Today Topics :

1.Version Controls
2.Git
3.Githud

Visual code editor:
download link :
https://code.visualstudio.com/

Useful extensions for VS Code:

1.Bracket pair colorizer
2.Cobalt2 theme official
3. Git lens
4.Live Server
Today Topics :

1.Version Controls
2.Git
3.GitHub

Visual code editor:
download link :
https://code.visualstudio.com/

Useful extensions for VS Code:
1.Bracket pair colorizer
2.Cobalt2 theme official
3. Git lens
4.Live Server
5. Vs-code-icons

Preference for VS Code:
- Go to file > preferences
- change stuff as per requirements, like font size, tab space (2 or 4)
- use as per your needs, this is entirely based on your personal preference

1) Version Controls:

What is Version Control?

Version control systems are a category of software tools that helps record changes to files by keeping a track of modifications done to the code.

Analogy: This is like playing a video game and saving at checkpoints, these checkpoints are equivalent to save states in Git version control.
Software version management that is, SVM (before 2005),
Resource for more information: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

2)Git:

Git is an Open Source Distributed Version Control System. Now that's a lot of words to define Git.
Let me break it down and explain the wording:
- Control System: This means that Git is a content tracker. So Git can be used to store content — it is mostly used to store code due to the other features it provides.
- Version Control System: The code which is stored in Git keeps changing as more code is added. Also, many developers can add code in parallel. So Version Control System helps in handling this by maintaining a history of what changes have happened. Also, Git provides features like branches and merges, which I will be covering later.
- Distributed Version Control System: Git has a remote repository that is stored in a server and a local repository that 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.

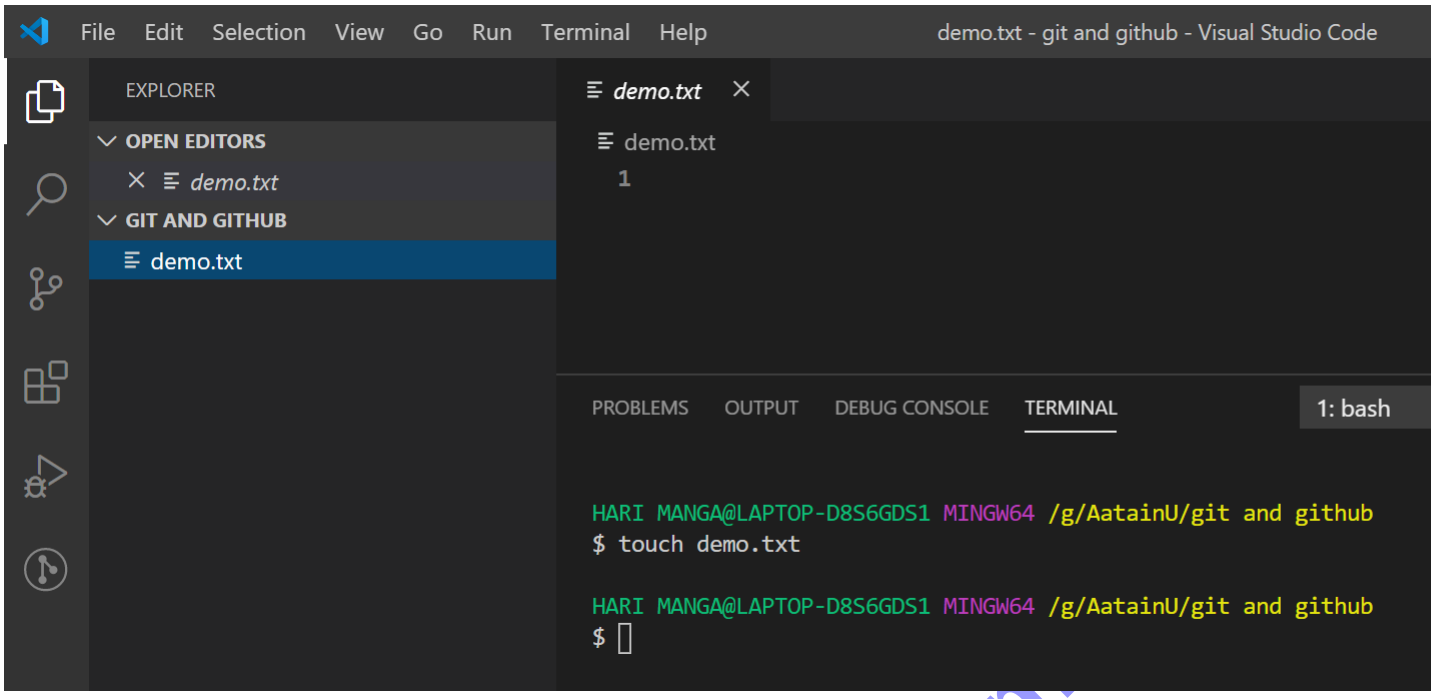
Why a Version Control System like Git is needed:

