cpu CODE SCHOOL

SETUP

- `git config --get init.defaultBranch`
  get git’s default branch
- `git config --global init.defaultBranch <branch>`
  set git’s default branch
- `git init`
  turn a directory into a git repo
- `git clone`
  create a copy of the target repo

STAGE AND SNAPSHOT

- `git diff`
  show differences between the working tree and index
- `git diff --cached`
  shows differences between the index and repo
- `git log`
  displays the commit history
- `git status`
  tells you what has changed in your directory - lists which files are staged, not staged, or not tracked
- `git add <file> / .`
  add a file / all changed files to the stage
- `git commit -m “<message>”`
  commit all staged files
- `git commit -am “<message>”`
  stage and commit edited files in one command
- `git show <commit>`
  show the commit info and the textual difference of the specified commit’s changes

UNDOING CHANGES

- I staged changes and then made new changes. I don’t want the new changes. I want to go back to the staged version.
  `git restore .`
- I staged a change but now want to commit a different file. How do I unstage?
  `git restore --staged .`
- I staged 2 files, but I only want to commit one of them. How do I unstage the other one?
  `git restore --staged <file name>`
- I don’t want any changes. I want to go back to the commited version
  `git restore --worktree --staged .`
- I commited some work. I want to completely trash it and go back to an earlier snapshot of my repo
  `git reset <commit> --hard`
- I commited some work. I want to delete the latest commit but keep my latest version as a staged change.
  `git reset <commit> --soft`
- I committed some work. I want to delete the latest commit and unstage my changes, but keep my latest changes in the working tree.
  `git reset <commit>`

cpu CODE SCHOOL

GIT BRANCHES

- `git branch`

  show branch that you're in
- `git branch -vv`

  show branches and their remotes
- `git branch <branch>`

  create new branch
- `git switch <branch>`

  switch branches
- `git switch -c <branch>`

  create a new branch and switch to it
- `git branch -d <branch>`

  delete a branch
- `git stash`

  save uncommitted changes for later use and remove the changes from your working tree
- `git stash list`

  list stash entries in stack-order
- `git stash clear`

  remove all stash entries
- `git stash pop`

  apply stashed changes to the working tree
- `git merge <branch>`

  merge a branch into the current branch

REMOTE REPOSITORIES

- `git remote add origin <url>`

  establish a remote connection with the repo
- `git push -u origin <branch>`

  set a remote connection for specified branch
- `git push --all -u origin`

  push changes in all branches
- `git fetch`

  downloads all branches and new changes from remote, but does not impact the index or working tree
- `git push`

  push the committed files in the current branch to the remote
- `git pull`

  update your current branch with upstream changes

BASIC CLI COMMANDS

- `cd`

  change the current directory
- `mkdir <directory>`

  create a new directory
- `dir`

  list the directories and files contained within the current directory
- `type <file>`

  get the file’s contents
- `del <file>`

  delete a file
- `cls`

  clear the CLI screen
- `echo [content] > [file]`

  create a text file with given content
- `notepad <file>`

  open the file in notepad