

Git Hub Configuration

`git --version`

`git config --global user.name "NessWrites"`

`git config --global user.email "np01ai4s230009@islingtoncollege.edu.np"`

`git config --list`

(user name verified)

Now open vs code

go to terminal, make new folder, open that folder in the terminal

Clone & Status (command)

Clone- cloning a repository on our local machine

`git clone <some link ->`

In terminal type gitclone and paste the link from github file copied link

Now the file is copied in your location

use cd to go into the directory of the file

type `ls` for list of files

type `ls -a` for list of hidden files

Status - displays the status of the code

type `git status`

When you modify a downloaded file from git hub , you need to do two things

- i. add "`git add <file_name>`" —>staging area
- ii. commit

staged(file is ready to be committed) and
modified(changed)

unmodified(unchanged)
untracked(new files git doesn't yet track)

```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE    COMMENTS

On branch main
Your branch is up to date with 'origin/main'.

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:   readme.txt

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

when untracked file is added

```
● (base) ness@Nesss-Laptop La-Strada % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   new.html

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:   readme.txt

○ (base) ness@Nesss-Laptop La-Strada % █
```

when all files are to be added type "git add ."

```
● (base) ness@Nesss-Laptop La-Strada % git add .
● (base) ness@Nesss-Laptop La-Strada % git status
On branch main
Your branch is up to date with 'origin/main'.
```

Changes to be committed:

```
(use "git restore --staged <file>..." to unstage)
  new file:  new.html
  modified:   readme.txt
```

to message

git commit -m "How "you doing"

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE COMMENTS

```
Your branch is up to date with 'origin/main'.
```

Changes to be committed:

```
(use "git restore --staged <file>..." to unstage)
  modified:   README.md
  new file:   name.txt
```

```
● (base) ness@Nesss-Laptop CourseWork % git commit -m"How you doing?"
[main ffc8419] How you doing?
 2 files changed, 2 insertions(+)
 create mode 100644 name.txt
```

 NessWrites	How you doing?	ffc8419	1 minute ago	 2 commits
 README.md	How you doing?		1 minute ago	
 name.txt	How you doing?		1 minute ago	

README.md 

CourseWork

Islington College Information System Hallo

Push Command

`git push origin main`

origin means github remote repository origin copy name
main is the branch

Git init Command

The git init command is used to create a new, empty Git repository or to reinitialize an existing one. It turns any directory into a Git repository. This is done by creating a hidden directory called .git that stores all of the objects and refs that Git uses and creates as part of your project's history. This hidden .git directory is what separates a regular directory from a Git repository

init - used to create a new git repo

`git init`

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git init
Initialized empty Git repository in /Users/Pleasure/python_CW_firstyear/.git/
● (base) ness@Nesss-Laptop python_CW_firstyear % ls -a
.   ..   .git
● (base) ness@Nesss-Laptop python_CW_firstyear % git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    database.txt
    inventories.txt
    main.py
    operations.py
    read.py
    returns_database.txt
    write1.py

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

`git add .`

`git status`

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git add .
● (base) ness@Nesss-Laptop python_CW_firstyear % git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file: database.txt
    new file: inventories.txt
    new file: main.py
    new file: operations.py
    new file: read.py
    new file: returns_database.txt
    new file: write1.py
```

git commit

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git commit -m"Commercial program"
[main (root-commit) 93cd404] Commercial program
 7 files changed, 882 insertions(+)
 create mode 100644 database.txt
 create mode 100644 inventories.txt
 create mode 100644 main.py
 create mode 100644 operations.py
 create mode 100644 read.py
 create mode 100644 returns_database.txt
 create mode 100644 write1.py
```

git remote add origin <-link->

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git remote add origin https://github.com/NessWrites/Commercial_Website_CW.git
```

