Name :- Vijay Gurung

ID Number : MST03 – 0070

Topic : Git & GitHub

Suggestion URL : [https://docs.chaicode.com/git-and-github]

From "Git" : Push & Pull_To_"GitHub"_Final Outcome_ URL :

[https://github.com/Vijaygurung1/Lets-Try-Git]

Git: Git is a version control software. **(Software)**

GitHub: GitHub is an online platform that hosts Git repositories, providing a service for collaborative development. It is **(Service)** basically.

Github was not only in the market earlier before that there was others

For Example Git :-

Think of playing a game like "**Call of Duty"** game. At certain points in the game, we can save our progress. For example, after completing the first stage, we save our game, then move on to the second stage and save again and so on. If we have completed 10 stages lets assume, but want to replay stages 4 and 5, we can go back to those saved points and start from there.

Similarly, in software, our software may not always work perfectly. Sometimes it works well and other times it doesn't. When the software is functioning correctly, we create checkpoints. If an issue arises later, we can return to a previous checkpoint where everything was working fine. These checkpoints are what we call it as **Git.**

Git allows us to navigate through our code at any point in time. We can go back to a previous state, remove changes or create new checkpoints. This makes **Git** extremely useful for managing and tracking changes in our software.

With **Git**, we often refer to it as a **Version Control System (VCS).** We create checkpoints which help to track or changes in files, making it easier to manage and collaborate on code.

Difference between Git and GitHub:-

Git is a version control system that is used to track changes to your files. It is a free and open-source software that is available for Windows, macOS, and Linux. Remember, GIT is a software and can be installed on your computer.

GitHub is a web-based hosting service for Git repositories. Github is an online platform that allows us to store and share our code with others. It is a popular platform for developers to collaborate on projects and to share code. It is not that GitHub is the only provider of Git repositories, but it is one of the most popular ones.

A little on version control systems :

Version control systems are used to manage the history of our code. They allow us to track changes to our files and to collaborate with others. Version control systems (VCS) are essential for software development. Consider version control as a checkpoint in game. We can move to any time in the game and we can always go back to the previous checkpoint. This is the same concept in software development.

Before Git became mainstream, version control systems (VCS) were used by developers to manage their code. They were called SCCS (Source Code Control System). SCCS was a proprietary software that was used to manage the history of code. It was expensive and not very user-friendly. Git was created to replace SCCS and to make version control more accessible and user-friendly. Some common version control systems are Subversion (SVN), CVS, and Perforce.

How to Install Git :

To install Git, We can use command line or you can visit official website and download the installer for your operating system. Git is available for Windows, macOS, and Linux and is available at
[https://git-scm.com/downloads]

Git Documentation :-

[https://git-scm.com/doc]

Conclusion :

Git is a version control software. Before Git, there were other version control systems like SVN and CVS. However, Git revolutionized the market. It is open-source and free and it is widely used everywhere. Git is used to create checkpoints in our code.

Git supports collaboration, allowing multiple people to work on the same project simultaneously without interfering with each other's work. Our files won't disrupt others and their files won't affect ours, even if we are working in the same directory. Git enables efficient collaboration within a team, helping to manage and merge code seamlessly.

Terminology - GitBash

In the software industry, we use different terms to refer to similar concepts. For Example, some people call it a **"Folder"** some call it a **"directory"** and others refer to it as a **"repository".** While this terms are essentially mean the same thing, just expressed in different ways.

We will going to learn the all the commands and try to understand step by steps.

To check your git version, you can run the following command:

# Step : 1

[git --version] to check the version

My version is this – **[git version 2.45.2.windows.1]** - If we find error then we have to cross check, mine is working so non it to do that.

# Step : 2

If we are using very 1st time then we have to setup config -

What does this mean is that, we have to tell them that who am I ?

My name and email ID my identity, if we make any changes into it my team will get to know about it.

So lets set the My name and Email ID into the Git.

Now type line code which help to set my name into the Git. Config file is there inside Git which help us to make those changes.

<u>To set Name inside Git :</u>

[git config --global user.name “Vijay Gurung”]

Similar way we will going to set the our email ID :

[git config --global user.email “vijaygurung555@gmail.com”]

So, my name and email IDs both are successfully changes to cross check, we can run this below command. We will get the know about it.

[git config --global user.email]

[git config --global user.name]

# Step: 3

If incase we would like to make some change into it into email or name so for that we can follow below steps :

Git config hypin hypin (--) global hypen hypin (- -) edit .

```
git config --global --edit
```

Its looks like this

```
[user]
        name = Vijay Gurung
        email = vijaygurung555@gmail.com
```

After make changes hit **Esc or Escap bottom** after that below line Type - [: wq] **column wq** hit enter to come out from this. Now we are back to our normal code.where we was writing the code.

# Step 4 :

Now we can start working on projects.

We can open Visual studio Code or VS code side by side. It support Python, Java etc.

Do not do anything just open it.

