

Bash - Bourne Again SHell
Cli - command-line interface
Bash uses repl - read eval print loop
whoami - user
pwd - print working dir
ls (-a all, -f forces all files to appear as is)
mkdir - make dir, rm - remove,
mv file path - move, -i to ask for confirmation before
overwriting, can also be used to rename files
-r recursive
cp file path - copy
When naming: no spaces, no dash begin, alphaNum
head/tail file - first/last x 3 lines
ls *.md lists all .md files, touch file_path - create file
grep string file, history - history of cli commands
man (bash command) to open manual, q to quit, space
to go down page, / starts search

ssh - secure shell, server is destination, client is you
Private key decrypts msgs from public key
Git repo stores info in .git dir
git add -p to stage changes interactively
git init to convert current dir to local repo
staging area allows some files to be committed only.
To add all updated, new, and deleted files in the git repo
you can type `git add --all`.
To only add all updated and new files in the git repo you
can type `git add .`.
To only add all updated and deleted files in the git repo
you can type `git add -u`,
To see which files you have staged, before committing
them, you can use `git status`.
To see the exact changes you have staged instead of
just the file names, use `git diff --staged`.
if you change your mind and want to remove a file from
the staging area, you can type
`git restore --staged name-of-your-file.txt`

git log --oneline -patch (displays commit log with one line
per commit and patches of changes visible)
git restore --source SHORT_SHA-1 filename
If changes on diff lines, git will auto merge
If on same line, merge conflict raised
git pull --rebase=false
git stash store local changes until brought back with
git stash pop, good for dropping n picking smtg else
code .gitignore, list files/dirs one per line

git reset short_sha to go back to prev. Commit
Add --hard to discard changes rather than in working
directory

Times when a force push is legitimate includes
when you have have rebased a feature branch to sync it
with master or if you are working alone and you are
100% sure that you want to remove a commit from
GitHub.

quarto --help, `quarto create [type]` creates boilerplate
template
quarto create project -> website
quarto preview - previews html, refreshes any changes
quarto render - generates html/pdf, with current changes
Github site: USERNAME.github.io/REPO_NAME
Or USERNAME or USERNAME.github.io

Change quarto.yml to output-dir: docs for website