

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (master)
$ git init
Initialized empty Git repository in C:/Users/P Havirbhavi/OneDrive/Desktop/Cognizance/.git/
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (master)
$ git add .
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (master)
$ git commit -m "comment"
[master (root-commit) 410b2dc] comment
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 git_commands (1).pdf
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (master)
$ git remote add origin https://github.com/Havirbhavi/Cognizance
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311.51 KiB | 6.23 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/Havirbhavi/Cognizance/pull/new/master
remote:
To https://github.com/Havirbhavi/Cognizance
 * [new branch]      master -> master
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (master)
$ git add git_commands.pdf
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (master)
$ git checkout -b git_commands.pdf
Switched to a new branch 'git_commands.pdf'
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git_commands.pdf)
$ git commit
Aborting commit due to empty commit message.
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git_commands.pdf)
$ git commit -m "Task-1"
[git_commands.pdf 4009741] Task-1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 git_commands.pdf
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git_commands.pdf)
$ git commit -a
Aborting commit due to empty commit message.
```

```
P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git_commands.pdf)
$ git fetch
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
```

```
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 596 bytes | 22.00 KiB/s, done.
From https://github.com/Havirbhavi/Cognizance
* [new branch]      main      -> origin/main
```

P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git\_commands.pdf)

```
$ git pull
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.
```

```
git pull <remote> <branch>
```

If you wish to set tracking information for this branch you can do so with:

```
git branch --set-upstream-to=origin/<branch> git_commands.pdf
```

P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git\_commands.pdf)

```
$ git push origin master
Everything up-to-date
```

P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git\_commands.pdf)

```
$ git reset
Unstaged changes after reset:
D      git commands (1).pdf
```

P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git\_commands.pdf)

```
$ git fork
git: 'fork' is not a git command. See 'git --help'.
```

The most similar commands are

```
fsck
hook
```

P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git\_commands.pdf)

```
$ git fsck
Checking object directories: 100% (256/256), done.
dangling tree d3d20d81f93f1f5bbbc569eeaf21131c12769761
```

P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git\_commands.pdf)

```
$ git clone -b https://github.com/Havirbhavi/Cognizance.git
fatal: You must specify a repository to clone.
```

usage: git clone [<options>] [--] <repo> [<dir>]

-v, --verbose	be more verbose
-q, --quiet	be more quiet
--progress	force progress reporting
--reject-shallow	don't clone shallow repository
-n, --no-checkout	don't create a checkout
--bare	create a bare repository
--mirror	create a mirror repository (implies bare)
-l, --local	to clone from a local repository
--no-hardlinks	don't use local hardlinks, always copy
-s, --shared	setup as shared repository
--recurse-submodules [= <pathspec>]	

fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

```
-v, --verbose      be more verbose
-q, --quiet        be more quiet
--progress         force progress reporting
--reject-shallow   don't clone shallow repository
-n, --no-checkout  don't create a checkout
--bare             create a bare repository
--mirror           create a mirror repository (implies bare)
-l, --local        to clone from a local repository
--no-hardlinks     don't use local hardlinks, always copy
-s, --shared       setup as shared repository
--recurse-submodules[=<pathspec>]
                    initialize submodules in the clone
--recursive ...    alias of --recurse-submodules
-j, --jobs <n>     number of submodules cloned in parallel
--template <template-directory>
                    directory from which templates will be used
--reference <repo> reference repository
--reference-if-able <repo>
                    reference repository
--dissociate        use --reference only while cloning
-o, --origin <name> use <name> instead of 'origin' to track upstream
-b, --branch <branch>
                    checkout <branch> instead of the remote's HEAD
-u, --upload-pack <path>
                    path to git-upload-pack on the remote
--depth <depth>    create a shallow clone of that depth
--shallow-since <time>
                    create a shallow clone since a specific time
--shallow-exclude <revision>
                    deepen history of shallow clone, excluding rev
--single-branch     clone only one branch, HEAD or --branch
--no-tags           don't clone any tags, and make later fetches not to follow them
--shallow-submodules
                    any cloned submodules will be shallow
--separate-git-dir <gitdir>
                    separate git dir from working tree
-c, --config <key=value>
                    set config inside the new repository
--server-option <server-specific>
                    option to transmit
-4, --ipv4          use IPv4 addresses only
-6, --ipv6          use IPv6 addresses only
--filter <args>     object filtering
--also-filter-submodules
                    apply partial clone filters to submodules
--remote-submodules
                    any cloned submodules will use their remote-tracking branch
--sparse            initialize sparse-checkout file to include only files at root
```

P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git\_commands.pdf)

```
$ git --version
git version 2.36.1.windows.1
```

P Havirbhavi@DESKTOP-5KR39F6 MINGW64 ~/OneDrive/Desktop/Cognizance (git\_commands.pdf)

```
$
```