

COGNIZANCE

TASK-1

BASIC OPERATIONS

1. Creating repository:
2. **git add**
Puts current working files into the stage.
3. **git commit**
Clone an existing repository into a new directory.

```
HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads
$ mkdir Cognizance

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads
$ cd Cognizance

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance
$ echo "# Cognizance" >> README.md

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance
$ git init
Initialized empty Git repository in C:/Users/HP/Downloads/Cognizance/.git/

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance (master)
$ git commit -m "first commit"
[master (root-commit) 4834f46] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

4. **git push**
Uploads changes from all local branches to the respective remote repositories.

```
HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance (master)
$ git branch -M main

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance (main)
$ git remote add origin https://github.com/kirthi0/Cognizance.git

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kirthi0/Cognizance.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

```
HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance (main)
$ mkdir TASK-1
```

```
HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance (main)
$ ls
README.md  TASK-1/

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance (main)
$ cd TASK-1

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ git add COGNIZANCE_TASK1.docx

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ 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:   COGNIZANCE_TASK1.docx
```

```
HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ git commit -m "second commit"
[main 65332c3] second commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 TASK-1/COGNIZANCE_TASK1.docx

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 10.13 KiB | 10.13 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kirthi0/Cognizance.git
 4834f46..65332c3  main -> main
branch 'main' set up to track 'origin/main'.
```

5. git checkout

Replaces the current working files with files from a branch.

```

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ vim file1.txt

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ cat file1.txt
This is a part of cognizance task1

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ git add file1.txt
warning: LF will be replaced by CRLF in TASK-1/file1.txt.
The file will have its original line endings in your working directory

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ git commit -m "third commit"
[main b7daebf] third commit
1 file changed, 1 insertion(+)
create mode 100644 TASK-1/file1.txt

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ vim file1.txt

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ cat file1.txt
This is a part of cognizance task1
Working with checout command

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ git checkout -- file1.txt

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ cat file1.txt
This is a part of cognizance task1

```

6. **git checkout -b**

Creates a new local branch from the current branch's tip.

```

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (main)
$ git checkout -b branch3
Switched to a new branch 'branch3'

```

7. **git clone**

Commits staged changes to a local branch.

```

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads
$ cd demo

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/demo
$ git clone https://github.com/kirthi0/Cognizance.git
Cloning into 'Cognizance'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), 10.32 KiB | 880.00 KiB/s, done.

```

```

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ git commit -a
[main 13151ab] "3rd commit"
1 file changed, 1 insertion(+), 1 deletion(-)

HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ cat file1.txt
This is a part of cognizance task1
Trying out a few basic git bash commands

```

8. git fetch

Downloads changes from a remote repository into the local clone.

```
HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (main)
$ git fetch https://github.com/MunGell/awesome-for-beginners.git
remote: Enumerating objects: 1206, done.
remote: Counting objects: 100% (60/60), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 1206 (delta 18), reused 31 (delta 10), pack-reused 1146
Receiving objects: 100% (1206/1206), 461.24 KiB | 1.27 MiB/s, done.
Resolving deltas: 100% (470/470), done.
From https://github.com/MunGell/awesome-for-beginners
 * branch                HEAD      -> FETCH_HEAD
```

```
HP@LAPTOP-K8VRM016 MINGW64 ~/Downloads/Cognizance/TASK-1 (branch1)
$ git merge branch2
Already up to date.
```

9. git rebase

Integrates changes from one branch into another. It is an alternative to the better known “merge” command.

```
HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2|REBASE 3/5)
$ git add file1.txt

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2|REBASE 3/5)
$ git rebase --continue
[detached HEAD c4146bb] "fourth commit"
 1 file changed, 4 insertions(+)
Successfully rebased and updated refs/heads/branch2.

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2)
$ git rebase main
Current branch branch2 is up to date.
```

10. git pull

Fetches remote changes on the current branch into the local clone, and merges them into the current working files.

```
HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2)
$ git pull origin main
From https://github.com/kirthi0/Cognizance
 * branch                main      -> FETCH_HEAD
Already up to date.
```



```

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2)
$ git commit -m "edited the doc"
On branch branch2
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:   COGNIZANCE_TASK1.docx

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

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2)
$ git commit -a
[branch2 199bb52] "edited the doc on branch2"
 1 file changed, 0 insertions(+), 0 deletions(-)

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (main)
$ git push origin main
Everything up-to-date

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (main)
$ git pull origin main
From https://github.com/kirthi0/Cognizance
 * branch          main          -> FETCH_HEAD
Already up to date.

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (main)
$

```

11. **git commit -a**

Commits all modified files to a local branch(shorthand for “git add” followed by “git commit” for each modified file).

12. **git merge**

Merges files from a given branch into the current branch.

```

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2)
$ git commit -a
[branch2 379911c] "minor changes"
 1 file changed, 0 insertions(+), 0 deletions(-)

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (branch2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (main)
$ git merge branch2
Updating 2bcd572..379911c
Fast-forward
 TASK-1/COGNIZANCE_TASK1.docx | Bin 12749 -> 420845 bytes
 TASK-1/file1.txt              | 12 ++++++++
 TASK-1/file2.txt              |  2 ++
 3 files changed, 14 insertions(+)
 create mode 100644 TASK-1/file2.txt

```

13. **git reset**

Makes the current branch point to some specific revision or branch

```

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (main)
$ git reset 06f82ba
Unstaged changes after reset:
M      TASK-1/COGNIZANCE_TASK1.docx
M      TASK-1/file2.txt

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 4 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

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:   COGNIZANCE_TASK1.docx
        modified:   file2.txt

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

HP@LAPTOP-K8VRM016 MINGW64 ~/downloads/cognizance/task-1 (main)
$ git log --oneline
06f82ba (HEAD -> main) modified
c4146bb "fourth commit"
abbb8a3 "3rd commit"
364cde6 "4th commit"
2bcd572 new commit made on main branch
65332c3 second commit
4834f46 first commit

```

```

HP@LAPTOP-K8VRM016 MINGW64 ~
$ 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.usehttp
path=true
file:C:/Program Files/Git/etc/gitconfig init.defaultbranch=master
file:C:/Users/HP/.gitconfig core.editor="C:\Users\HP\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
file:C:/Users/HP/.gitconfig user.name=kirthi0
file:C:/Users/HP/.gitconfig user.email=mvsridhar1998@gmail.com

```

git fork

Fork will create a copy of the repository in your GitHub account so that you can make changes to the project.

knaxus / problem-solving-javascript

Public

Watch21

Fork211

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security


Insights

Create a new fork


A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Owner *

Repository name *


 kirithi0

/

problem-solving-javascript 

By default, forks are named the same as their parent repository. You can customize the name to distinguish it further.

Description (optional)

 You are creating a fork in your personal account.

Create fork