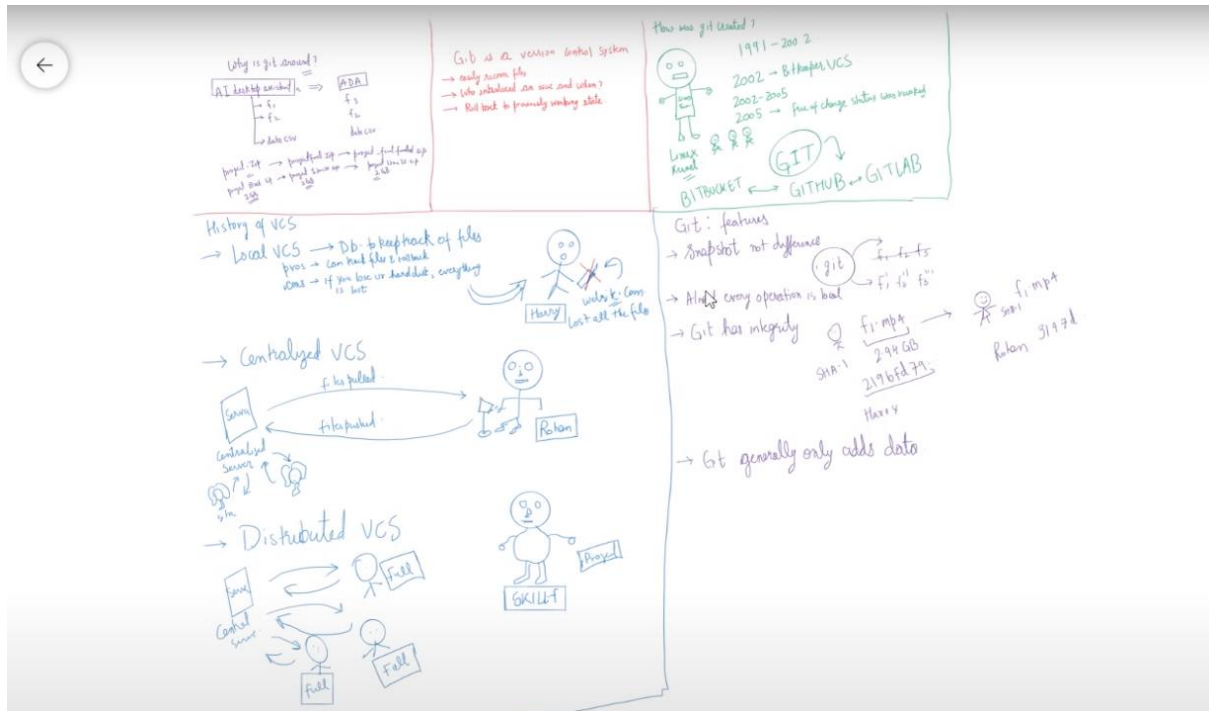


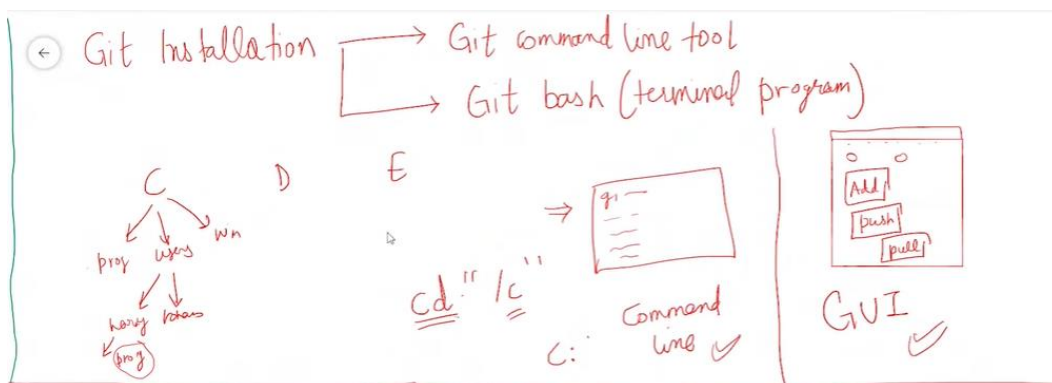
🔥 Git & GitHub 🔥

Day 1 = You Can Check image file To Understand What Is Git or GitHub & Why We Need Git 🔥.

→ Git is a Version Control System ✓



Day 2 = Here We Install or Setup Git Bash in Our Local System You Can See in Image. 😊



