

## GIT Task

**Task:** Establish a new directory, populate it with script files, initiate an empty repository on GitHub, convert the local directory into a Git repository, and link it to GitHub for pushing the code into the repository. Perform merge, rebase, stash commands in following github repo.

### 1. Establish a new directory, populate it with script files.

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~ (master)
$ cd "C:\Users\krant\Documents\git"

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git (ravali)
$ mkdir my-scripts

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git (ravali)
$ cd my-scripts

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (ravali)
$ touch script1.sh script2.sh

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (ravali)
$ |
```

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (ravali)
$ cat script1.sh script2.sh
#!/bin/bash

echo "" hello world"
# !/bin/bash
echo "Hi, i am ravali"
```

### 2. initiate an empty repository on GitHub.

The screenshot shows the GitHub repository creation interface. It consists of two main sections: 'General' and 'Configuration'.

**General Section:**

- Owner:** Dugyalaravali
- Repository name:** my-scripts (highlighted with a green border)
- A note says: "my-scripts is available."
- Description:** A text input field with placeholder "Great repository names are short and memorable. How about [solid-lamp](#)?" and character count "0 / 350 characters".

**Configuration Section:**

- Choose visibility:** Public (selected)
- Add README:** Off (switch is off)
- Add .gitignore:** No .gitignore (dropdown menu)
- Add license:** No license (dropdown menu)

**Bottom Right:** A large green button labeled "Create repository".

### 3. convert the local directory into a Git repository.

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (ravali)
$ git init
Initialized empty Git repository in C:/Users/krant/Documents/git/my-scripts/.git/

```

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (master)
```

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (master)
$ git add .
warning: in the working copy of 'script1.sh', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'script2.sh', LF will be replaced by CRLF the next time Git touches it
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (master)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   script1.sh
    new file:   script2.sh
```

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (master)
$ git commit -m "scripts updated"
[master (root-commit) 4f607b0] scripts updated
 2 files changed, 7 insertions(+)
 create mode 100644 script1.sh
 create mode 100644 script2.sh
```

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (master)
$ git remote add origin https://github.com/Dugyalaravali/my-scripts.git
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (master)
$ git remote -v
origin  https://github.com/Dugyalaravali/my-scripts.git (fetch)
origin  https://github.com/Dugyalaravali/my-scripts.git (push)

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (master)
$ |
```

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (master)
$ git branch -M main
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 328 bytes | 328.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Dugyalaravali/my-scripts.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (main)
```

my-scripts Public

main 1 Branch 0 Tags

Ravali changes updated 751350e · 38 minutes ago 2 Commits

script1.sh scripts updated 52 minutes ago

script2.sh changes updated 38 minutes ago

README

## 5. Perform merge.

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (main)
$ git branch
* main

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (main)
$ git checkout -b dev
Switched to a new branch 'dev'

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ echo "i am testing" >> script2.sh

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git add .
warning: in the working copy of 'script2.sh', LF will be replaced by CRLF the next time Git touches it

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git commit -m "changes updated"
[dev 751350e] changes updated
 1 file changed, 1 insertion(+)
```

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (main)
$ git merge dev
Updating 4f607b0..751350e
Fast-forward
  script2.sh | 1 +
  1 file changed, 1 insertion(+)
```

## 6. Perform rebase.

```
Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (feature)
$ echo "echo Feature work" >> script1.sh
git add .
git commit -m "Feature commit"
warning: in the working copy of 'script1.sh', LF will be replaced by CRLF the next time Git touches it
[feature 4eea2de] Feature commit
 1 file changed, 1 insertion(+)

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (feature)
$ git checkout feature
Already on 'feature'

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (feature)
$ git rebase main
Current branch feature is up to date.

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (feature)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Ravali@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (main)
$ git rebase main
Current branch main is up to date.
```

```

$ git log
commit 4eea2de011092628159d70b481d5d26b62c9bcc (HEAD -> feature)
Author: Raveli <ravalidugyala@gmail.com>
Date:   Thu Jan 22 15:29:26 2026 +0530

    Feature commit

commit 751350e138d14c778ae4b737b521c59e24226a63 (origin/main, main, dev)
Author: Raveli <ravalidugyala@gmail.com>
Date:   Thu Jan 22 15:19:05 2026 +0530

    changes updated

commit 4f607b0a0168bab8527554f0b30ee174c69a93fe
Author: Raveli <ravalidugyala@gmail.com>
Date:   Thu Jan 22 15:05:24 2026 +0530

    changes updated

```

## 7. perform stash.

```

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git stash list

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ touch script3.sh

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ echo "i am testing git" >> script3.sh

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git stauts
git: 'stauts' is not a git command. See 'git --help'.

The most similar command is
status

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git status
On branch dev
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    script3.sh

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

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git stash
No local changes to save

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git add .
warning: in the working copy of 'script3.sh', LF will be replaced by CRLF the next time Git touches it

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git stash
Saved working directory and index state WIP on dev: cc63cda Save current dev changes

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git stash list
stash@{0}: WIP on dev: cc63cda Save current dev changes

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git stash pop
On branch dev
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   script3.sh

Dropped refs/stash@{0} (c1a4c9098444df197bf3fd3c9fc0d898361b35b4)

Raveli@DESKTOP-IAS39G4 MINGW64 ~/Documents/git/my-scripts (dev)
$ git stash list

```