—> no changes will happen do git remote simultaneously

git remote -v (to verify remote)

```
      remote
● (base) ness@Nesss-Laptop python_CW_firstyear % git remote -v
origin  https://github.com/NessWrites/Commercial_Website_CW.git (fetch)
origin  https://github.com/NessWrites/Commercial_Website_CW.git (push)
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch
 * main
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch -M Main
```

git branch (to check branch)

```
      remote
● (base) ness@Nesss-Laptop python_CW_firstyear % git remote -v
origin  https://github.com/NessWrites/Commercial_Website_CW.git (fetch)
origin  https://github.com/NessWrites/Commercial_Website_CW.git (push)
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch
* main
```

git branch -M main(to rename branch)

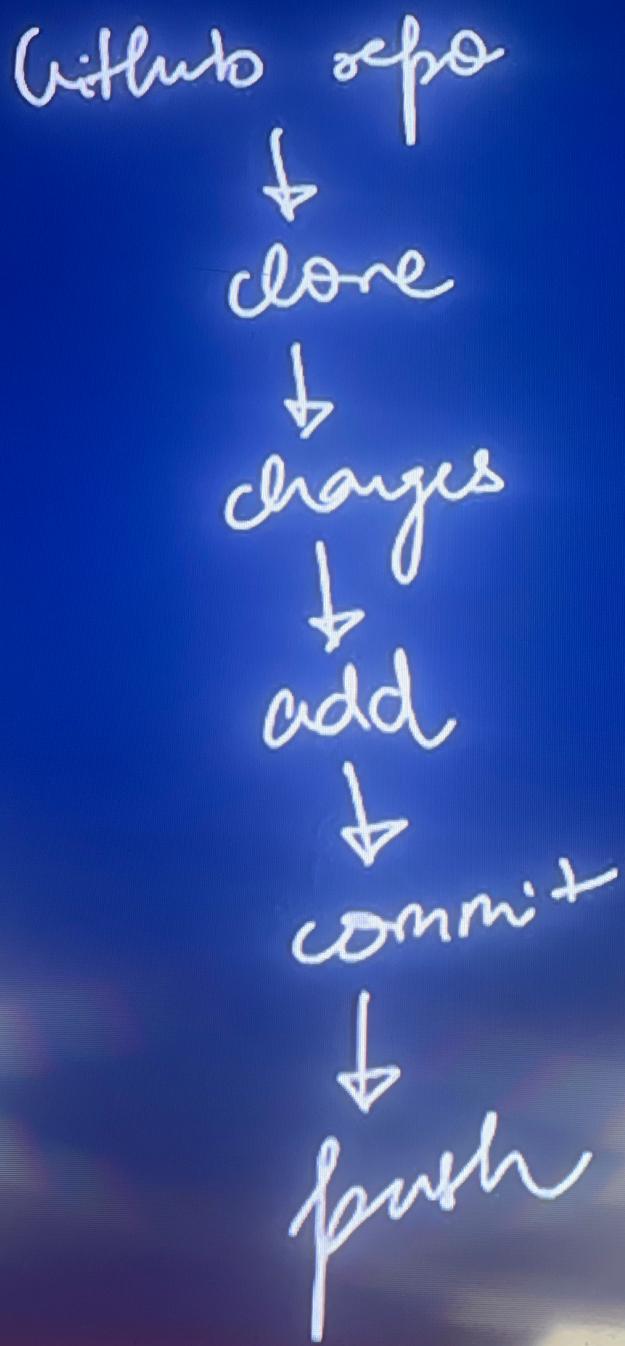
```
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch -M Main
```

git push -u origin Main

```
remote main cannot be resolved to a branch.
● (base) ness@Nesss-Laptop python_CW_firstyear % git push origin Main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 7.56 KiB | 7.56 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'Main' on GitHub by visiting:
remote:     https://github.com/NessWrites/Commercial_Website_CW/pull/new/Main
remote:
To https://github.com/NessWrites/Commercial_Website_CW.git
 * [new branch]      Main -> Main
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch
* Main
○ (base) ness@Nesss-Laptop python_CW_firstyear %
```

main is changed to Main

Work flow



How to change branch

`git checkout <-branch name ->` (to navigate)