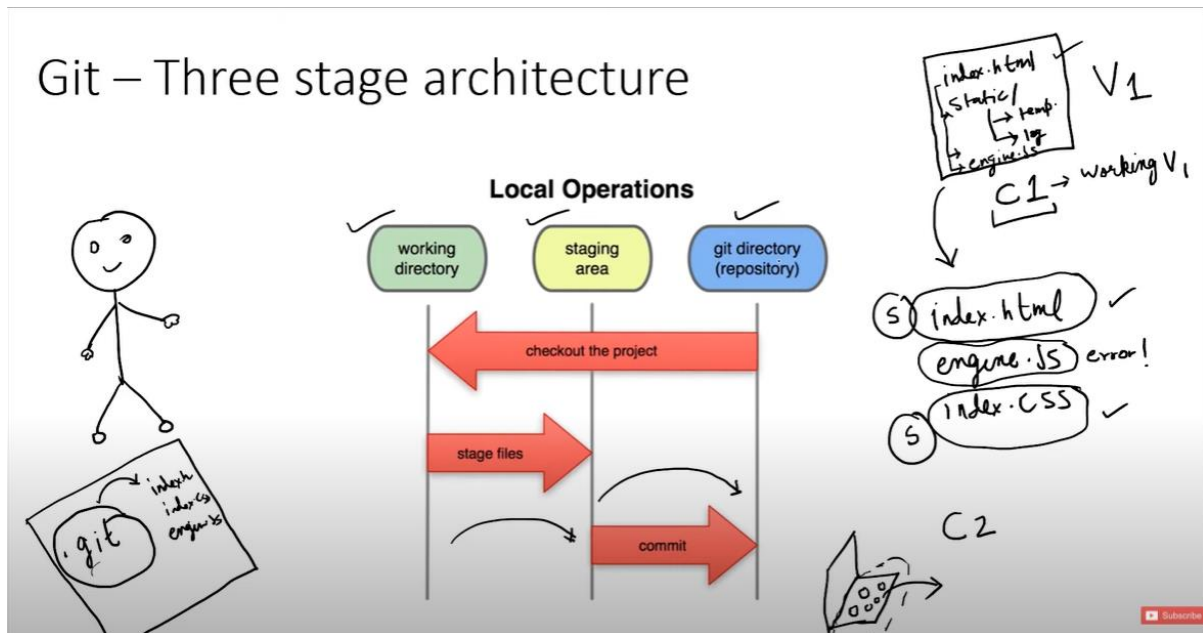
- **Open New Folder With Git Bash & Now Setup Git Bash.** 🔥

1.Now Set User Name & Email For What Your Commit in You Git Repo. 😊

- **(pwd , cd , ls) – Basic Command Of Linux**
- **git config --global user.name "Satya"**
- **git config --global user.email "vikashraj979898@gmail.com"**
- **git config --list**
- **git config --global core.editor vim -- for choose You Editor In Linux**
- **git config user.name ----For Show Your Name in Git Bash**
- **git config user.email ----For Show Your email in Git Bash**

Day 3 = Here We Know About Git Three Stage Architecture 😊

Git – Three stage architecture



→ Working Dir --- Staging Area --- Git Commit ✓

Day 4 = Tracking Our first Git Project 🔥

git status = To Know About status of Your Dir.

git init = To Initialized a Git Repo.

git status = To Know About status of Your Dir.

Check Out Fig. 😊

```
$ git status
fatal: not a git repository (or any
of the parent directories): .git

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials
$ git status
fatal: not a git repository (or any of the parent directories): .git

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials
$ git init
Initialized empty Git repository in C:/Users/Haris/Desktop/code playground/Tuts/Git tutorials/.git/

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    db.acddb
    first.txt
    myexcel.xlsx

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

git add --a = For Staging Your File in Git Repo.

git status = To Know About status of Your git Dir.

Check Out Fig. 😊

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git add --a

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   db.accdb
    new file:   first.txt
    new file:   myexcel.xlsx
```

git commit -m "Your Msg" == Time To Commit Your File After Complete .

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git commit -m "Initial Commit"
[master (root-commit) 43699a1] Initial Commit
3 files changed, 1 insertion(+)
create mode 100644 db.accdb
create mode 100644 first.txt
create mode 100644 myexcel.xlsx
```

git status = Know About Status Of Your Git Dir

git log = Know About What You Commit In Your Git

Check Out Fig. 😊

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git log
commit 43699a1faf937e34aa741f83c42d11c361d7c271 (HEAD -> master)
Author: Harry <harry@codewithharry.com>
Date:   Wed Oct 23 15:44:02 2019 +0530

    Initial Commit
```

After Changes In Your File Again We Have To Commit Using Command 🔥

- 1 . git status – Run This Command.**
- 2. git add yourfile.txt – For Single File Commit.**
- 3 . git status - Run This Command.**
- 4 . git commit -m "Your Msg 2"**

rm -rf .git – For Remove You Git Repo (X).

