help". The status bar at the top right shows "manthan -- ssh -- 155x45", a battery icon at 71%, and the date/time "Sat Oct 19 10:48 AM". The terminal text shows the same login as the previous window. The user "manthan@Manthans-MacBook-Air" runs the command "git init", which outputs "Reinitialized existing Git repository in /Users/manthan/.git/". The prompt "manthan@Manthans-MacBook-Air ~ %" is followed by a cursor. The dock at the bottom contains icons for Finder, Launchpad, Safari, Mail, Messages, Photos, App Store, and Trash.

```
Terminal Shell Edit View Window Help
manthan -- ssh -- 155x45
Last login: Sat Oct 19 10:05:05 on ttys000
manthan@Manthans-MacBook-Air ~ % git --version
git version 2.39.5 (Apple Git-154)
manthan@Manthans-MacBook-Air ~ % git init
Reinitialized existing Git repository in /Users/manthan/.git/
manthan@Manthans-MacBook-Air ~ %
```

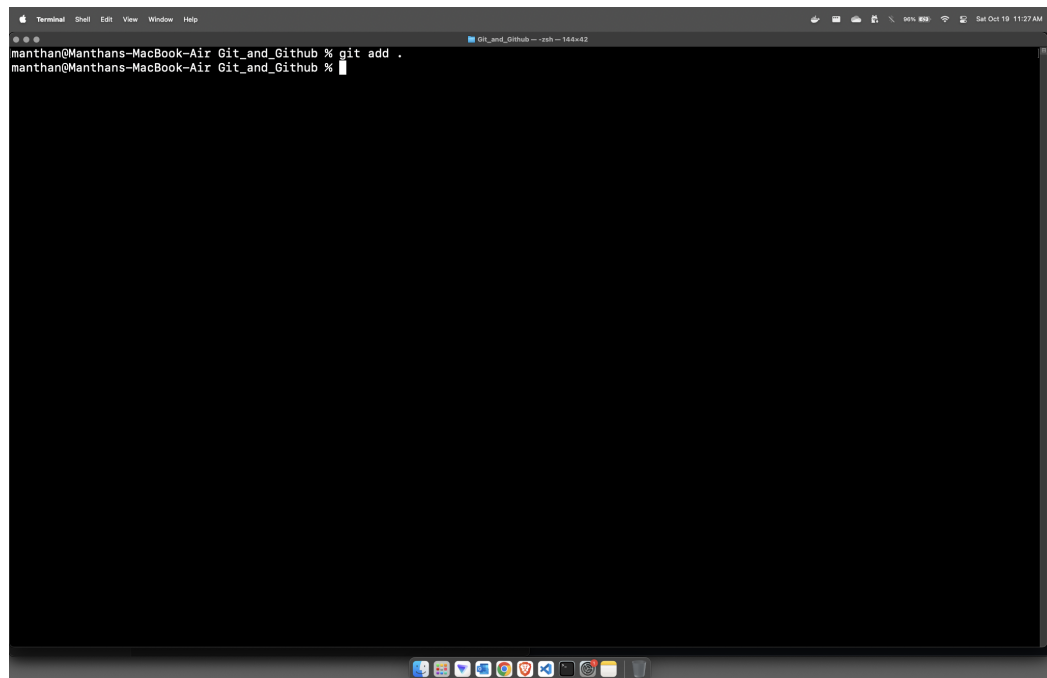
4. Basic Operations:

