Yesterday we have created a project using GitLab and then clone it to your local repo. In today’s session first, we will see what is difference between GitHub and GitLab.

**Core Difference**

The core difference is GitLab has Continuous Integration/Continuous Delivery (CI/CD) and DevOps workflows built-in. GitHub lets you work with the CI/CD tools of your choice, but you'll need to integrate them yourself. GitHub users typically work with a third-party CI program such as Jenkins, CircleCI, or TravisCI.

Another important difference is GitHub prioritizes speed, while GitLab focuses on reliability.

**Branches: merge or separate?**

Specifically, GitHub advocates merging new branches (your new and unique changes) with the master (main) branch. That way, you're set up to quickly deploy, and you can speedily reinstate your old version if something goes awry.

In GitLab's workflow, you create multiple yet separate stable branches beyond that of the master branch for each set of changes you make. At a minimum, you'll have production and pre-production stable branches. The multiple branch approach does require a multiple-step testing process. A single code review upon the merge request isn't enough.

**Software services**

Another core difference is that GitLab offers you a complete software development solution. They advertise themselves as a complete DevOps platform for a reason. That said, GitLab does offer integrations with some third-party programs and platforms such as Jira, Microsoft Teams, Slack, Gmail, and numerous other apps and platforms.

GitHub, on the other hand, offers fewer services within its own program but offers ways to integrate with many outside programs and services. These include software that GitHub has worked on to integrate with the service and hundreds of other programs via GitHub Marketplace.

For details, go through following url.

<https://www.zdnet.com/article/github-vs-gitlab-the-key-differences/>

You have already tried GitLab yesterday, I hope you remember that we have created a project in GitLab and cloned it to our working directory.

But now, we will focus on GitHub.

**So, as we have already seen configuration in GIT.**

**Review again:**

**To Configure GIT, we will use following commands:**

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo

$ git init

Initialized empty Git repository in C:/Users/Bhushanp5/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/.git/

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo (master)

$ git config --global user.name Bhushanp5

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo (master)

$ git config --global user.email bhushan.paradkar@citiustech.com

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo (master)

$ git config --global user.email

bhushan.paradkar@citiustech.com

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo (master)

$ git config --global user.name

Bhushanp5

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo (master)