git status

5. Cloning a Remote Git Repository 😊

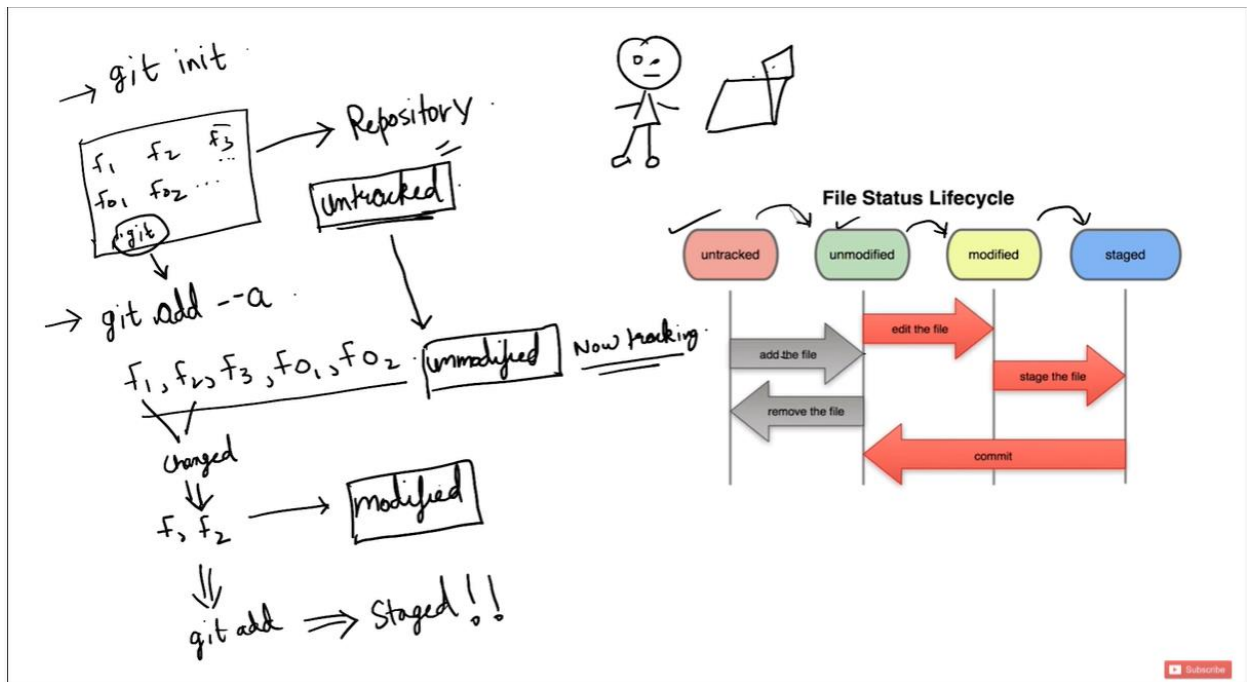
For Cloning Any Public Git Repo You Can Use This Command & Also Select Folder Carefully 😁

After Then Tensorflow Repo In Your Local System Now You Can Edit File Add Etc and After That You Can Stage This Repo And Commit It By Using Following Commands As We Know That. & You Can Also See Log Using (git log) and See Log Of Tensorflow. 😊

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials
$ git clone https://github.com/tensorflow/tensorflow.git
Cloning into 'tensorflow'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (29/29), done.
Receiving objects: 17% (126578/744574), 86.02 MiB | 10.46 MiB/s
```

→ Git clone (Your Git Repo Link).

6.File Status Lifecycle 😊



7. .gitignore: Ignoring Files 😊

We Make (.gitignore) For ignore File & Folder Create By Mistake or Created By Your App. Just Create a .gitignore and Add a File To Ignore. Lets Get Start . 😊

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ touch error.log

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    error.log

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ touch .gitignore
```

Touch Command For Create File & Also Create .gitignore File

As You Can See 😁


```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        error.log

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git add --a

```

Open .gitignore File & Write error.log After That Check Status You Can See error.log File Ignore By Git Repo. And Stage Files & Commit It. Using Commit Command. 😊

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        error - Copy (2).log
        error - Copy (3).log
        error - Copy (4).log
        error - Copy (5).log
        error - Copy (6).log
        error - Copy (7).log
        error - Copy.log

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

```

Here You Can Ignore Many Files By Using *.log in You Ignore File. 😊

8. Git Diff: Showing Changes Between Commits/Staging Area & Working Directory 😊

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (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:   .gitignore
        modified:   static/tex.txt

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git add .

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore
        modified:   static/tex.txt

```

➤ **Git diff - Command is For Compare With Staging Area To Working Directory .**

```

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:   static/tex.txt

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git diff
diff --git a/static/tex.txt b/static/tex.txt
index 9118d6c..553a5c4 100644
--- a/static/tex.txt
+++ b/static/tex.txt
@@ -1,1 @@
-asdasd
\ No newline at end of file
+harry bhai
\ No newline at end of file

```

git diff --staged – Command is For Compare With Pervious Commit To Current Staging Area. 😊


```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git diff

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git diff --staged
diff --git a/.gitignore b/.gitignore
index a135832..bf00e86 100644
--- a/.gitignore
+++ b/.gitignore
@@ -1,2 +1,3 @@
*.log
-dir/
\ No newline at end of file
+/dir/
+static/dir
\ No newline at end of file
diff --git a/static/tex.txt b/static/tex.txt
index e69de29..553a5c4 100644
--- a/static/tex.txt
+++ b/static/tex.txt
@@ -0,0 +1 @@
+harry bhai
\ No newline at end of file

```

9. Skipping the Staging Area. 😊

**If We Want to Skip Staging Area Then Run This Command –
git commit -a -m “Your Message Here”**

If We Change any File in Git Repo Then We Can Commit That File Without Going to Staging Area. Using Upper Command.

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

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/
Git tutorials (master)
$ git commit -a -m "Direct commit"
[master b53f569] Direct commit
1 file changed, 2 insertions(+), 1 deletion(-)

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/
Git tutorials (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        second.txt

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/
Git tutorials (master)
$
```

But If We Add New File or Something Then We Have to Staged That File Otherwise That File Not Going To Commit 😊

10. Moving and Renaming Files in Git. 😊

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git rm third.txt |
rm 'third.txt'

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    third.txt

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git commit -m "removed bad file third.txt"
[master 1b15ec9] removed bad file third.txt
1 file changed, 4 deletions(-)
delete mode 100644 third.txt
```

Here We Know How to Remove a File Using Git. 😊

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git mv first.txt first_renamed.txt

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:   first.txt -> first_renamed.txt

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git commit -m "renamed"
[master 275bcf2] renamed
1 file changed, 0 insertions(+), 0 deletions(-)
rename first.txt => first_renamed.txt (100%)

```

→ Here We Know How to Rename a File Using Git. 😊

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (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:   db.accdb

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git rm --cached db.accdb
rm 'db.accdb'

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    db.accdb

```

→ Here If We Want to Untracked Any File Then Use Command. (git rm --cached Your File Name). 😊

11. Git Log: Viewing & Changing Commits in Git 😊

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ rm -rf .git

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials
$ git status
fatal: not a git repository (or any of the parent directories): .git

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials
$ git clone https://github.com/pandas-dev/pandas.git mypanda
Cloning into 'mypanda'...