Real-life projects generally have multiple developers working in parallel. So a version control system like Git is needed to ensure there are no code conflicts between the developers.
Additionally, the requirements in such projects change often. So a version control system allows developers to revert and go back to an older version of the code.
Finally, sometimes several projects which are being run in parallel involve the same codebase. In such a case, the concept of branching in Git is very important.
Verify if Git is installed by using the following command in the command prompt:
git -- version

step 1: open VS code --> file --> open folder .
step 2: open terminal -- > view --> select terminal option --> select default shell to bash (located at right side of vs)

then create a file using with touch command

Command : touch <space> demo.txt

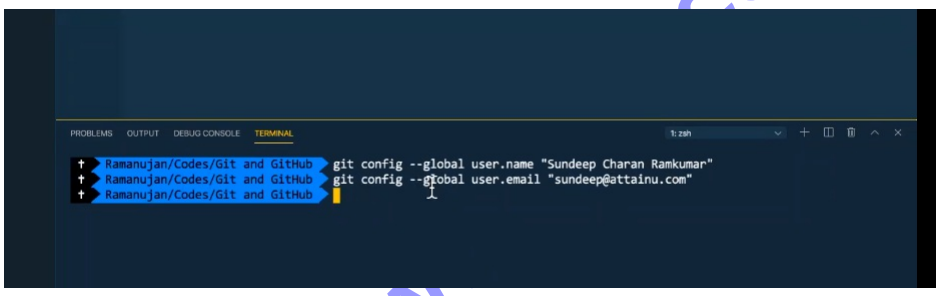


after this, we need to follow 3 steps

1. initialize the repository
2. stage
3. commit
4. push (optional)

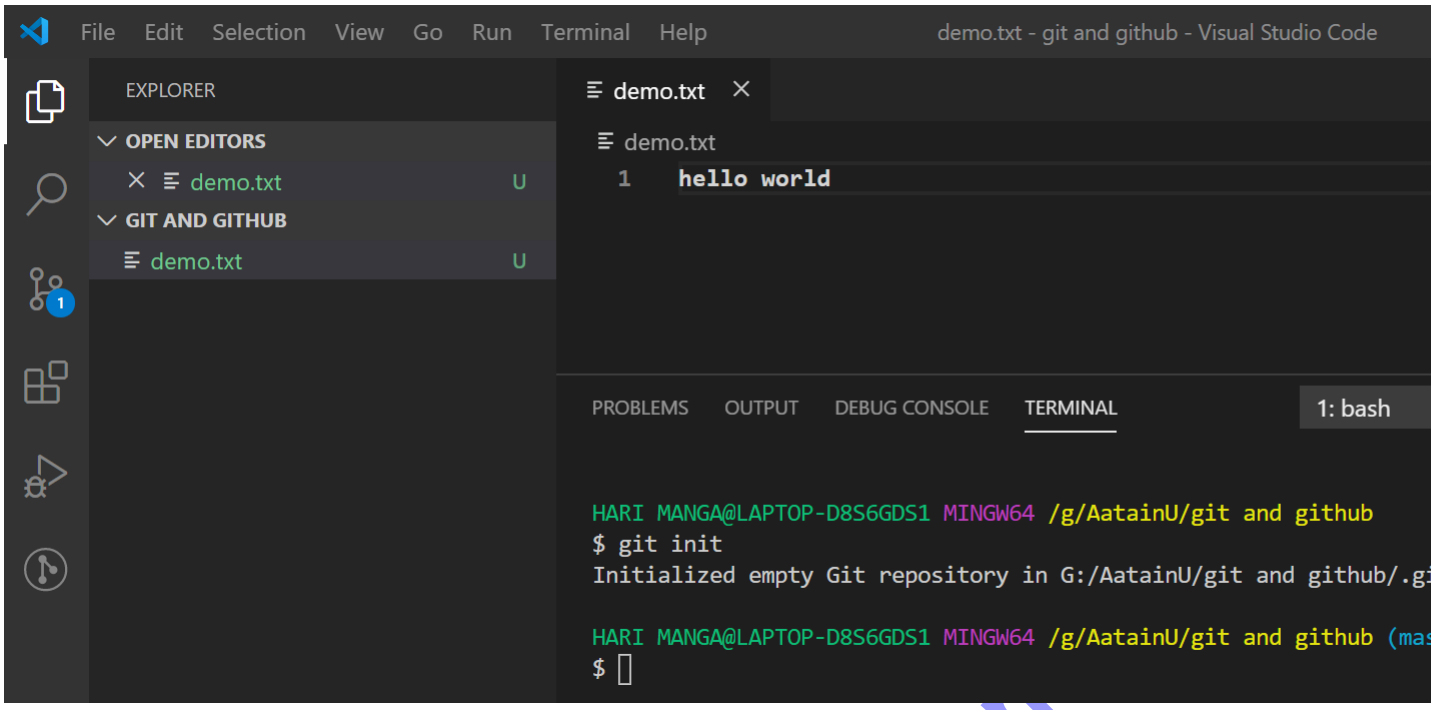
- **initialization is just one-time operation)**

for new users, you need to do git config for user name and email
git config -- global user. name " your name"
git config --global user.email " email id "