To create new branch

`git checkout -b <-new branch name ->` (to create new branch)

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git checkout -b feature
Switched to a new branch 'feature'
```

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch
  Main
* feature
● (base) ness@Nesss-Laptop python_CW_firstyear % git checkout feature
Already on 'feature'
```

How to go back to Main

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git checkout Main
Switched to branch 'Main'
Your branch is up to date with 'origin/Main'.
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch
* Main
  feature
○ (base) ness@Nesss-Laptop python_CW_firstyear % █
```

How to delete branch

[git branch -d <branch_name>](#)

We cannot delete the branch when we are in the branch
So got to feature and delete feature2

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch
  Main
  feature
* feature2
○ (base) ness@Nesss-Laptop python_CW_firstyear % git branch -d feature2
git: 'brand' is not a git command. See 'git --help'.
```

The most similar command is
 branch

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git checkout feature
Switched to branch 'feature'
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch -d feature2
Deleted branch feature2 (was 173a5ff).
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch
  Main
* feature
○ (base) ness@Nesss-Laptop python_CW_firstyear % █
```

How to create branch - Practical approach
First create branch
Edit code on VS code

when pushing push on branch... git push origin <branch_name>

1.add

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git branch
  Main
* feature
● (base) ness@Nesss-Laptop python_CW_firstyear % git add inventories.txt
```

2. commit and push

```
● (base) ness@Nesss-Laptop python_CW_firstyear % git commit -m "How"
dquote>
dquote> clear
dquote>
[feature d12a60c] "How
  1 file changed, 1 insertion(+)
● (base) ness@Nesss-Laptop python_CW_firstyear % git push origin feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 338 bytes | 338.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/NessWrites/Commercial_Website_CW/pull/new/feature
remote:
To https://github.com/NessWrites/Commercial_Website_CW.git
 * [new branch]      feature -> feature
● (base) ness@Nesss-Laptop python_CW_firstyear % git status
On branch feature
nothing to commit, working tree clean
```

Merging

Way 1

git diff <-branch name-> (to compare commits, branchews, files & more)

git merge<-branch name-> (to merge two branches)

Compare branch and main

git diff main

```

● (base) ness@Nesss-Laptop python_CW_firstyear % git diff main
diff --git a/inventories.txt b/inventories.txt
index 6dedb6e..b435763 100644
--- a/inventories.txt
+++ b/inventories.txt
@@ -3,3 +3,4 @@ Microphone Set, Audio Technica, $189, 95
 Disco Light Set, Sonoff, $322, 15
 7.1 Surround Sound Speaker Set, Dolby, $489, 31
 Dinner Table 8x5, Panda Furnitures, $344, 31
+Love is for suckers, Love, $3, 1
○ (base) ness@Nesss-Laptop python_CW_firstyear %

```

Merge

`git merge main`

Way 2

Create a pull request

The screenshot shows a GitHub repository page for 'Commercial_Website_CW'. The repository is public and has 2 branches and 0 tags. A yellow banner at the top indicates that the 'Main' branch had recent pushes less than a minute ago. Below the banner, it says 'This branch is 2 commits ahead, 1 commit behind main.' The repository contains several files: database.txt, inventories.txt, main.py, operations.py, read.py, readme.txt, returns_database.txt, and write1.py. The 'readme.txt' file is expanded, showing its content: 'Author name: Ness', 'Project name: Commercial Website', 'Org : Islington', 'Year: 1st', and 'Project: Python Course Work'. On the right side of the page, there are sections for 'About', 'Releases', and 'Packages', all of which are currently empty.