- `git add [file]`



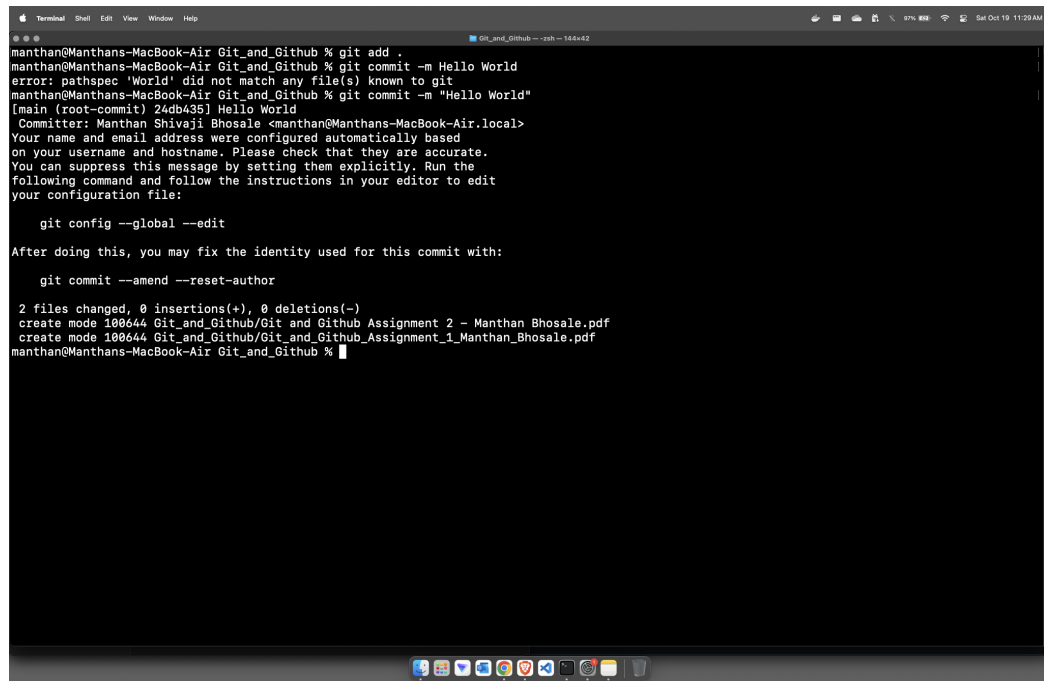
```
manthan@Manthans-MacBook-Air Git_and_Github % git add Git_and_Github_Assignment_1_Manthan_Bhosale.pdf
manthan@Manthans-MacBook-Air Git_and_Github %
```

- `git add .`



```
manthan@Manthans-MacBook-Air Git_and_Github % git add .
manthan@Manthans-MacBook-Air Git_and_Github %
```

- git commit -m [message]

A terminal window on a Mac showing the execution of git commands. The user runs 'git add .' and 'git commit -m Hello World'. An error message appears: 'error: pathspec 'World' did not match any file(s) known to git'. The user then runs 'git commit -m "Hello World"'. The terminal shows the commit hash '24db435f', the commit message 'Hello World', and the committer 'Manthan Shivaji Bhosale <manthan@Manthans-MacBook-Air.local>'. It also shows the configuration of the user's name and email address, and the creation of two files: 'Git_and_Github/Git_and_Github_Assignment_2 - Manthan Bhosale.pdf' and 'Git_and_Github/Git_and_Github_Assignment_1_Manthan_Bhosale.pdf'.