1)Creating A new repository initialize

Command: git init



we found a message
initialized empty git repository in ----------

The git init command adds a local Git repository to the project.

when the moment we have done the initialize we can found a color change in a file name

there are some colors vs code automatically vs gives for us

Green - new file not staged yet

Yellow - it means code has staged not yet committed

Red: that particular file is not committed and skipped

2)Stage: there are two types of stages

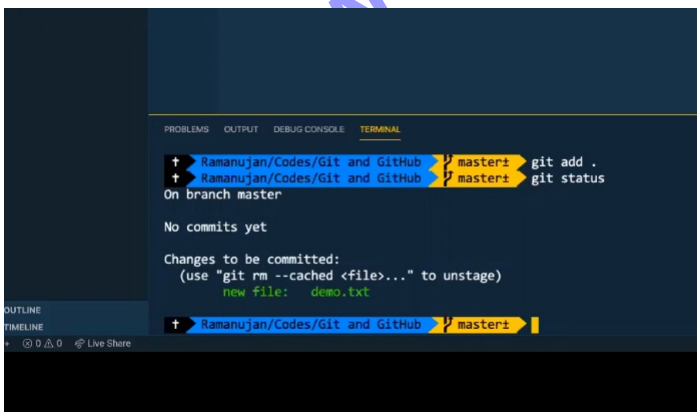
1. specific file, (Command : add <space> file name (demo.txt))
2. all the files. (Command : add <space> .)

is up to u

*when we have done this file color will change color and the letter will change U to A

then check the Git Status

command : git <space> status



3)commit :

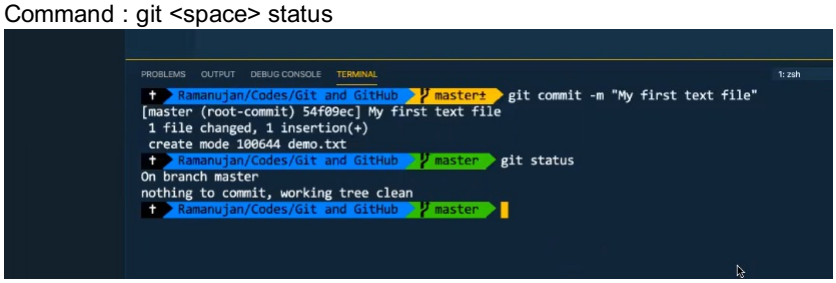
Command: git commit -m " your message"

“Initial Commit” is the commit message here. Enter a relevant commit message to indicate what code changes were done in that particular commit.

to remove the file's contents from the index
git rm --cached -r or you can add the file name to .git ignore file to do the same

