GitHub CLI GitHub from the terminal

CONCEPTS

Inside a git clone, gh picks the repo from the remotes. Elsewhere add -R <owner>/<repo>.

Branch local commits
↓ `gh pr create`
Pull request review and checks
↓ `gh pr merge`
Merged base branch
↓ `gh run watch`
Actions run CI and deploy

AUTH AND SETUP

`gh auth login`
Sign in interactively

`gh auth status`
Show who you are signed in as

REPOSITORIES

`gh repo clone <owner>/<repo>`
Clone a repository

`gh repo create <name> --public --clone`
Create a repo and clone it

`gh repo create <name> --private --source=. --push`
Publish the current folder

`gh repo view --web`
Open the repo in the browser

PULL REQUESTS

`gh pr create --fill`
Open a PR with title from commits

`gh pr create --draft --title "<t>" --body "<b>"`
Open a draft PR

`gh pr list`
List open PRs

`gh pr view <number>`
Show a PR

PULL REQUESTS (cont.)

`gh pr checkout <number>`
Check out a PR branch

`gh pr status`
PRs that involve you

`gh pr checks --watch`
Watch CI checks on this PR

`gh pr review --approve`
Approve a PR

`gh pr merge --squash --delete-branch`
Squash-merge and clean up

ISSUES

`gh issue list`
List open issues

`gh issue create --title "<t>" --body "<b>"`
File an issue

`gh issue view <number>`
Show an issue

`gh issue list --assignee @me`
Issues assigned to you

ACTIONS

`gh run list`
Recent workflow runs

`gh run view <run-id>`
Summary of a run

`gh run watch`
Follow a run live

`gh run view <run-id> --log-failed`
Logs of failed steps only

`gh run rerun <run-id> --failed`
Rerun only failed jobs

API AND SCRIPTING

`gh api repos/{owner}/{repo}/issues`
GET a REST endpoint

`gh api <endpoint> --jq '<expr>'`
Filter the response with jq