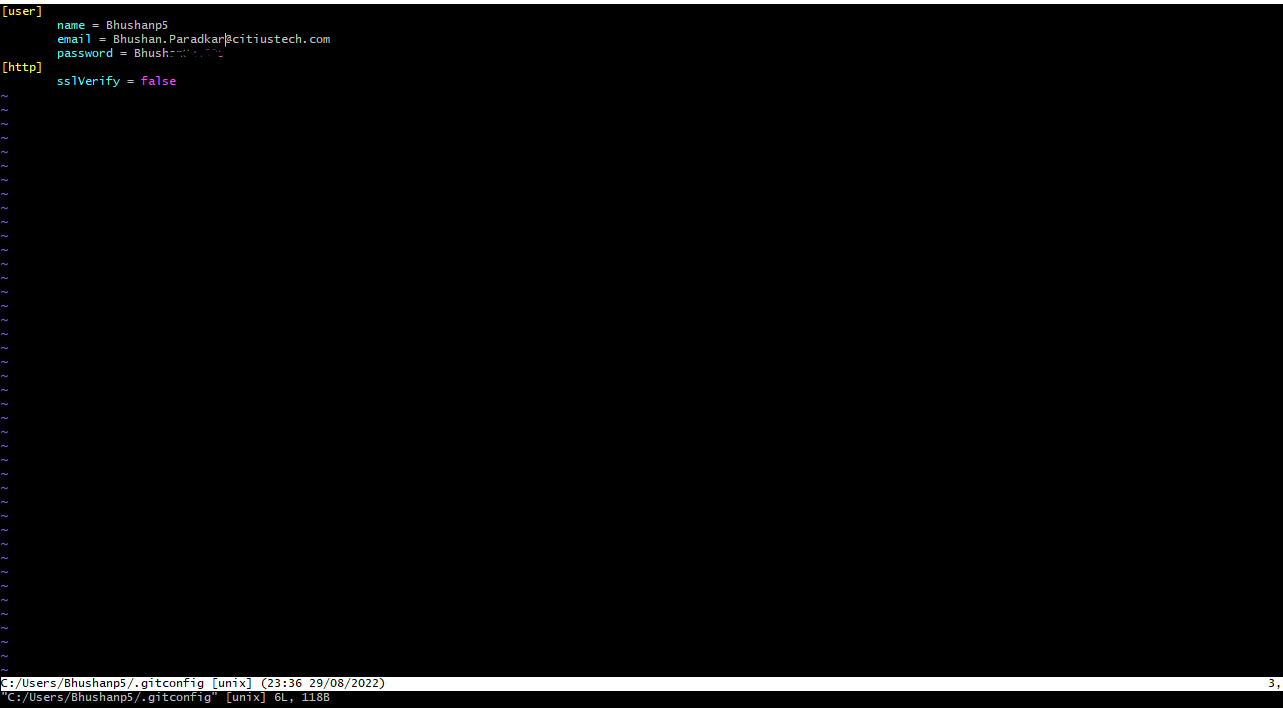
$

**Now, let’s assume if we will have to edit email or name then we can also edit it. For editing follow given below commands.**

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo (master)

$ git config --global --edit

**It will open vm editior then by pressing i we can do changes and if you will have to save changes then you will have to press esc then, : then wq, our file will get saved.**



So, now our git is configured. I already told you that GIT can work with any file it maybe c,c++,

Java, HTML and so on.

Now, let’s understand it well. First, we will use VS Code and create a project in it. For doing this we will fire given below commands.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo (master)

$ mkdir myTestProject

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo (master)

$ cd myTestProject

Now, you know very well how to open this folder with VS Code so open this folder in VSCode.

Once it opened then this project folder I want to make a repo then which command I should fire?

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git init

Initialized empty Git repository in C:/Users/Bhushanp5/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject/.git/

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ ls -a

./ ../ .git/

Our folder now become local repo. Right?

Now, let’s start writing some code in VS Code.

In our project, I have created a created a file index.html.

Now, which command we have seen yesterday to see what is status of our local repo to check whether any changes made in repo?

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

index.html

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

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$

Which gives us tracking on modifications.

Now, as discussed yesterday can you tell me which command we will have to use for adding any file to ready for staging?

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git add index.html

Now again fire command that is git status for checking current status of file.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git status

On branch master

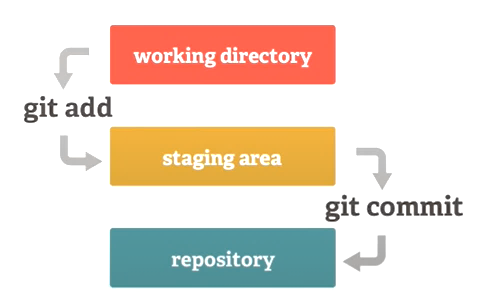
No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: index.html

So, from this we can conclude, using git add command we are putting our file in staging area.



In addition to Yesterday’s commands, we will see one more command now that is commit.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git commit -m "Index.html file completed"

[master (root-commit) 2a8f993] Index.html file completed

1 file changed, 12 insertions(+)

create mode 100644 index.html

So, here we have commit only one file. If you have to check how many files we committed so for this purpose, we have command that is git log.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git log

commit 2a8f9938b79eb098516e43d87e8aa99f3acb9d6a (HEAD -> master)

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:01:55 2022 +0530

Index.html file completed

Note: When we commit, at that time we are using -m “”. This is nothing but it is message we are adding to get information what is done or what purpose for this commit. So, message name should be meaningful.

Now, Again I will go to my VS Code and do changes in index.html. Again, fire command git status.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

no changes added to commit (use "git add" and/or "git commit -a")

So, here we can see the index.html file in which I made changes is notified as modified.

