

GITHUB CLI 빠른 참조

저장소, 이슈, PR, Actions, 릴리즈, API

설정	
설치	
<code>brew install gh</code>	macOS - Homebrew
<code>sudo apt install gh</code>	Debian / Ubuntu
<code>winget install GitHub.cli</code>	Windows - winget
<code>conda install gh</code>	conda-forge
인증	
<code>gh auth login</code>	# interactive login
<code>gh auth login --with-token < token.txt</code>	
<code>gh auth status</code>	# check auth state
<code>gh auth refresh -s repo,gist</code>	# add scopes

설정	
<code>gh config set editor vim</code>	
<code>gh config set pager less</code>	
<code>gh config set git_protocol ssh</code>	
<code>gh config list</code>	

저장소	
저장소 명령	
<code>gh repo create my-app --public --clone</code>	
<code>gh repo clone owner/repo</code>	
<code>gh repo fork owner/repo --clone</code>	
<code>gh repo view owner/repo --web</code>	
저장소 옵션	
<code>--public --private</code>	저장소 공개 여부 설정
<code>--template owner/repo</code>	템플릿 저장소에서 생성
<code>--clone</code>	생성 후 클론
<code>--add-readme</code>	README로 초기화
<code>gh repo list owner</code>	소유자의 저장소 목록
<code>gh repo delete owner/repo</code>	저장소 삭제 (확인 필요)
<code>gh repo rename new-name</code>	현재 저장소 이름 변경
<code>gh repo archive owner/repo</code>	저장소 아카이브

이슈	
이슈 관리	
<code>gh issue create --title "Bug" --body "Details here"</code>	
<code>gh issue list --state open --label bug</code>	
<code>gh issue view 42</code>	
<code>gh issue close 42 --reason completed</code>	
이슈 옵션	
<code>--assignee @me</code>	본인에게 할당
<code>--label bug,urgent</code>	레이블 추가
<code>--milestone v2.0</code>	마일스톤 설정
<code>--project "Board"</code>	프로젝트에 추가
<code>gh issue edit 42</code>	이슈 대화형 편집
<code>gh issue reopen 42</code>	닫힌 이슈 다시 열기
<code>gh issue comment 42 -b "msg"</code>	이슈에 댓글 추가
<code>gh issue pin 42</code>	이슈 저장소에 고정

플리킷리스트	
PR 생성 & 관리	
<code>gh pr create --title "feat: add auth" --body "..."</code>	
<code>gh pr create --fill</code>	# title/body from commits
<code>gh pr list --state open</code>	
<code>gh pr view 123 --web</code>	
리뷰 & 머지	
<code>gh pr checkout 123</code>	# check out PR branch
<code>gh pr diff 123</code>	# view PR diff
<code>gh pr review 123 --approve</code>	
<code>gh pr merge 123 --squash --delete-branch</code>	
PR 옵션	
<code>--draft</code>	드래프트 PR로 생성
<code>--reviewer user1,user2</code>	리뷰어 요청
<code>--base main</code>	베이스 브랜치 설정
<code>--merge --squash --rebase</code>	머지 전략
<code>--auto</code>	검사 통과 시 자동 머지
<code>--delete-branch</code>	머지 후 브랜치 삭제
<code>gh pr ready 123</code>	드래프트 PR을 준비 완료로 표시

Actions	
워크플로우 명령	
<code>gh run list</code>	# recent runs
<code>gh run view 12345</code>	# run details
<code>gh run view 12345 --log-failed</code>	# failed step logs
<code>gh run watch 12345</code>	# live status
실행 & 관리	
<code>gh workflow run deploy.yml --ref main</code>	
<code>gh workflow list</code>	
<code>gh workflow view deploy.yml</code>	
<code>gh run rerun 12345 --failed</code>	# rerun failed jobs
Actions 옵션	
<code>-f key=value</code>	workflow_dispatch에 입력 전달
<code>--json</code>	JSON으로 출력
<code>-b branch</code>	브랜치로 필터
<code>gh run download 12345</code>	실행 아티팩트 다운로드
<code>gh cache list</code>	Actions 캐시 목록
<code>gh cache delete KEY</code>	캐시 항목 삭제