```
manthan@Manthans-MacBook-Air Git_and_Github % git add .
manthan@Manthans-MacBook-Air Git_and_Github % git commit -m Hello World
error: pathspec 'World' did not match any file(s) known to git
manthan@Manthans-MacBook-Air Git_and_Github % git commit -m "Hello World"
[main (root-commit) 24db435f] Hello World
Committer: Manthan Shivaji Bhosale <manthan@Manthans-MacBook-Air.local>
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:

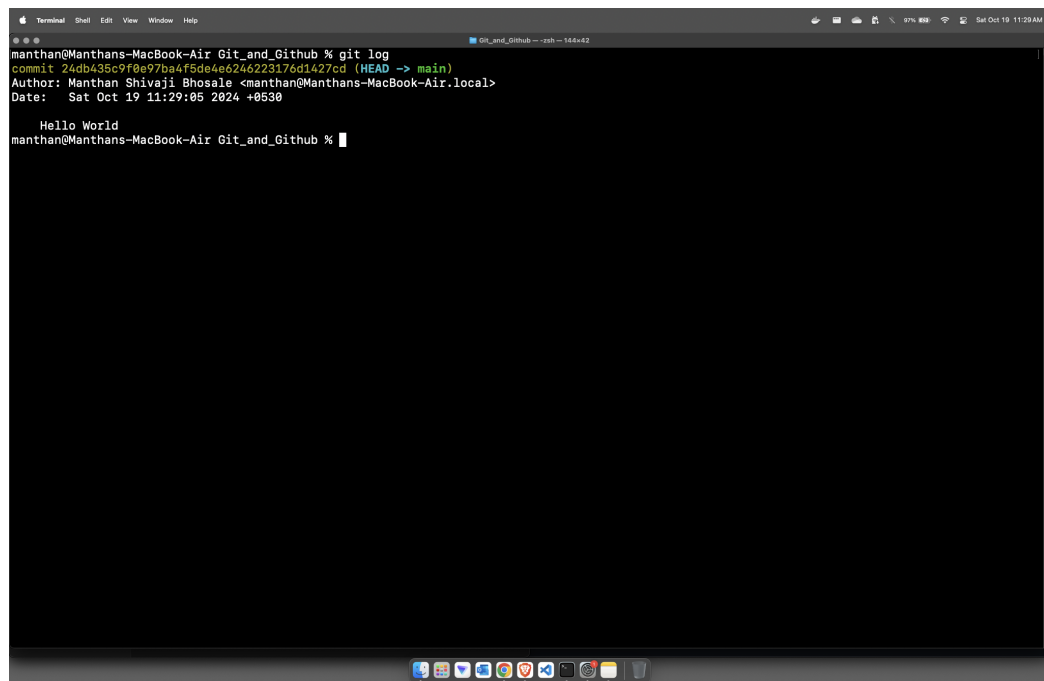
    git config --global --edit

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

    git commit --amend --reset-author

2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Git_and_Github/Git_and_Github_Assignment_2 - Manthan Bhosale.pdf
create mode 100644 Git_and_Github/Git_and_Github_Assignment_1_Manthan_Bhosale.pdf
manthan@Manthans-MacBook-Air Git_and_Github %
```