Now, I will add one more html file to our project. Let’s consider we have added aboutus.html. Now, again file command git status.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: index.html

Untracked files:

(use "git add <file>..." to include in what will be committed)

aboutus.html

no changes added to commit (use "git add" and/or "git commit -a")

Now, we can add both the files or we can add one by one. If I want to add one file, then you know how to add. So, you can try by adding both files by using following command.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git add --all

**Note** – We can use git add . also to add all

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: aboutus.html

modified: index.html

So, now both the files are added in staging area.

Now, we have to files to commit. So, we will do commit with a meaningful message.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git commit -m "Index and Aboutus.html done "

[master a43f105] Index and Aboutus.html done

2 files changed, 13 insertions(+)

create mode 100644 aboutus.html

Now, as discussed, we will check out log commit. How many files we committed so far.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git log

commit a43f105884f7a9df4235707cb44022a9933dec57 (HEAD -> master)

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:19:59 2022 +0530

Index and Aboutus.html done

commit 2a8f9938b79eb098516e43d87e8aa99f3acb9d6a

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:01:55 2022 +0530

So, here you can see. We committed twice and therefor in log e got it as result with message.

Now, we will see How we can use any specific commit.

Reason for using commit is, lets assume that we did some changes in existing file but if I want to get that version of my file then I can use Hashcode of that commit so, I can get previous version of my file.

There is simple command we have to use for this requirement.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git checkout 2a8f9938b79eb098516e43d87e8aa99f3acb9d6a

Note: switching to '2a8f9938b79eb098516e43d87e8aa99f3acb9d6a'.

You are in 'detached HEAD' state. You can look around, make experimental

changes and commit them, and you can discard any commits you make in this

state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may

do so (now or later) by using -c with the switch command. Example:

git switch -c <new-branch-name>

Or undo this operation with:

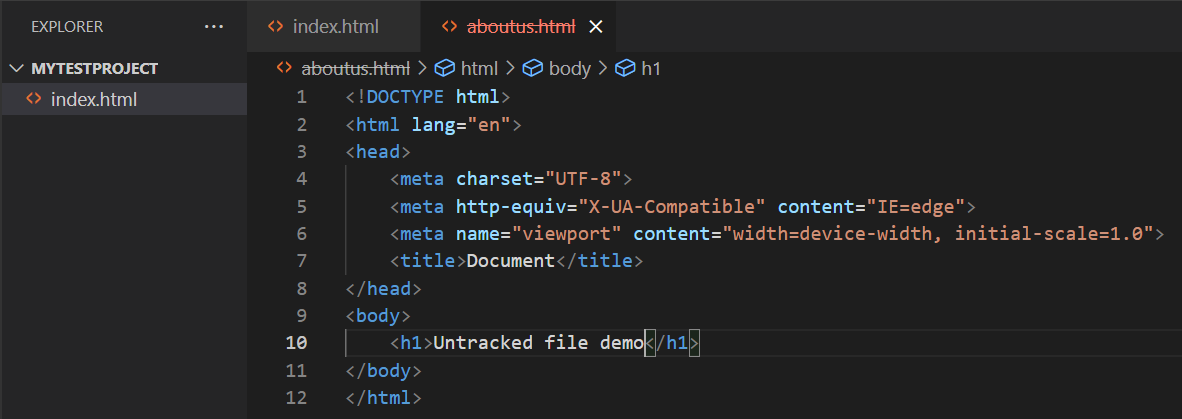
git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 2a8f993 Index.html file completed

(2a8f9938b79eb098516e43d87e8aa99f3acb9d6a – **Hashcode for commit**)

After hitting enter to this command, my new file will get deleted and I will reach to that point where I committed earlier.