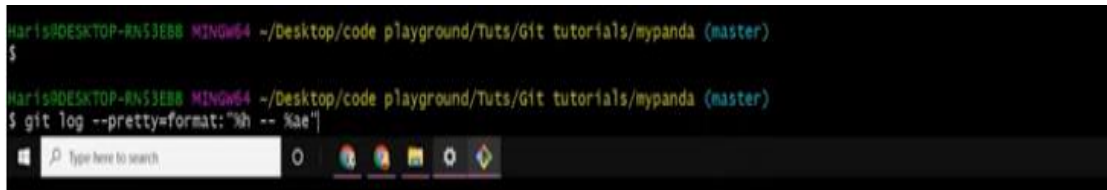
```

Important Commands 🧑

→ git log – See All Commit in Your Repo.

→ git log -p – Commit with Remove Line

- **git log -p -n** – Where N is No. of Commit You Want to See
- **git log --stat** – Short Msg About Commits.
- **git log --pretty=oneline** – Commit in One Line.
- **git log --pretty=short**
- **git log --pretty=full**
-
- **Filter in Your Git Repo.**
- **git log --since=2.days**
- **git log --since=2.months**
- **git log --since=2.year**

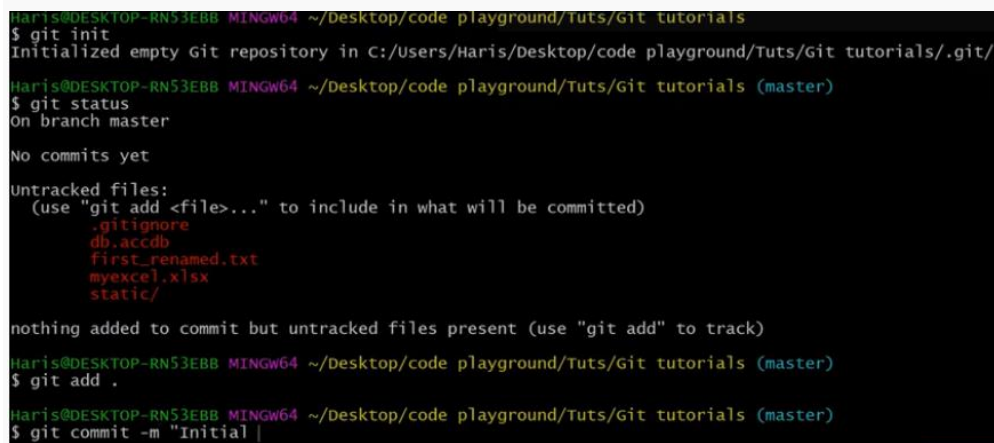


```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/mypanda (master)
$
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/mypanda (master)
$ git log --pretty=format:"%h -- %ae"

```

12. Unstaging & Unmodifying Files in Git. 🐼



```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials
$ git init
Initialized empty Git repository in c:/Users/Haris/Desktop/code playground/Tuts/Git tutorials/.git/

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        db.accdb
        first_renamed.txt
        myexcel.xlsx
        static/

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git add .

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git commit -m "Initial |

```

If We Want to Unstaged Our File Then See in Fig.

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   first_renamed.txt

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git restore --staged first_renamed.txt

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (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:   first_renamed.txt

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

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ |

```

If We Want To UnModifiy Then Use That Command.

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (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:   first_renamed.txt

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

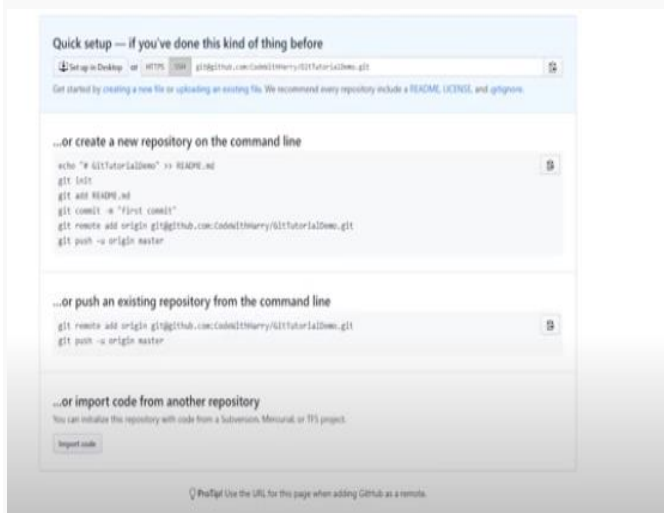
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git checkout -- first_renamed.txt

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$

```

13. Working with Remote Repositories. 🤝



Make a Repo. 🤝

Now We Can Push Our Code to Remote Server 1ST We Have To Make GitHub Account & Follow These Step 😊


```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git remote

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ AC

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git remote add origin git@github.com:CodeWithHarry/GitTutorialDemo.git