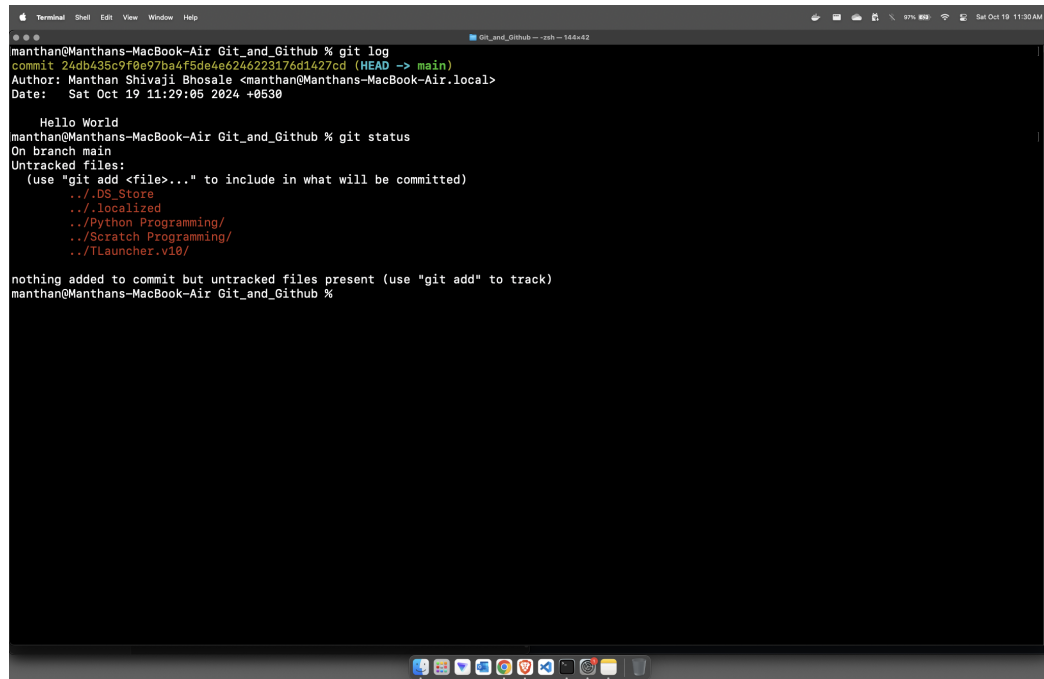
- git log

A terminal window on a Mac showing the output of the 'git log' command. The output displays the commit hash '24db435c9f9e97ba4f8de4e6246223176d1427cd', the commit message 'Hello World', the author 'Manthan Shivaji Bhosale <manthan@Manthans-MacBook-Air.local>', and the date 'Sat Oct 19 11:29:05 2024 +0530'.

```
manthan@Manthans-MacBook-Air Git_and_Github % git log
commit 24db435c9f9e97ba4f8de4e6246223176d1427cd (HEAD -> main)
Author: Manthan Shivaji Bhosale <manthan@Manthans-MacBook-Air.local>
Date: Sat Oct 19 11:29:05 2024 +0530

    Hello World
manthan@Manthans-MacBook-Air Git_and_Github %
```

- git status

A screenshot of a macOS Terminal window. The window title is "Terminal" with menu options "Shell", "Edit", "View", "Window", and "Help". The terminal shows the following commands and output:
manthan@Manthans-MacBook-Air Git_and_Github % git log
commit 24db435c9f8e97ba4f6de4e6246223176d1427cd (HEAD -> main)
Author: Manthan Shrivaji Bhosale <manthan@Manthans-MacBook-Air.local>
Date: Sat Oct 19 11:29:05 2024 +0530

Hello World
manthan@Manthans-MacBook-Air Git_and_Github % git status
On branch main
Untracked files:
(use "git add <file>..." to include in what will be committed)
../.DS_Store
../.localized
../Python Programming/
../Scratch Programming/
../TLauncher.v10/

nothing added to commit but untracked files present (use "git add" to track)
manthan@Manthans-MacBook-Air Git_and_Github %
The terminal window has a dark background with light-colored text. The macOS dock is visible at the bottom with various application icons.

Punishment:

1. Git is a free and open-source distributed version control system designed for tracking changes in source code during software development. It allows developers to work on the same project without overwriting each other's files, and provides mechanisms for collaboration, branching, merging and rolling back changes. Git differs from other version control systems such as Subversion in that it is a distributed system rather than centralized, which means changes are stored locally and can be shared with others through a network.
2. The "git add" command is used to add files or directories to the staging area for further processing by Git's commit command. This allows developers to selectively choose what files they want to include in their commits without having to make changes directly on the repository itself.
3. The "git commit" command saves all local changes to the repository, while the "git push" command sends those changes to a remote repository for other developers to see and work with. This allows developers to collaborate and

work together on projects even when they are not in the same physical location.

4. To create a new branch in Git, you can use the "git branch" command followed by the name of the new branch you want to create. For example: "git branch newbranchname". Once the branch has been created, you can switch between it and your main branch using the "git checkout" command.
5. The purpose of Git branches is to allow developers to work on different features or enhancements simultaneously without affecting each other's progress. Branches are also useful for creating feature branches that can be merged into the master branch at a later date, allowing developers to test and validate changes before they are merged into production code.
6. To merge two Git branches together, you can use the "git merge" command followed by one of the branches you want to merge into another. For example: "git merge mainbranch". This will create a new branch that combines both sets of changes from the original branches.
7. A remote repository is an external location where your Git repository is stored and can be accessed by other developers. It allows multiple people to work on a project simultaneously without overwriting each other's files, since they are all working on different branches. Remote repositories are also useful for collaboration with other teams or organizations who may not have access to your local environment.
8. To revert a change in Git, you can use the "git reset" command followed by a commit hash. This will undo the changes made in that specific commit and leave your files untouched.
9. To collaborate with others using Git, you can share the codebase via a remote repository, allowing other developers to work on the same project without being in the same physical location. You can also use version control systems like Git to create pull requests, which allow other developers to review and comment on changes before they are merged into the main branch.
10. The difference between "git fetch" and "git pull" is that "git fetch" only downloads changes from a remote repository without merging them into your local codebase, while "git pull" also incorporates those changes into your local environment.