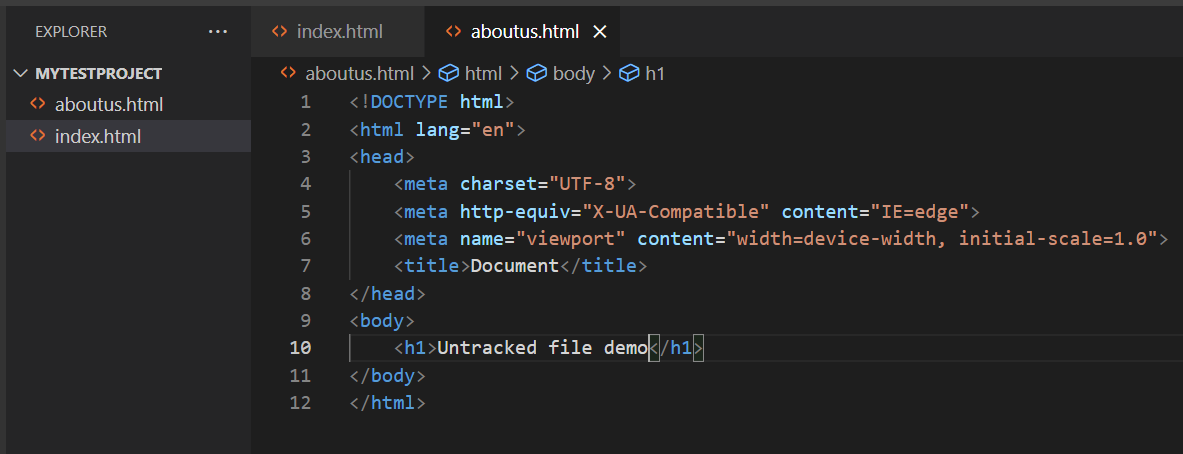
But again, I want to be at position where I have added aboutus.html file and committed. Then I will have to use same command but instead of Hashcode I will use master because as it’s my recent command so, my header/ master is there.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject ((2a8f993...))

$ git checkout master

Previous HEAD position was 2a8f993 Index.html file completed

Switched to branch 'master'



You can see, I got my file back.

Now, fire a command that is git log once more.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git log

commit a43f105884f7a9df4235707cb44022a9933dec57 (HEAD -> master)

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:19:59 2022 +0530

Index and Aboutus.html done

commit 2a8f9938b79eb098516e43d87e8aa99f3acb9d6a

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:01:55 2022 +0530

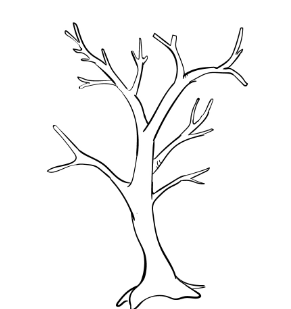
Index.html file completed

So, from above output we can conclude that we have committed twice and we got Hashcode for those both commits.

Using checkout command, we can go to that respective committed point which we have seen too.

We can also use this check out along with branching.

So before start branching can you tell me guys what exactly is branching?

 In general English we know what is branch. But usually all branches should have root from where it’s started. Correct? So, In Git also internally branch is same concept. There may have n number of branches but it main or root is master. For understanding I am simply typing a command.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git branch

\* master

**Que. Can we create our branches?**

Ans. Yes, we can create our branches.

But you might have question what is need for creating branches?

The reason is, let’s assume our team is working on a big project and there are n number of team members working on different module. So, if they are using master branch that is root branch and keep committing their files or we are creating some additional functionality and this commit we will do it in master branch then it will become more complex. So, as best practice we can create new branch then commit so, master will be untouched and once we are satisfied with implementation then we can merge branches so, other child branches will be merged with maser.

Let’s try to understand it with example. **Command 1 for understanding branching.**

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git branch

\* master

If we have multiple branches then using checkout we can navigate to another branch.

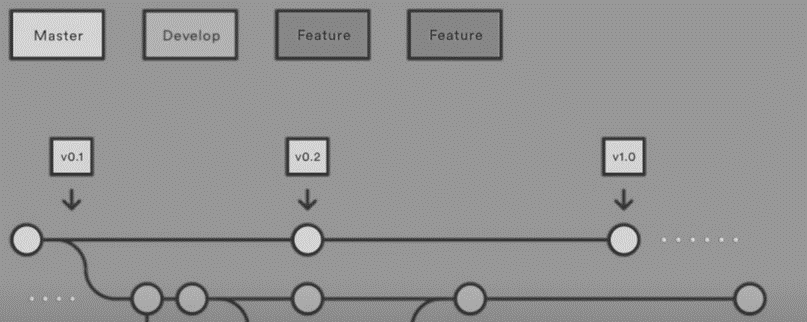
BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git checkout master

Already on 'master'

But right now, I have only one branch that is master. So, it’s giving me above output.