Now we can type below code to my GitBase continue :

[mkdir LetsTryGit] what it does is that with the help of mkdir or make directory command it will create a new folder I will name it as a **LetTryGit**

Now next we will type **[cd LetsTryGit] So Cd or change directory** means which help me that, we will come to that folder which I have created just now as a name of **LetTryGit.**

Now come to the VS Code and here I have created a new file name it as

[Sum.Python]

Same place where my I have created directory. I have save that code inside file name and where my main **LetTryGit** Directory I have created**.**

I have created file name it as – [Sum.Python] but basically it's a Python code sum inside it.

[git status] - I have checked it suggested me to give the file name.

[git add Sum.Python] Our code is successfully working

After this again run this code to check status.

[git status]

After this my file started tracking earlier it suggested me to add file name. Now it stated tracking my file.

Staging Area under- where we can hold changes before final commit.

May be there is so many files where we need to commit. But I wanted to here just few 10-15 only allowed.

So, now here when we add my file is now in the staging area.

Now I am going to commit.

[git commit -m “initial commit”]

with the help of this command it will commit, my code. Here -m (hypin m) means we are trying to write some message after this inside this Double coma “ “. What changed we have done inside it we have to write some long message so another person will come and check this code he will understand what changes I have done.

Now

[git log] with the help of these it will show me where we have committed before.

This person has committed this things in this particular date and time.

Next,

So, we will do minor change to understand to this file – [**Sum.Python]**

Below code, we will make some changes.

```
public class Sum {

    public static void main(String[] args) {
        int a = 5;
        int b = 6;

        System.out.println(a + b);
    }
}
```

This is my current code and run this code

```
public class Sum {

    public static void main(String[] args) {
        int a = 5;
        int b = 6;

        System.out.println("The sum is" + (a + b));
    }
}
```

Come back to **Git** and enter this line of code

```
[ git status ]
```

Error message will get saying that some modification has done. This file - [**Sum.Python]**

Now I will add 1 more file into it. [Diff.Python]

I will not add anything now leave it as it is.

[Command + K]

with the help of it I can clear my terminal that I have written here clear it.That is my choice or I can continue write code.

I will come and type

[Git status] - it will tell me that

Modified - [**Sum.Python] & Untrack file is – [Diff. Python]**

Git does not know we have added one new file name as - **[Diff. Python]**

Now we will add this 2nd file

Command [git add Diff.Python] my file added.

[Diff.Python] This file ok

Now if we check the status of both.

[Git status]

As a message its saying that one is committed but another is still modified.

Note :- Just for understanding. Google it and install its a Google chrome extension. Do not type inside Git. leave it.

{ Fish shall } – It help to auto commit. Install it later

Now Next,

WE will going to type this command.

[git commit -m "adding Diff.Python"]

Next,

[git log]

Then after doing commit [git log] that we can see the actual change over there, here we can see that it showing me 2 commit message.

Adding Diff. Python &

Initial commit

One this we can observed here is that both # code is different different.

Note :- Just for understanding don't code inside Git.

SO, here one point to note is that, **{ Head – Master }.** This one we see one code. This is basically a pointer it move commit to commit.

Now If I will

Comment – [git Status] it showing that message that Modified Sum.Python it means its still not committed properly.

For that we can do

[git add Sum.Python] or else another way [gitadd .] If we do git add. (dot)

What will happened is that, All the files comes to the staging area.

Now next,

[git commit -m “added message in Sum.Python”]

Next

[git log]

As we can see here total 3 commit is done.

Lets try to understand that how we can discard all the changes, we can go back on time and made changes.

So as we can see we have total 3 commit.

Let copy this # Code

[c73633470b152cabd55dde8a539f75dca4558a74]

We will write then copy paste # Code

[git checkout c73633470b152cabd55dde8a539f75dca4558a74]

Because of these it will delete the, 2nd file which is - **[Diff.Python]**

If we will go & check in the VS code we

can see that It has delected the our 2nd

File [Diff.Python]

From the VS code.

With the help of check out we can do this things.

If we want to go back again then, we can write the command that.

[git checkout master]

Because of this command it will return me the both the files. Below is the outcome.

1. [Sum.Python"]
2. [Diff.Python]

So, lets type [command + K] with help of this we will delect all and go back to that again my last our previous command.

Now,

We came back to our 1st starting code

[Sum.Python]

```
public class Sum {

   public static void main(String[] args) {
      int a = 5;
      int b = 6;

      System.out.println(a + b);
   }
}
```

SO again do

[git checkout c73633470b152cabd55dde8a539f75dca4558a74]

If I will do it again checkout I will go back to my old command very 1st one.

Branch Master Branches we can make as much as we want to make it. Multiple branch,

Whatever changes we made new one old brach does not know about it

<u>Git barch</u> :

Merge inside it what is the benefit of it.

A lot of people are working on different different features and other will donot what is happing to others branches. What code changes going on.

<u>Git Ignore</u> :

Whatever folder we will craet git will ignore it.

From git to Github code push.

[git remote -v] (Hypen V)

Nothing shown

Copy paste below on in Git.

[git remote add origin https://github.com/Vijaygurung1/Lets-Try-Git.git]

Now if call

[git remote -v]

2nd command to push copy from GitHub

[git branch -M main]

Now paste to the Git Above code copy from GitHUb

Below is the my final Outcome

Pushed to GitHub successfully.

URL :- [https://github.com/Vijaygurung1/Lets-Try-Git]