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git remote
origin

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git remote -v
origin  git@github.com:CodeWithHarry/GitTutorialDemo.git (fetch)
origin  git@github.com:CodeWithHarry/GitTutorialDemo.git (push)

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ |

```

Now We Are Going to Push Our Repo To Our GitHub Then You Will Received a Error Then 1st You Have To Get Permission. 🙏

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git push -u origin master
ERROR: Permission to CodeWithHarry/GitTutorialDemo.git denied to haris989.
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

```

So We Have To 1st Generate SSH key Of Your Local Computer Then Set Into Your GitHub Account 🤖

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ ssh-keygen -t rsa -b 4096 -C "harry@codewithharry.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Haris/.ssh/id_rsa):
/c/Users/Haris/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Saving key "/c/Users/Haris/.ssh/id_rsa" failed: Permission denied

```

And Add SSH Key to Your GitHub Account.

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git push -u origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (11/11), 26.22 KiB | 536.00 KiB/s, done.
Total 11 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:CodeWithHarry/GitTutorialDemo.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)

```

Now We Can Push Our Files to Our GitHub Account 🤖

Now Refresh Your GitHub Page..

14. Setting Alias in Git 🤖

Alias Means Shortcut of Your Big Command We Can Set a Long Command as Shortcut.

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git st
git: 'st' is not a git command. See 'git --help'.

The most similar commands are
    status
    reset
    stage
    stash
    svn

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git config --global alias.st status

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git st
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
```

As You Can See in Figure That We Set (git st) Command as (git Status) Using This command.

→ **git config --global alias.st status**

→ **git config --global alias.ci commit**

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git config --global alias.ci commit

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git ci
On branch master
Your branch is up to date with 'origin/master'.

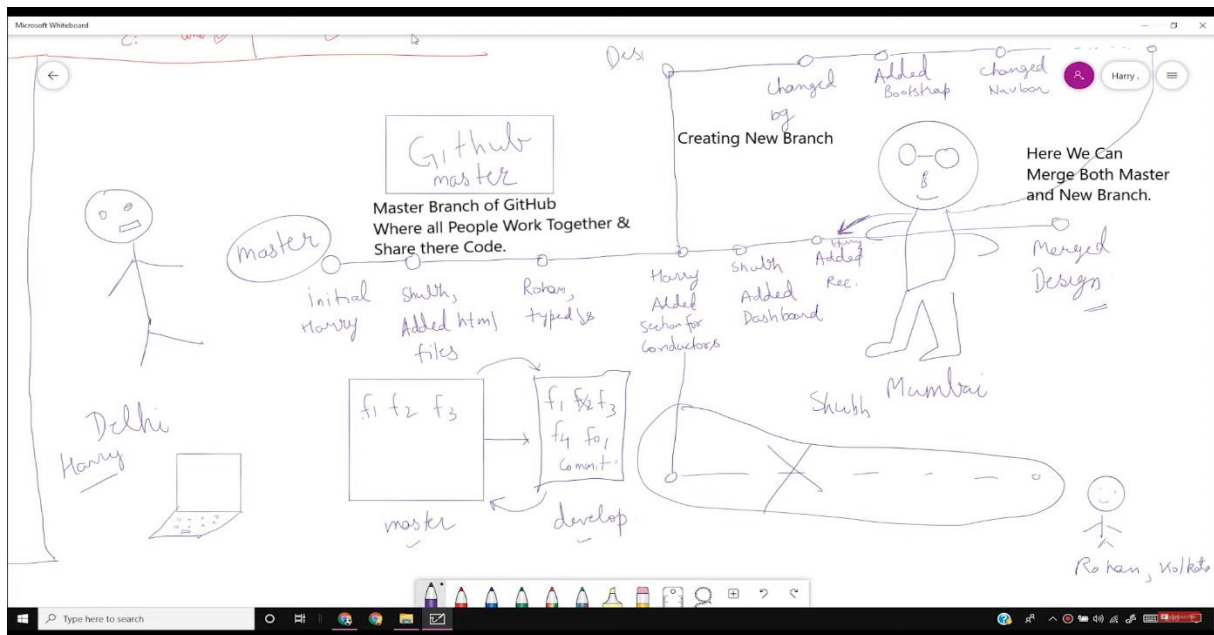
nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git commit
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

As You Can Set All Alias As You Want. 😁 ✓

15. Creating & Switching Branches in Git.



```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (
master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (
master)
$ git checkout -b develop
Switched to a new branch 'develop'

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (
develop)
$

```

Here We Create New Branch Called “develop” and Change To develop Branch

After The Change in Your Local System.

As Some Time When We Want To Change Into Our Master Branch Then ✓

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (develop)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$

```

You Can See Something Change Into Your Local Computer.

16. Branching & Merging a Production Grade Project 🧑‍💻

1st Download Vs Code in Your Local System . 😊

```

Harris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/MyWebsite (
master)
$ git branch
* master
  trycleanup

```

git branch for Know Total Branches in Your Git Repo.

```

Harris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/MyWebsite (
master)
$ git checkout trycleanup
Switched to branch 'trycleanup'

Harris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/MyWebsite (
trycleanup)
$

```

git checkout “yourbranchname” – To Move to Other Branches.