Let’s explore branching in depth.



As I told you there is root branch that is master, but we can also create new branch by ourselves for keeping master branch untouched. Usually people create different branches. Usually in development team we can create dev branch. Then again in dev branch we can create some nested branches. Like features.

**Let’s see how to create dev branch?**

Our target is to create a dev branch for development purpose.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git branch

\* master

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git branch dev

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git branch

dev

\* master

Now, we are in master branch. How we can get to know about it.

But we have created a branch dev.

Anybody is having idea, how I can move to dev branch.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git checkout dev

Switched to branch 'dev'

Now, we have created a new branch and changed branch too. But same thing I can do in one line only using checkout.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git checkout -b bhushan/services

Switched to a new branch 'bhushan/services'

can you tell me currently I am working in which branch?

So, currently I am in dev/bhushan/services branch and if I will work in this branch other branches won’t be affected.

Now, I will add a html file as services.html file in our project.

Now, using git status I will check status.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (bhushan/services)

$ git status

On branch bhushan/services

Untracked files:

(use "git add <file>..." to include in what will be committed)

services.html

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

I will add this file now using git add command.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (bhushan/services)

$ git add --all

Now, I will commit this file.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (bhushan/services)

$ git commit -m "services.html is added"

[bhushan/services 3823bb3] services.html is added

1 file changed, 12 insertions(+)

create mode 100644 services.html

Next step I am doing is I will change to master branch.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/

$ git checkout master

Switched to branch 'master'

As soon as I am going to change branch, you can’t see services.html file in VS code because we have changed branch and services.html file I have added and committed in bhushan/services branch.

For more, clear picture we will fire command git log which will gives us all commit info.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (master)

$ git log

commit a43f105884f7a9df4235707cb44022a9933dec57 (HEAD -> master, dev)

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:19:59 2022 +0530

Index and Aboutus.html done

commit 2a8f9938b79eb098516e43d87e8aa99f3acb9d6a

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:01:55 2022 +0530

Index.html file completed

So, here we can see there are three commits but we can see only two because last commit for services we did it in another branch.

Now, I think my services.html is completed and ready to go. I want to add it in master branch then what I am supposed to do for this.

We will have to use **merge** command.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git merge bhushan/services

Updating a43f105..3823bb3

Fast-forward

services.html | 12 ++++++++++++

1 file changed, 12 insertions(+)

create mode 100644 services.html

Here we can say these two branches are merged and now, if I will use git log then I will get all commits here only.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git log

commit 3823bb31e926855f09595e218ff6ce56802e4b36 (HEAD -> dev, bhushan/services)

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 12:10:08 2022 +0530

services.html is added

commit a43f105884f7a9df4235707cb44022a9933dec57 (master)

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:19:59 2022 +0530

Index and Aboutus.html done

commit 2a8f9938b79eb098516e43d87e8aa99f3acb9d6a

Author: Bhushanp5 <Bhushan.Paradkar@citiustech.com>

Date: Tue Aug 30 00:01:55 2022 +0530

Index.html file completed

So, in short, we can say branching is very useful when we will have to let our teams working on different modules and others doesn’t let bother others then they can create such branch then merge with master branch.

**Git ignore File.**

There might be scenarios where you need a file in your project but you don’t want to let git tracks that file then for this purpose we have provision. Let’s assume there is a file in our project which I want just for my reference and I don’t want to let it track by git. Let’s see progrmatically.

I have added a file as myCredential.txt in which I kept my credentials and this file I don’t want to track by git.

It’s very simple.

**Step 1.** Create a file as .gitignore

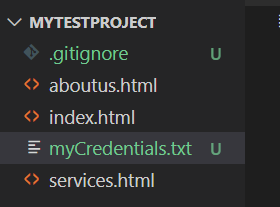
There are two ways for doing this.

1. By using command touch .gitignore

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ touch .gitignore

1. Simply create a file and rename it as .gitignore



So, now whichever file you don’t want to track by git those all files you can specify in .gitignore.

