



# Task 1

---

COGNIZANCE CLUB

KAIF AHMED R

# Get Init

Initiating an empty Git repository in the location mentioned below

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance  
$ git init  
Initialized empty Git repository in C:/Users/KAIF AHMED R/Desktop/Cognizance/.git/
```

---

# Creating folder and file

Creating folder Test1 using mkdir command and reame.txt file using touch command

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ mkdir Test1

KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ touch readme.txt
```

---

# STAGING STAGE

We can use either `git add .` or `git add -A` command

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git add .
```

Verifying by using status command

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   readme.txt
```

# COMMITTING CHANGES

Captures a snapshot of the project's currently staged changes

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git commit -m "1st commit"
[master (root-commit) d979c3d] 1st commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 readme.txt
```

---

# ORIGIN AND PUSH

Adding the origin

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)  
$ git remote add origin https://github.com/l-kaif-l/Cognizance.git
```

Pushing it into master

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)  
$ git push origin master  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 213 bytes | 213.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/l-kaif-l/Cognizance.git  
* [new branch]      master -> master
```

# Branching

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git checkout -b branch1
Switched to a new branch 'branch1'

KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (branch1)
$ git branch
* branch1
  master

KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (branch1)
$ touch test.txt

KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (branch1)
$ git add .

KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (branch1)
$ git commit -m "test.txt"
[branch1 8948ale] test.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.txt
```

Creating a branch branch1 and creating a new file test.txt in it and committing it.

And then pushed to branch1

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (branch1)
$ git push origin branch1
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 244 bytes | 244.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:      https://github.com/l-kaif-l/Cognizance/pull/new/branch1
remote:
To https://github.com/l-kaif-l/Cognizance.git
 * [new branch]      branch1 -> branch1
```



# Merging

Coming back to master and merging it

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (branch1)
$ git checkout master
Switched to branch 'master'

KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git merge branch1
Updating d979c3d..8948a1e
Fast-forward
 test.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test.txt
```

Pushing it back now

---

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/l-kaif-l/Cognizance.git
 d979c3d..8948a1e  master -> master
```

---



# Cloning

Cloning the same Cognizance file

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git clone https://github.com/1-kaif-1/Cognizance.git
Cloning into 'Cognizance'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ ls
Cognizance/  Test1/  readme.txt  test.txt
```

---

# Fetching, Pulling, Rebase

## Fetching

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git fetch --all
Fetching origin
```

## Pull

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git pull origin master
From https://github.com/l-kaif-l/Cognizance
 * branch      master      -> FETCH_HEAD
Already up to date.
```

---

## Rebase

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git rebase branch1
Current branch master is up to date.
```

# git config --list --show -

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git config --list --show-
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
```

# git config --list --show -origin

```
KAIF AHMED R@LAPTOP-HSNSHP79 MINGW64 ~/Desktop/Cognizance (master)
$ git config --list --show-origin
file:C:/Program Files/Git/etc/gitconfig diff.astextplain.textconv=astextplain
file:C:/Program Files/Git/etc/gitconfig filter.lfs.clean=git-lfs clean -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Program Files/Git/etc/gitconfig filter.lfs.process=git-lfs filter-process
file:C:/Program Files/Git/etc/gitconfig filter.lfs.required=true
file:C:/Program Files/Git/etc/gitconfig http.sslbackend=openssl
file:C:/Program Files/Git/etc/gitconfig http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
file:C:/Program Files/Git/etc/gitconfig core.autocrlf=true
file:C:/Program Files/Git/etc/gitconfig core.fscache=true
file:C:/Program Files/Git/etc/gitconfig core.symlinks=false
file:C:/Program Files/Git/etc/gitconfig pull.rebase=false
file:C:/Program Files/Git/etc/gitconfig credential.helper=manager-core
file:C:/Program Files/Git/etc/gitconfig credential.https://dev.azure.com.usehttppath=true
file:C:/Program Files/Git/etc/gitconfig init.defaultbranch=master
file:C:/Users/KAIF AHMED R/.gitconfig filter.lfs.smudge=git-lfs smudge -- %f
file:C:/Users/KAIF AHMED R/.gitconfig filter.lfs.process=git-lfs filter-process
file:C:/Users/KAIF AHMED R/.gitconfig filter.lfs.required=true
file:C:/Users/KAIF AHMED R/.gitconfig filter.lfs.clean=git-lfs clean -- %f
file:C:/Users/KAIF AHMED R/.gitconfig user.name=l-kaif-l
file:C:/Users/KAIF AHMED R/.gitconfig user.email=kaifahmed1052004@gmail.com
file:C:/Users/KAIF AHMED R/.gitconfig core.editor="D:\Microsoft VS Code\bin\code" --wait
file:./git/config core.repositoryformatversion=0
file:./git/config core.filemode=false
file:./git/config core.bare=false
file:./git/config core.logallrefupdates=true
file:./git/config core.symlinks=false
file:./git/config core.ignorecase=true
file:./git/config remote.origin.url=https://github.com/l-kaif-l/Cognizance.git
file:./git/config remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
```

# Forking

I forked my friend Abhay's repository.

