GIT COMMANDS TUTORIAL

* GIT stands for Global Information Tracker.
* First we need to install the git bash, visual studio code, windows power shell and then we have to create a git hub account for reaching out all these commands.
* Basically git bash is a command line interface for git that allow users to interact with git repositories using unix style commands on windows.
* In windows power shell the commands will execute line by line it will be very complex for multiple lines.
* So here we mostly uses git bash and visual studio code.

**1.NAVIGATE TO YOUR PROJECT DIRECTORY**

**SYNTAX:** cd/path/to/your/project

Example: cd “/c/users/annep/one drive/desktop/git”

* To get this path open the file explorer and go to the folder which you want and right click on the address bar and copy the path.

Output:

annep@Niharika MINGW64 ~/OneDrive/Desktop/git (master)

**2.FOR SETTING GLOBAL USERNAME AND EMAIL**

**SYNTAX:** git config –global user.name “your name”

Git config –global user. email “[your\_email@example.com](mailto:your_email@example.com)”

Example: git config –global user.name “Annepaka Niharika”

Git config –global user. Email “[annepakaniharika@gmail.com](mailto:annepakaniharika@gmail.com)”

* These commands sets the author name and email address respective to your commits.

**3.TO START A NEW REPOSITORY**

**SYNTAX:** git init [repository name]

* It is already present it will reinitialize it again.

Output:

Reinitialized existing Git repository in C:/Users/annep/OneDrive/Desktop/git/.git/

**4.TO CHECK THE VERSION OF A GIT**

**SYNTAX:** git –version

Output: git version 2.48.1.windows.1

**5.TO CHECK THE GIT CONFIGURATION**

**SYNTAX:** git config –list

* As we set the configurations of username & email.
* It will displays the current git configurations settings on our system.

Output: diff.astextplain.textconv=astextplain

filter.lfs.clean=git-lfs clean -- %f

filter.lfs.smudge=git-lfs smudge -- %f

filter.lfs.process=git-lfs filter-process

filter.lfs.required=true

http.sslbackend=schannel

core.autocrlf=true

core.fscache=true

core.symlinks=false

pull.rebase=false

credential.helper=manager

credential.https://dev.azure.com.usehttppath=true

init.defaultbranch=master

filter.lfs.clean=git-lfs clean -- %f

filter.lfs.smudge=git-lfs smudge -- %f

filter.lfs.process=git-lfs filter-process

filter.lfs.required=true

user.name=Annepaka Niharika

user.email=annepakaniharika@gmail.com

core.repositoryformatversion=0

core.filemode=false

core.bare=false

core.logallrefupdates=true

core.symlinks=false

core.ignorecase=true

remote.origin.url=https://github.com/Niharika2004-Annepaka/Git.git

remote.origin.fetch=+refs/heads/\*:refs/remotes/origin/\*

branch.master.remote=origin

branch.master.merge=refs/heads/master

**6.TO CLONE THE EXISTING REPOSITORY**

**SYNTAX:** git clone<repository\_url>

* We can find this repository\_url under the homepage of git on the rightside , we are having a code which is in green colour ,while clicking on it there will be displayed urls of HTTPS,SSH,GIT HUB CLI.
* Output:
* Cloning into 'Git'...
* remote: Enumerating objects: 12, done.
* remote: Counting objects: 100% (12/12), done.
* remote: Compressing objects: 100% (7/7), done.
* remote: Total 12 (delta 1), reused 12 (delta 1), pack-reused 0 (from 0)
* Receiving objects: 100% (12/12), done.
* Resolving deltas: 100% (1/1), done.

HTTPS

SYNTAX: git clone <https://github.com/username/repository.git>

Example: $ git clone <https://github.com/Niharika2004-Annepaka/Git.git>

Output;

fatal: destination path 'Git' already exists and is not an empty directory.

SSH:

SYNTAX: git clone [git@github.com:username/repository.set](mailto:git@github.com:username/repository.set)

* This will download the repository to our local machine.
* This command is used to obtain a repository from an existing url.

Output:

fatal: destination path 'Git' already exists and is not an empty directory.

7**.TO CHECK THE REPOSITORY STATUS**

**SYNTAX:** git status

Output;

On branch master

Your branch is up to date with 'origin/master'.

Changes to be committed:

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

deleted: example.py

Untracked files:

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

Git/

example.py

puropale\_documents

* It shows the current state of our working directories (modified, staged or untracked files )

8**.TO ADD A FILE IN A GIT**

**SYNTAX:** git add <filename>

Example: $ git add puropale\_documents

* This command adds a file to the staging area.