Now We Understand How to Merge Master With New Branches 😊

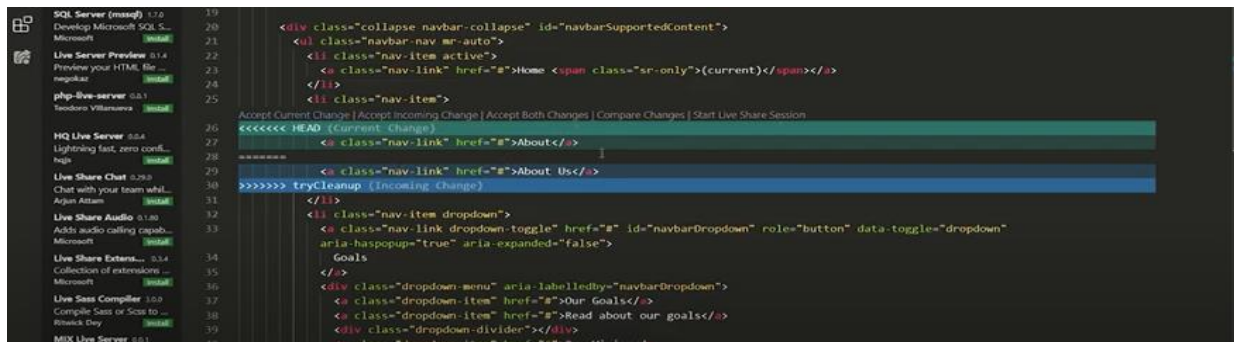
```

Harris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/MyWebsite (
master)
$ git merge trycleanup
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

Harris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/MyWebsite (
master|MERGING)
$ |

```

Here We Have To Select ✓



```

Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/MyWebsite (
master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   index.html

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

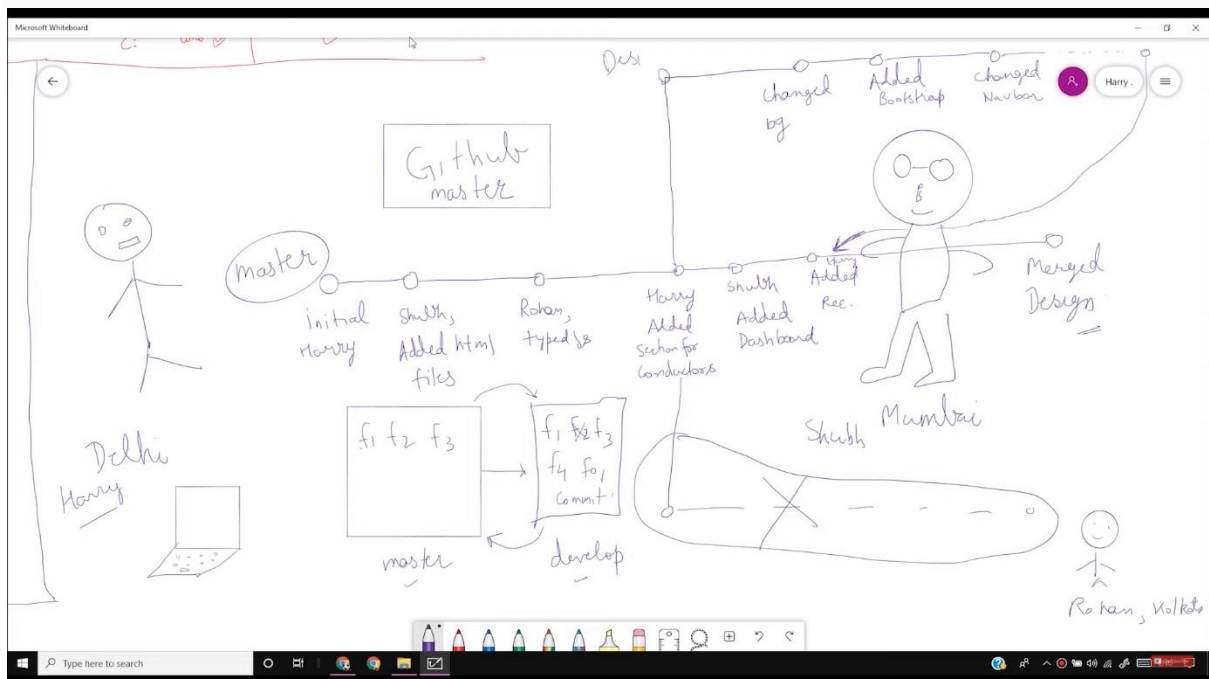
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/MyWebsite (
master|MERGING)
$ git add .

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials/MyWebsite (
master|MERGING)
$ git commit
hint: Waiting for your editor to close the file... |

```

Here Now We Merge Both Master & New Branch 😁✓.

17. Resolving Merge Conflicts 🤖



← Branch management

git branch	git branch -v
* master	* master Commit hash: 97b13c Commit message: "This will fix"
develop	develop
system	

git branch --merged → already merged branches.
 git branch --no-merged → not " " "

Deleting branches:
 git branch -d develop ⇒ gives ~~was~~ error if develop is not merged
 " " -D ⇒ no error & branch gets deleted

Now Lets Practice :

Just Do Simple Commit As We Do in Pervious ..

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
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)

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git add .

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git commit -m "added index.html"
[master 2997f8a] added index.html
1 file changed, 12 insertions(+)
create mode 100644 index.html
```

```
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (issue1)
$ git commit -a -m "this is c4"
[issue1 47ed348] this is c4
1 file changed, 1 insertion(+), 1 deletion(-)

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (issue1)
$ |
```

```

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git checkout -b issue1
Switched to a new branch 'issue1'

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (issue1)
$ git add -a -m "this is c4"
error: unknown switch `a'
usage: git add [<options>] [--] <pathspec>...