릴리즈	
릴리즈 관리	
<code>gh release create v1.0.0 --generate-notes</code>	
<code>gh release create v1.0.0 ./dist/*.tar.gz</code>	
<code>gh release list</code>	
<code>gh release view v1.0.0</code>	
릴리즈 옵션	
<code>--title "Release v1.0.0"</code>	릴리즈 제목 설정

<code>--notes "Changelog here"</code>	릴리즈 노트 인라인 설정
<code>--notes-file CHANGELOG.md</code>	파일에서 노트 가져오기
<code>--generate-notes</code>	커밋에서 자동 생성
<code>--draft</code>	드래프트로 생성
<code>--pre-release</code>	사전 릴리즈용 표시
<code>--latest</code>	최신 릴리즈로 표시
<code>gh release download v1.0.0</code>	릴리즈 아티팩트 다운로드
<code>gh release delete v1.0.0</code>	릴리즈 삭제
<code>gh release edit v1.0.0</code>	릴리즈 메타데이터 편집

Gist	
Gist 명령	
<code>gh gist create file.py -d "My snippet"</code>	
<code>gh gist create file1.js file2.js # multi-file gist</code>	
<code>gh gist list</code>	
<code>gh gist view <id></code>	
Gist 옵션	
<code>-d "description"</code>	Gist 설명 설정
<code>--public</code>	공개 Gist 생성 (기본: 비공개)
<code>--web</code>	브라우저에서 Gist 열기
<code>gh gist edit <id></code>	Gist 파일 편집
<code>gh gist clone <id></code>	Gist 로컬 클론
<code>gh gist delete <id></code>	Gist 삭제

API	
API 호출	
<code>gh api repos/owner/repo</code>	
<code>gh api repos/owner/repo/issues --method POST \</code>	
<code>-f title="Bug" -f body="Details"</code>	
<code>gh api graphql -f query='{ viewer { login } }'</code>	
출력 형식 지정	
<code>gh api repos/owner/repo --jq 'stargazers count'</code>	
<code>gh api repos/owner/repo --template '{{ full_name }}'</code>	
<code>gh pr list --json number,title --jq '[:].title'</code>	

API 옵션	
<code>--method GET POST PUT DELETE</code>	HTTP 메서드
<code>-f key=value</code>	문자열 필드 설정
<code>-F key=@file</code>	파일에서 필드 설정
<code>--jq 'expression'</code>	jq 문법으로 JSON 필터
<code>--template 'tmpl'</code>	Go 템플릿으로 형식 지정
<code>--paginate</code>	모든 페이지 가져오기
<code>-H 'Accept: ...'</code>	커스텀 헤더 설정

별칭	
별칭 관리	
<code>gh alias set co 'pr checkout'</code>	
<code>gh alias set bugs 'issue list --label bug'</code>	
<code>gh alias set last 'run list -L 1'</code>	
<code>gh alias list</code>	
고급 별칭	
<code># Alias with shell command</code>	
<code>gh alias set --shell pv 'gh pr view --json url --jq .url pbcopy'</code>	
<code># Delete alias</code>	
<code>gh alias delete co</code>	

별칭 팁	
<code>gh alias set name 'cmd'</code>	단순 별칭 생성
<code>--shell</code>	셸을 통해 실행 (파이프 지원)
<code>gh alias list</code>	정의된 모든 별칭 표시
<code>gh alias delete name</code>	별칭 제거
일괄 페더	
일상 워크플로우	
<code>gh issue list --assignee @me</code>	# my open issues
<code>gh pr status</code>	# PRs needing attention
<code>gh pr checks 123</code>	# CI status for PR
<code>gh run list -b main -L 5</code>	# recent CI runs
검색	
<code>gh search repos --language rust --stars ">1000"</code>	
<code>gh search issues --repo owner/repo "memory leak"</code>	
<code>gh search prs --state open --review required</code>	

유용한 플래그	
<code>--json field1,field2</code>	특정 필드를 JSON으로 출력
<code>--jq 'expr'</code>	JSON 출력 필터
<code>-L N</code>	결과를 N개로 제한
<code>--web</code>	브라우저에서 결과 열기
<code>-R owner/repo</code>	특정 저장소 대상 지정
<code>GH_TOKEN=xxx</code>	환경 변수로 인증