**9.FOR ADDING ONE OR MORE FILES IN GIT**

**SYNTAX:** git add \*

Output:

warning: adding embedded git repository: Git

hint: You've added another git repository inside your current repository.

hint: Clones of the outer repository will not contain the contents of

hint: the embedded repository and will not know how to obtain it.

hint: If you meant to add a submodule, use:

hint:

hint: git submodule add <url> Git

hint:

hint: If you added this path by mistake, you can remove it from the

hint: index with:

hint:

hint: git rm --cached Git

hint:

hint: See "git help submodule" for more information.

hint: Disable this message with "git config set advice.addEmbeddedRepo false"

* This command adds one or more to the staging.

**10. GIT COMMIT**

**SYNTAX:** git commit -m “Type in the commit message”

Example : $ git commit -m "added puropale"

Output:

[master 174907d] added puropale

1 file changed, 0 insertions(+), 0 deletions(-)

rename example.py => puropale\_documents (100%)

* This command records or snapshots the file permanently in the version history.

**SYNTAX:** git commit -a

* This command commits any files you have added with the git add command and also commits any files you have changed since then.

Output:

On branch master

Your branch is ahead of 'origin/master' by 1 commit.

(use "git push" to publish your local commits)

nothing to commit, working tree clean

**11. GIT DIFFERENCES**

**SYNTAX:** git diff

* This will show the differences which are not yet stagged.

**SYNTAX:** git diff –staged

**Output:** diff --git a/Git b/Git

new file mode 160000

index 0000000..ecd9ca1

--- /dev/null

+++ b/Git

@@ -0,0 +1 @@

+Subproject commit ecd9ca12c52a3aea32449aef0df8c7b6874e0f07

diff --git a/puropale\_documents b/puropale\_documents

new file mode 100644

index 0000000..e69de29

* This command shows the difference between the files in the staging area and the latest version present.

**SYNTAX:** git diff [first branch] [second branch]

**Example:** $ git diff [master][main]

* This command shows the differences between the two branches mentioned.

**12.GIT RESET**

**SYNTAX:** git reset[file]

Example : git reset ecd9ca12c52a3aea32449aef0df8c7b6874e0f07

Output:

Unstaged changes after reset:

M example.py

* This command undoes all the commit after the specified commit and preserves the changes locally and move the branch pointer back to a previous commit.

**SYNTAX:**  git reset –hard [commit]

Output: git reset --hard ecd9ca12c52a3aea32449aef0df8c7b6874e0f07

HEAD is now at ecd9ca1 commit

**13. GIT LOG**

**SYNTAX:** git log

**Output:** commit 174907d0976c71e1aa918dfc03cbf2cfd65b3099 (**HEAD** -> **master**, **origin/master**)

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Tue Mar 4 10:55:25 2025 +0530

added puropale

commit ecd9ca12c52a3aea32449aef0df8c7b6874e0f07

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Mon Mar 3 15:27:50 2025 +0530

commit

commit f919d3509bf9a65942531e785a128d7b09707949

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Mon Mar 3 15:25:30 2025 +0530

commit

commit e8e79d344e0fdeba5f0c02a9b16e86b06859e658

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Mon Mar 3 15:21:46 2025 +0530

commit

commit eef3ee49b2c44aeee42242692eb5eb0b13195015

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Mon Mar 3 11:18:05 2025 +0530

this is my first commit

**SYNTAX:** git log –follow Add.py

Output : commit f919d3509bf9a65942531e785a128d7b09707949

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Mon Mar 3 15:25:30 2025 +0530

commit

commit e8e79d344e0fdeba5f0c02a9b16e86b06859e658

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Mon Mar 3 15:21:46 2025 +0530

commit

commit eef3ee49b2c44aeee42242692eb5eb0b13195015

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Mon Mar 3 11:18:05 2025 +0530

this is my first commit

* This command is used for lists version history for a file including the renaming of files also.

**14. REMOVING A FILE**

**SYNTAX:** git rm[file]

Example: $ git rm Add.py

Output:

rm 'Add.py'

* This command deletes the files from our working directory and stages the deletion.

**15.GIT SHOW**

**SYNTAX:** git show [commit]

warning: refname 'e8e79d344e0fdeba5f0c02a9b16e86b06859e658' is ambiguous.

Git normally never creates a ref that ends with 40 hex characters

because it will be ignored when you just specify 40-hex. These refs

may be created by mistake. For example,

git switch -c $br $(git rev-parse ...)

where "$br" is somehow empty and a 40-hex ref is created. Please

examine these refs and maybe delete them. Turn this message off by

running "git config set advice.objectNameWarning false"

commit e8e79d344e0fdeba5f0c02a9b16e86b06859e658

Author: NiharikaAnnepaka <annepakaniharika@gmail.com>

Date: Mon Mar 3 15:21:46 2025 +0530

commit

diff --git a/Add.py b/Add.py

index 72b97f0..d0c7ded 100644

--- a/Add.py

+++ b/Add.py

@@ -2,4 +2,11 @@

a=200

b=300

sum = a+b

-print("sum of two numbers is:",sum)

\ No newline at end of file

+diff = a-b

+mult = a\*b

+div = a/b

+print("sum of two numbers is:",sum)

+print("diff of two numbers is:",sum)

+print("mult of two numbers is :",mult)

+

+print("div of two numbers is :",div)

\ No newline at end of file

* This command shows the meta data and content changes of the specified amount.

