

GIT AND GITHUB OPERATIONS

**DONE BY:
MOULISH GA**

1) Creating github local repo

```
moulishga@MOULISHGA047A MINGW64 ~  
$ pwd  
/c/Users/moulishga
```

```
moulishga@MOULISHGA047A MINGW64 ~  
$ mkdir cognizance
```

```
moulishga@MOULISHGA047A MINGW64 ~  
$ cd cognizance
```

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance  
$ git init  
Initialized empty Git repository in C:/Users/moulishga/cognizance/.git/
```

2) Git add

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (master)  
$ touch git_sample
```

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (master)  
$ git add git_sample
```

3) Git commit

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (master)
$ git commit -a
[master (root-commit) 6eaaf6d] this is a text file of sample
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 git_sample

moulishga@MOULISHGA047A MINGW64 ~/cognizance (master)
$ git log
commit 6eaaf6d45cf40347472b144140aada0dc26e9f22 (HEAD -> master)
Author: moulishga <mghmmoulish@gmail.com>
Date: Tue Feb 15 11:42:37 2022 +0530

    this is a text file of sample
```

4) Git remote

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (master)  
$ git remote add origin https://github.com/MoulishGA/cognizance.git
```


5)Git branch

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (master)  
$ git branch -M main  
  
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)  
$ git push -u origin main
```


6) Git push

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 220 bytes | 110.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MoulishGA/cognizance.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Remote repository



Pull requests

Issues

Marketplace

Explore

MoulishGA / **cognizance**

Public

Pin

Unwatch 1

Fork 0

Star 0

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main

1 branch

0 tags

Go to file

Add file

Code

MoulishGA this is a text file of sample

6eaa6d 44 minutes ago 1 commit

git_sample

this is a text file of sample

44 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

About

No description, website, or topics provided.

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

7) Git clone

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)
$ git clone https://github.com/MoulishGA/cognizance.git
Cloning into 'cognizance'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```


8) Git checkout

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)  
$ git checkout  
Your branch is up to date with 'origin/main'.
```

9) Git checkout -b


```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)
$ git checkout -b sample-1
Switched to a new branch 'sample-1'

moulishga@MOULISHGA047A MINGW64 ~/cognizance (sample-1)
$ git push origin sample-1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'sample-1' on GitHub by visiting:
remote:      https://github.com/MoulishGA/cognizance/pull/new/sample-1
remote:
To https://github.com/MoulishGA/cognizance.git
 * [new branch]      sample-1 -> sample-1
```

Your branches

`sample-1` Updated 1 hour ago by MoulishGA

0 | 0


 New pull request



Active branches

`sample-1` Updated 1 hour ago by MoulishGA

0 | 0

 New pull request



10) Git fetch

Since there is no change in remote repository

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)  
$ git fetch
```

11) Git merge

All the things are up to date

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)
$ git merge
Already up to date.
```


12) Git pull

Everything is up to date

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)  
$ git pull  
Already up to date.
```

13) Git reset

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)
$ git log --oneline
6eaaf6d (HEAD -> main, origin/sample-1, origin/main, sample-1) this is a text file of sample
```

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)
$ git reset 6eaaf6d --hard
HEAD is now at 6eaaf6d this is a text file of sample
```

14)Git rebase

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)  
$ git rebase main  
Current branch main is up to date.
```

15) Git status

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

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


16) git config --list --show -origin

```
moulishga@MOULISHGA047A MINGW64 ~/cognizance (main)
$ git config --list --show -origin
error: ambiguous option: show (could be --show-origin or --show-scope)
usage: git config [<options>]

Config file location
  --global          use global config file
  --system          use system config file
  --local           use repository config file
  --worktree        use per-worktree config file
  -f, --file <file> use given config file
  --blob <blob-id>  read config from given blob object

Action
  --get             get value: name [value-pattern]
  --get-all        get all values: key [value-pattern]
  --get-regexp      get values for regexp: name-regex [value-pattern]
  --get-urlmatch    get value specific for the URL: section[.var] URL
  --replace-all    replace all matching variables: name value [value-pattern]
  --add            add a new variable: name value
  --unset          remove a variable: name [value-pattern]
  --unset-all      remove all matches: name [value-pattern]
  --rename-section  rename section: old-name new-name
  --remove-section remove a section: name
  -l, --list        list all
  --fixed-value     use string equality when comparing values to 'value-pattern'
  -e, --edit        open an editor
  --get-color       find the color configured: slot [default]
  --get-colorbool   find the color setting: slot [stdout-is-tty]

Type
  -t, --type <>    value is given this type
  --bool           value is "true" or "false"
  --int            value is decimal number
  --bool-or-int    value is --bool or --int
  --bool-or-str    value is --bool or string
  --path           value is a path (file or directory name)
  --expiry-date    value is an expiry date

Other
  -z, --null        terminate values with NUL byte
  --name-only       show variable names only
  --includes        respect include directives on lookup
  --show-origin     show origin of config (file, standard input, blob, command line)
  --show-scope      show scope of config (worktree, local, global, system, command)
  --default <value> with --get, use default value when missing entry
```