GIT Cheat Sheet

- Configure Git:
  git config
  Will showcase all options that can be passed as parameters with this command
- Configuration level flag:
  --local: Applies to the current git repository
  git config --local <option>
  --global: Applies to all git repositories for the current user
  git config --global <option>
  --system: Applies to all git repositories for all users in the system
  git config --system <option>
- Configuration level file:
  --local:
  .git/config
  --global:
  ~/.gitconfig
  --system:
  /etc/gitconfig
- View configuration:
  git config <option> --list
- Configure Aliases:
  git config --global alias.co checkout
  git config --global alias.br branch
  git config --global alias.ci commit
  git config --global alias.st status
- Initialize a new local repository:
  git init
- View remote repository:
  git remote (list without remote URL)
  git remote -v (list with remote URL)
- Add remote in local repository:
  git remote add <name> <URL>
  Example:
  git remote add origin https://github.com/username/repo

- Clone a repository into a new directory:
  git clone <URL>
  Or
  git clone <URL> <dir_name>
- Clone a specific branch of a repository:
  git clone -b <branch_name> <URL>
- Add file contents to the index found in the working tree:
  git add <file_name>
  Or
  git add .
- Record changes to the repository:
  git commit -m <commit_message>
- Modify the last commit:
  git commit --amend -m <commit_message>
- Update remote refs along with associated objects:
  git push <remote> <branch>
  Upload indexed contents of the local branch to the remote repository branch
- Update the local repository with all changes from the remote repository:
  git pull
  Or
  git pull <remote> <branch>
- Show the working tree status:
  git status
- Retrieve updates from one or multiple remote repositories:
  git fetch
  git fetch <remote>
  git fetch --all
- Combine multiple sequences of commits into one unified history:
  git merge <branch>
  git merge --commit <branch>
  git merge --no-commit <branch>
  git merge --no-ff <branch>
  git merge --ff <branch>
  git merge --ff-only <branch>
  git merge --squash <branch>
- Change current branch:
  git checkout <branch_name>

- Create and switch to a new branch:
  git checkout -b <new_branch_name>
- Show the commit history of the current branch:
  git log
- Show changes made to HEAD:
  git reflog
- Reset the current HEAD to the specified state:
  git reset --soft <commit_hash>
  git reset --hard <commit_hash>
  We can use this to go back to a specific commit
- Undo changes by creating a new commit:
  git revert <commit_hash>
  git revert <start_commit_hash>..<end_commit_hash>
- Undo changes by creating a new commit with an inline commit message:
  git revert -m <commit_message> <commit_hash>
- Remove some commits from the remote repository:
  git reset --hard <commit_hash>
  git push <remote> <branch> --force
- Take and apply a specific or a range of commits from one branch to another branch:
  git cherry-pick <commit_hash>
  Or
  git cherry-pick <start_commit_hash>..<end_commit_hash>
- Go back to a specific commit in the detached HEAD state:
  git checkout <commit_hash>
- Create a new branch from the current state of detached HEAD:
  git checkout <commit_hash>
  git checkout -b <branch>
- Put back detached HEAD to an existing branch:
  git checkout <branch>
  git merge <detached-HEAD-commit>
  Or
  git cherry-pick <detached-HEAD-commit>
- Reapply commits on top of another base tip:
  git rebase <branch>
- Remove all the changes from the working directory:
  git stash