Let’s test it. Before adding myCredentials.txt file into .gitignore and after adding these files in .gitignore

**Before:**

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git status

On branch dev

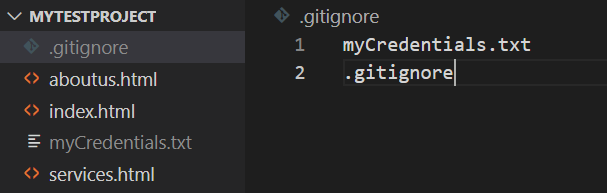
Untracked files:

(use "git add <file>..." to include in what will be committed)

.gitignore

myCredentials.txt

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



**After:**

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git status

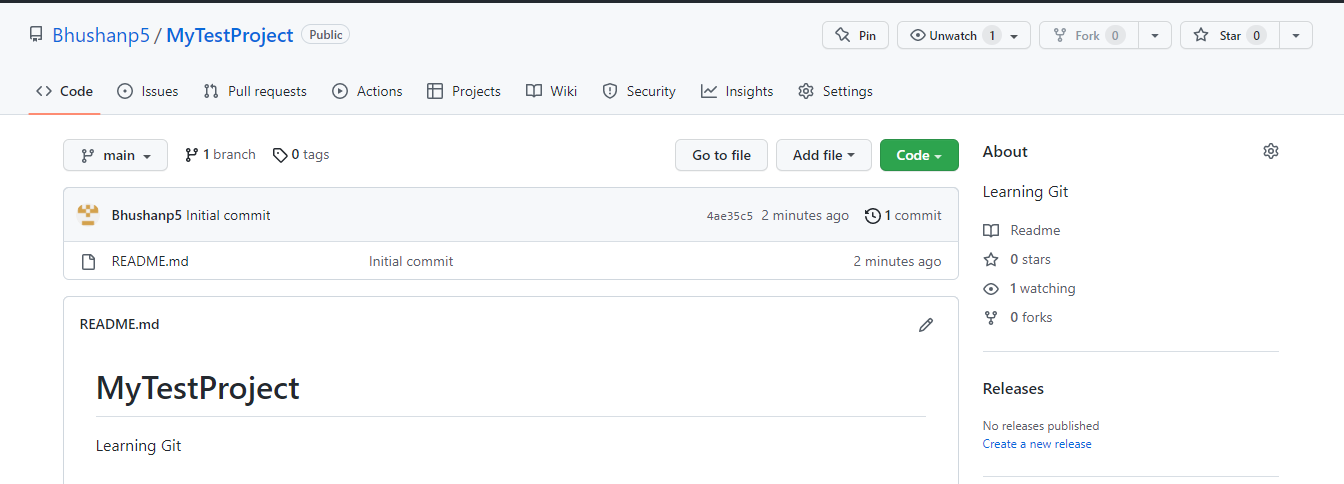
On branch dev

nothing to commit, working tree clean.

**Let’s discuss about GitHub now**. Yesterday we have seen Gitlab. It is going to work in same fashion but there are technical differences which I mentioned on top.

Steps to create new repo.

1. Click on right hand corner and select new repository.
2. Then give name for repository.
3. And create repository.



1. This project will look like this.
2. Next, we will connect with this project using local git which is at our system.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git remote -v

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git remote add origin https://github.com/Bhushanp5/MyTestProject

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git remote -v

origin https://github.com/Bhushanp5/MyTestProject (fetch)

origin https://github.com/Bhushanp5/MyTestProject (push)

Now, we can see commands for fetching as well as for push.

Now let’s push our local repo to remote one.

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git remote -v

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git remote add origin https://github.com/Bhushanp5/MyTestProject

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git remote -v

origin https://github.com/Bhushanp5/MyTestProject (fetch)

origin https://github.com/Bhushanp5/MyTestProject (push)

BhushanP5@IMCCBCP48-MSL1 MINGW64 ~/Desktop/MyData/MyDrive/OneDrive - CitiusTech/Freshers Training July 2022/GIT/Day 2/Demo/myTestProject (dev)

$ git branch -M master