Basic Git Commands

The git commands are executed in a command window. VS, VS Code, Command Prompt and PowerShell command window in Github Desktop all have the ability to open command windows.

To execute these commands, you must have Git installed. If you have Github Desktop installed then you already have Git installed. If not, google download Git and install.

Typical syntax: git command [option(s)]

Example: to track and prepare your repository changes for committing you could use:

git add . (all changes in files)

git add -a (all changes in files)

git add <filename> (just that single file)

For help with the syntax of the command, you can use:

git command -help (display summary to window)

git command --help (open browser window)

|  |  |
| --- | --- |
| Command | Action |
| git status | This shows in a list the current status of tracked and untracked files in your local repository |
| git add . | Stages all untracked changes to the repository for committing. |
| git commit -m “your subject message” | Commits staged files |
| git push | Pushes your local commits to github.com (origin/xxx) |
| git pull  (git fetch and get merge) | Fetch and merge from github.com |
| git stash  git stash list  git stash apply | You and a partner have been working on a common file in a meeting. Both have changes to the file. Who commits?  Stash takes the dirty state of your current repo directory and saves the changes on the “stack” of unfinished changes.  Now, one person can do their stage, commit and push. Then the other person can do a pull. Then retrieve the “stashed” changes. Do their stage, commit and push. |
| git log  options:  -n x (where is a number)  --author=”…” (changes by)  --after=”2021-11-22”  --before=”yesterday”  --after=”…” --before=”…” | Shows the history of your commits in the repo |