    -n, --dry-run          dry run
    -v, --verbose          be verbose

```

When We Want To Merge Two Branch

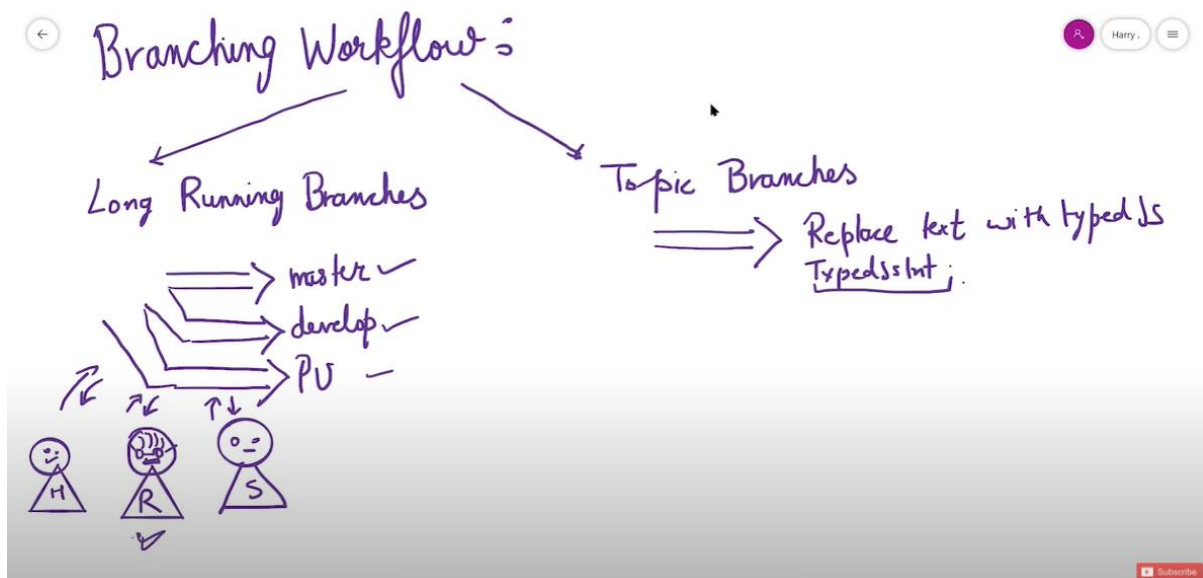
```

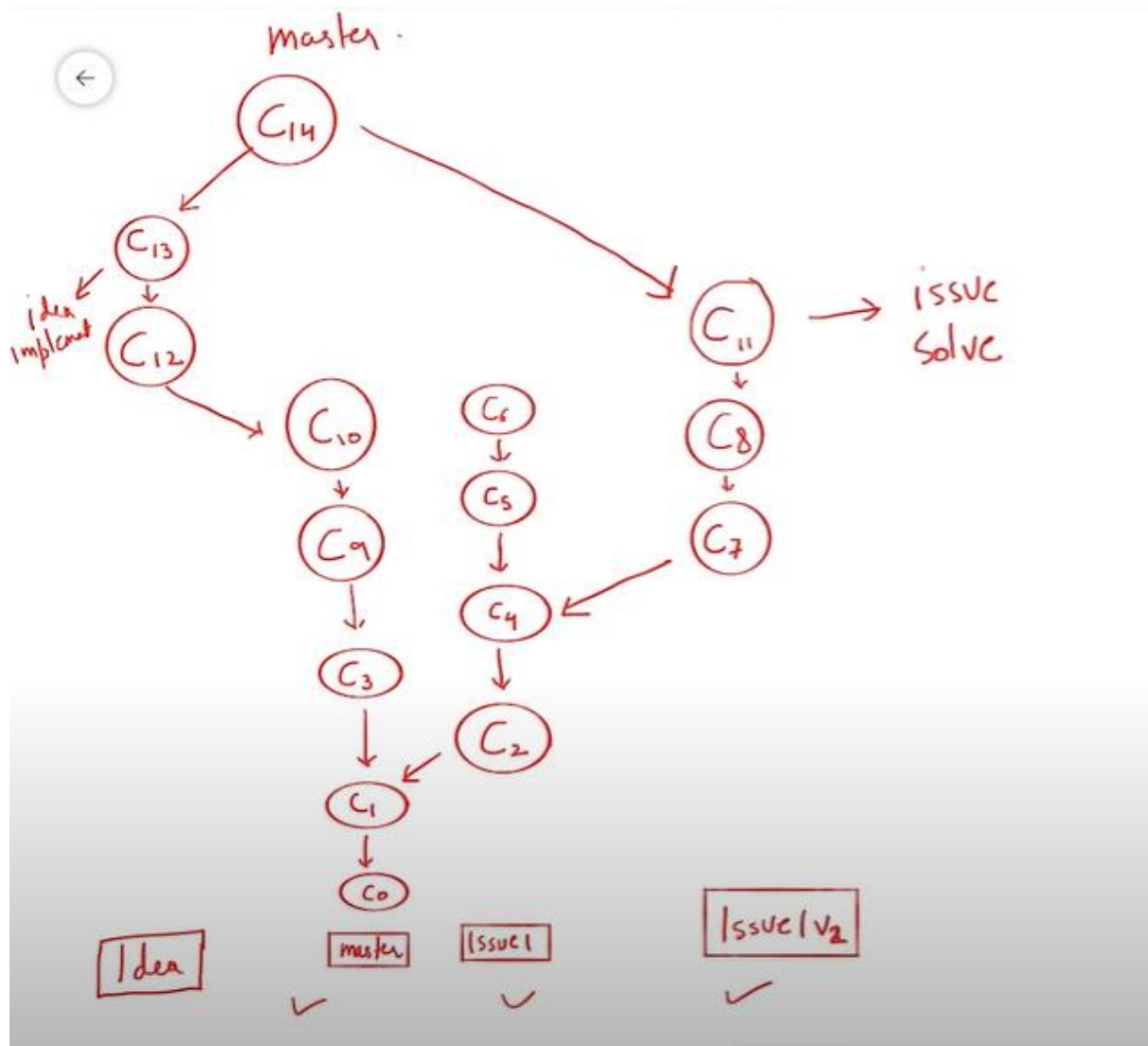
Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master)
$ git merge issue1
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

