Git config

```
C:\Users\Krishna>git config --global user.name "krkopash"

C:\Users\Krishna>git config --global user.email "krk.opash@gmail.com"

C:\Users\Krishna>git config --global --list
user.name=krkopash
user.email=krk.opash@gmail.com
```

```
C:\Users\Krishna>git config user.name
krkopash
```

Git commit

```
C:\Users\Krishna\Desktop\day 5>git add --all
warning: adding embedded git repository: Day-5-Git-GitHub-Workflow
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> Day-5-Git-GitHub-Workflow
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached Day-5-Git-GitHub-Workflow
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config set advice.addEmbeddedRepo false"

C:\Users\Krishna\Desktop\day 5>git commit -m "add"
[main (root-commit) 191c3ba] add
 1 file changed, 1 insertion(+)
 create mode 160000 Day-5-Git-GitHub-Workflow
```

Git push

```
C:\Users\Krishna\Desktop\day 5>git log
commit 191c3ba2f4d607f341bf76b5bd34380910ad7c4a (HEAD -> main)
Author: krkopash <krk.opash@gmail.com>
Date:   Wed Jan 7 15:05:54 2026 +0530

    add

C:\Users\Krishna\Desktop\day 5>git remote add origin https://github.com/krkopash/Day-5-Git-GitHub-Workflow.git
error: remote origin already exists.

C:\Users\Krishna\Desktop\day 5>git push -u origin main
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 206 bytes | 206.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/krkopash/Day-5-Git-GitHub-Workflow.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\Krishna\Desktop\day 5>
```

Git branch

```
C:\Users\Krishna\Desktop>cd day 5

C:\Users\Krishna\Desktop\day 5>git branch hello

C:\Users\Krishna\Desktop\day 5>git branch
  hello
* main

C:\Users\Krishna\Desktop\day 5>git checkout hello
Switched to branch 'hello'

C:\Users\Krishna\Desktop\day 5>git status
On branch hello
nothing to commit, working tree clean
```

Basic git commands:

start a working area (see also: git help tutorial)
- clone Clone a repository into a new directory
- init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
- add Add file contents to the index
- mv Move or rename a file, a directory, or a symlink
- restore Restore working tree files
- rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect   Use binary search to find the commit that introduced a bug
   diff   Show changes between commits, commit and working tree, etc
   grep   Print lines matching a pattern
   log   Show commit logs
   show   Show various types of objects
   status   Show the working tree status

grow, mark and tweak your common history
   backfill Download missing objects in a partial clone
   branch   List, create, or delete branches
   commit   Record changes to the repository
   merge   Join two or more development histories together
   rebase   Reapply commits on top of another base tip
   reset   Reset current HEAD to the specified state
   switch   Switch branches
   tag   Create, list, delete or verify tags

collaborate (see also: git help workflows)
   fetch   Download objects and refs from another repository
   pull   Fetch from and integrate with another repository or a local branch
   push   Update remote refs along with associated objects

GitHub Flow

- Create a Branch: Start new work without affecting the main code.
- Make Commits: Save progress as you make changes.
- Open a Pull Request: Ask others to review your work.
- Review: Discuss and improve the changes together.
- Deploy: Test your changes before merging.
- Merge: Add your finished work to the main branch.

Git Workflow Commands Overview

- Working Directory - Where you make changes
- git add - Stage changes
- git commit - Save changes to your repository
- git push - Share changes with others
- git status - Check what's going on
- Undo/Amend - Fix mistakes (`git restore`, `git reset`, `git commit --amend`)