**16. GIT TAG**

**SYNTAX:** git tag [commit ID]

Example:

$ git tag 174907d0976c71e1aa918dfc03cbf2cfd65b3099

fatal: tag '174907d0976c71e1aa918dfc03cbf2cfd65b3099' already exists

* It will tag to the head🡪master.

**17. GIT BRANCH**

**SYNTAX:** git branch

Example:

$ git branch

Output:

\* master

* This command lists all the local branches in the current repository.

**SYNTAX:** git branch [branch name]

Example:

git branch niha

* This will creates a new branch.

**SYNTAX:** git branch -d [branch name]

Example:

git branch -d niha

output

Deleted branch niha (was ecd9ca1)

* This command deletes the feature branch.

**18. GIT CHECKOUT**

**SYNTAX:** git checkout [branch name]

Example:

$ git checkout harika

Output:

D Add.py

Switched to branch 'harika'

annep@Niharika MINGW64 ~/OneDrive/Desktop/git (harika)

* This command creates a new branch & switches to it.

**19. GIT MERGE**

**SYNTAX:** git merge [branch name]

Example:

$ git merge master

Output:

Already up to date.

* This command merges the specified branches history into the current branch.

**20. GIT REMOTE**

**SYNTAX:** git remote add [variable name] [remote server link]

git remote add origin https://github.com/Niharika2004-Annepaka/repo.git

$ git push origin niha

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'niha' on GitHub by visiting:

remote: https://github.com/Niharika2004-Annepaka/Git/pull/new/niha

remote:

To https://github.com/Niharika2004-Annepaka/Git.git

\* [new branch] niha -> niha

**21. push**

**SYNTAX:** git push [variable name] master

$ git push origin niha

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'niha' on GitHub by visiting:

remote: https://github.com/Niharika2004-Annepaka/Git/pull/new/niha

remote:

To https://github.com/Niharika2004-Annepaka/Git.git

\* [new branch] niha -> niha

* This command sends the committed changes of niha branch to our remote repository.

**SYNTAX:**  git push [variable name] [branch]

$ git push origin niha

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)

remote:

remote: Create a pull request for 'niha' on GitHub by visiting:

remote: https://github.com/Niharika2004-Annepaka/Git/pull/new/niha

remote:

To https://github.com/Niharika2004-Annepaka/Git.git

\* [new branch] niha -> niha

* This commands sends the branch commits to your remote repository.

**SYNTAX:** git push [variable name] :[branch name]

Example:

$ git push origin :niha

Output:

To https://github.com/Niharika2004-Annepaka/Git.git

- [deleted] niha

* This command deletes a branch on your remote repository.

**SYNTAX:** git push –all [variable name]

Example: git push –all origin

Output: everything is up to date.

* This command pushes all branches to remote repository.

**22. GIT PULL**

**SYNTAX:** git pull [repository link]

* This command fetches and merges changes on the remote server to your working directory.

**23. GIT STASH**

**SYNTAX:** git stash

$ git stash save

Output:

Saved working directory and index state WIP on master: ecd9ca1 commit

**SYNTAX:** git stash pop

Example:

$ git stash pop

Output:

On branch master

warning: refname '174907d0976c71e1aa918dfc03cbf2cfd65b3099' is ambiguous.

Git normally never creates a ref that ends with 40 hex characters

because it will be ignored when you just specify 40-hex. These refs

may be created by mistake. For example,

git switch -c $br $(git rev-parse ...)

where "$br" is somehow empty and a 40-hex ref is created. Please

examine these refs and maybe delete them. Turn this message off by

running "git config set advice.objectNameWarning false"

Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.

(use "git pull" to update your local branch)

Changes not staged for commit:

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

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

deleted: Add.py

Untracked files:

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

Git/

puropale\_documents

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

Dropped refs/stash@{0} (880bdc24cdfcd157be757aafb25216b527d233aa)

* This command restores the most recently stashed files.

**SYNTAX:**  git stash list

Example:

$ git stash list

Output:

stash@{0}: WIP on master: ecd9ca1 commit

* This command lists all stashed change sets.

**SYNTAX:** git stash drop

Example:

$ git stash drop

Output:

No stash entries found.

* This command discards the most recently stashed change set.