Haris@DESKTOP-RN53EBB MINGW64 ~/Desktop/code playground/Tuts/Git tutorials (master|MER
GING)
$ |

```

18.Git Branching Workflow in Production





19. Pushing Git Branches To Remote Repositories 😊

```

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git branch
* master

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git checkout -b bugfix

```

Here We Create New Branch Called bugfix. 😊

```

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git branch
* master

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git checkout -b bugfix
Switched to a new branch 'bugfix'

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (bugfix)
$ git branch
* bugfix
  master

```

Change Something in Your Ripo. 😊

```

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git branch
  bugfix
* master

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git remote
origin

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git remote -v
origin https://github.com/CodewithHarry/DemoRepository.git (fetch)
origin https://github.com/CodewithHarry/DemoRepository.git (push)

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git remote add origin https://github.com/CodewithHarry/DemoRepository.git
$ git remote
origin

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git remote -v
origin https://github.com/CodewithHarry/DemoRepository.git (fetch)
origin https://github.com/CodewithHarry/DemoRepository.git (push)

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git push -u origin master

```

Here We Push Our Master Branches to Our Repository.
Now let See How Can Be Push Our Branches in Our Repository. 😊

```

$ git branch
  bugfix
* master

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

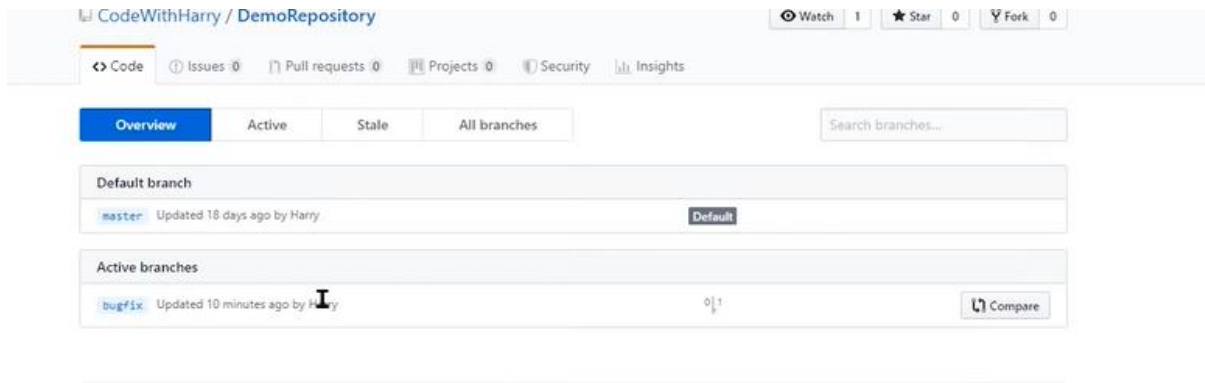
Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git checkout bugfix
Switched to branch 'bugfix'

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (bugfix)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git checkout bugfix
Switched to branch 'bugfix'

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (bugfix)
$ git push origin bugfix

```



You Can Merge Manually Using GitHub. 😊

```
Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (bugfix)
$ git status
On branch bugfix
nothing to commit, working tree clean

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (bugfix)
$ git push origin bugfix:mybugfix
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: create a pull request for 'mybugfix' on GitHub by visiting:
remote:   https://github.com/CodeWithHarry/DemoRepository/pull/new/mybugfix
remote:
To https://github.com/CodeWithHarry/DemoRepository.git
 * [new branch]      bugfix -> mybugfix
```

Here We Push New Branch Name OF Pervious Branch Name as You Can See in Fig. 🙌

```
Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git branch
  bugfix
* master

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git branch -d bugfix
error: The branch 'bugfix' is not fully merged.
If you are sure you want to delete it, run 'git branch -D bugfix'.
```

Here if We Want To Delete Our Branch Then Use This Command.

➤ **Git branch -d your_branch_name.**

Here We Know How To Delete Origin Branches.

```
Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git branch
* master

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/CodewithHarry/DemoRepository.git
6092c46..43df330 master -> master

Haris@DESKTOP-RN53EBB MINGW64 /d/Business/code playground/Tuts/Git tutorials (master)
$ git push -d origin bugfix
To https://github.com/CodewithHarry/DemoRepository.git
- [deleted]          bugfix
```

And Refresh You GitHub Repository. 🙌😊

Happy Coding 🙌😊

Satya.py - IG