When changes are being made on different lines, automerge is done by github
But when there is a change in a single line,

When the merging is done and pull request is opened and closed
and we got to make changes in the main of our local machine
we do pull command

Pull Command

```
git pull origin main
```

Resolving merge conflicts

```
git diff main
```

```
● (base) ness@Nesss-Laptop python_CW_fistyear % git diff main
diff --git a/inventories.txt b/inventories.txt
index b435763..fbff1be 100644
--- a/inventories.txt
+++ b/inventories.txt
@@ -3,4 +3,4 @@ Microphone Set, Audio Technica, $189, 95
 Disco Light Set, Sonoff, $322, 15
 7.1 Surround Sound Speaker Set, Dolby, $489, 31
 Dinner Table 8x5, Panda Furnitures, $344, 31
-Love is for suckers, Love, $3, 1
:...skipping...
diff --git a/inventories.txt b/inventories.txt
index b435763..fbff1be 100644
--- a/inventories.txt
+++ b/inventories.txt
@@ -3,4 +3,4 @@ Microphone Set, Audio Technica, $189, 95
 Disco Light Set, Sonoff, $322, 15
 7.1 Surround Sound Speaker Set, Dolby, $489, 31
 Dinner Table 8x5, Panda Furnitures, $344, 31
-Love is for suckers, Love, $3, 1
+Love is for suckers, Love, $3, 00
~
~
~
```

```
git merge main
```

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure with files like `readme.txt`, `inventories.txt` (marked with a red outline), `read.py`, and `database.txt`.
- Editor View:** Displays the content of `inventories.txt` with a merge conflict at line 5. The conflict text is:


```

1 Velvet Table Cloth, Saathi, $8, 113
2 Microphone Set, Audio Technica, $189, -95
3 Disco Light Set, Sonoff, $322, -15
4 7.1 Surround Sound Speaker Set, Dolby, $489, -31
5 Dinner Table 8x5, Panda Furnitures, $344, 31
      
```

 Below this, there are three options for accepting changes: "Accept Current Change", "Accept Incoming Change", "Accept Both Changes", and "Compare Changes".
- Terminal View:** Shows the command-line history of git operations:


```

(base) ness@Ness-Laptop python_CW_firstyear % git add .
(base) ness@Ness-Laptop python_CW_firstyear % git commit -m"Wrong"
[Main 129153e] Wrong
  1 file changed, 1 insertion(+), 1 deletion(-)
(base) ness@Ness-Laptop python_CW_firstyear % git checkout feature
Switched to branch 'feature'
(base) ness@Ness-Laptop python_CW_firstyear % git add .
(base) ness@Ness-Laptop python_CW_firstyear % git commit -m"Wrong"
[feature 6b5045d] Wrong
  1 file changed, 1 insertion(+), 1 deletion(-)
(base) ness@Ness-Laptop python_CW_firstyear % git diff main
diff --git a/inventories.txt b/inventories.txt
index 3c2ab6c..9dfc360 100644
--- a/inventories.txt
+++ b/inventories.txt
@@ -4 +5 @@ Microphone Set, Audio Technica, $189, 95
Disco Light Set, Sonoff, $322, 15
7.1 Surround Sound Speaker Set, Dolby, $489, 31
Dinner Table 8x5, Panda Furnitures, $344, 31
-Love is for suckers, Love, $3, 1111111111
+Love is for suckers, Love, $3, 9998
      
```

 A conflict message follows: "CONFLICT (content): Merge conflict in inventories.txt. Automatic merge failed; fix conflicts and then commit the result."
- Status Bar:** Shows "Ln 5, Col 45" and other terminal settings.

See Accept current change below line 5, its a feature of github.. click one option out of three according to your needs

Or remove all the following lines: 6,8,10 and keep both modified lines manually. then add and commit

check status > git status
 check difference with main. > git diff main
 go to main > git checkout main
 git merge feature1 (to merge with feature 1)
 git push
 refresh

Undoing Changes

Case 1 :Stage changes

git reset <file name>

Case 2: Commit changes (for one commit)

git reset HEAD~1

q to quit

Case 3:

git log shows the records of all the commits

-> **git log**

git log shows the records of all the commits

git reset <hashcode>

to reset hascode from the vscode as well

git reset --hard<hashcode>

Fork

To make new repository which would be the rough copy of old one