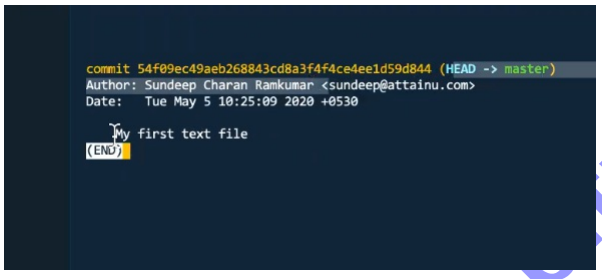
for Git status

Command : git <space> status



*Git Log: **for history**

Command : Git <space> log

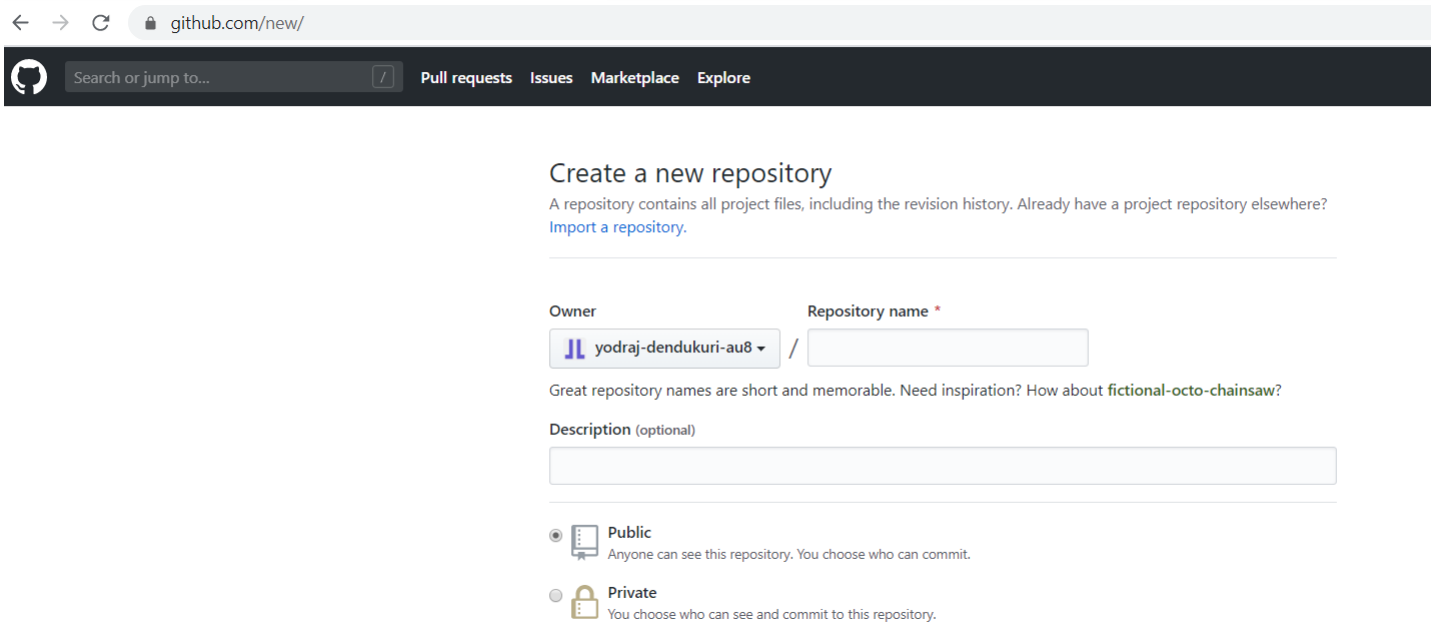


if you press Q it will end it and return to where your working

4)push:

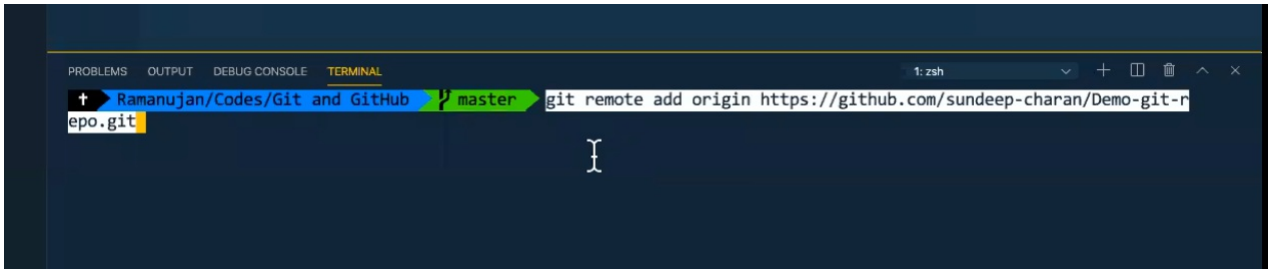
before pushing your code to GitHub

we have to log in to GitHub

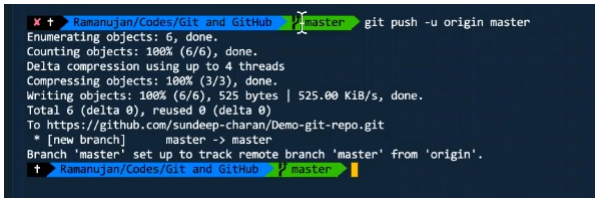


we did add and commit now we are using

with the help of above commands, we are adding cloud available path to the local machine



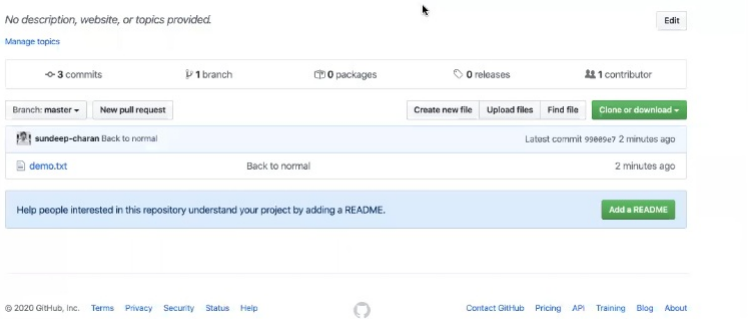
adding your local code to cloud master branch using with git push -u origin master command



make sure that your branch should be the same branch what you're about uploaded

then come back to GitHub and refresh the page

you can see all the required data that has been stored



Session - End

Thank You