

# GitHub CLI 빠른 참조

저장소, 이슈, PR, Actions, 릴리즈, API

## 설정

### 설치

<b>brew install gh</b>	macOS - Homebrew
<b>sudo apt install gh</b>	Debian / Ubuntu
<b>winget install GitHub.cli</b>	Windows - winget
<b>conda install gh</b>	conda-forge

### 인증

<b>gh auth login</b>	# interactive login
<b>gh auth login --with-token &lt; token.txt</b>	
<b>gh auth status</b>	# check auth state
<b>gh auth refresh -s repo,gist</b>	# add scopes

### 설정

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

## 저장소

### 저장소 명령

<b>gh repo create my-app --public --clone</b>
<b>gh repo clone owner/repo</b>
<b>gh repo fork owner/repo --clone</b>
<b>gh repo view owner/repo --web</b>

### 저장소 옵션

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

## 이슈

### 이슈 관리

<b>gh issue create --title "Bug" --body "Details here"</b>
<b>gh issue list --state open --label bug</b>
<b>gh issue view 42</b>
<b>gh issue close 42 --reason completed</b>

### 이슈 옵션

<b>--assignee @me</b>	본인에게 할당
<b>--label bug,urgent</b>	레이블 추가
<b>--milestone v2.0</b>	마일스톤 설정
<b>--project "Board"</b>	프로젝트에 추가
<b>gh issue edit 42</b>	이슈 대화형 편집
<b>gh issue reopen 42</b>	닫힌 이슈 다시 열기
<b>gh issue comment 42 -b "msg"</b>	이슈에 댓글 추가
<b>gh issue pin 42</b>	이슈 저장소에 고정

## 풀 리퀘스트

### PR 생성 & 관리

<b>gh pr create --title "feat: add auth" --body "..."</b>	
<b>gh pr create --fill</b>	# title/body from commits
<b>gh pr list --state open</b>	
<b>gh pr view 123 --web</b>	

## 리뷰 & 머지

<b>gh pr checkout 123</b>	# check out PR branch
<b>gh pr diff 123</b>	# view PR diff
<b>gh pr review 123 --approve</b>	
<b>gh pr merge 123 --squash --delete-branch</b>	

### PR 옵션

<b>--draft</b>	드래프트 PR로 생성
<b>--reviewer user1,user2</b>	리뷰어 요청
<b>--base main</b>	베이스 브랜치 설정
<b>--merge   --squash   --rebase</b>	머지 전략
<b>--auto</b>	검사 통과 시 자동 머지
<b>--delete-branch</b>	머지 후 브랜치 삭제
<b>gh pr ready 123</b>	드래프트 PR을 준비 완료로 표시

## Actions

### 워크플로우 명령

<b>gh run list</b>	# recent runs
<b>gh run view 12345</b>	# run details
<b>gh run view 12345 --log-failed</b>	# failed step logs
<b>gh run watch 12345</b>	# live status

### 실행 & 관리

<b>gh workflow run deploy.yml --ref main</b>	
<b>gh workflow list</b>	
<b>gh workflow view deploy.yml</b>	
<b>gh run rerun 12345 --failed</b>	# rerun failed jobs

### Actions 옵션

<b>-f key=value</b>	workflow_dispatch에 입력 전달
<b>--json</b>	JSON으로 출력
<b>-b branch</b>	브랜치로 필터
<b>gh run download 12345</b>	실행 아티팩트 다운로드
<b>gh cache list</b>	Actions 캐시 목록
<b>gh cache delete KEY</b>	캐시 항목 삭제

## 릴리즈

### 릴리즈 관리

<b>gh release create v1.0.0 --generate-notes</b>
<b>gh release create v1.0.0 ./dist/*.tar.gz</b>
<b>gh release list</b>
<b>gh release view v1.0.0</b>

### 릴리즈 옵션

<b>--title "Release v1.0"</b>	릴리즈 제목 설정
<b>--notes "Changelog here"</b>	릴리즈 노트 인라인 설정
<b>--notes-file CHANGELOG.md</b>	파일에서 노트 가져오기
<b>--generate-notes</b>	커밋에서 자동 생성
<b>--draft</b>	드래프트로 생성
<b>--prerelease</b>	사전 릴리즈로 표시
<b>--latest</b>	최신 릴리즈로 표시
<b>gh release download v1.0.0</b>	릴리즈 에셋 다운로드
<b>gh release delete v1.0.0</b>	릴리즈 삭제
<b>gh release edit v1.0.0</b>	릴리즈 메타데이터 편집

## Gist

### Gist 명령

<b>gh gist create file.py -d "My snippet"</b>	
<b>gh gist create file1.js file2.js</b>	# multi-file gist
<b>gh gist list</b>	
<b>gh gist view &lt;id&gt;</b>	

## Gist 옵션

<b>-d "description"</b>	Gist 설명 설정
<b>--public</b>	공개 Gist 생성 (기본: 비공개)
<b>--web</b>	브라우저에서 Gist 열기
<b>gh gist edit &lt;id&gt;</b>	Gist 파일 편집
<b>gh gist clone &lt;id&gt;</b>	Gist 로컬 클론
<b>gh gist delete &lt;id&gt;</b>	Gist 삭제

## API

### API 호출

<b>gh api repos/owner/repo</b>
<b>gh api repos/owner/repo/issues --method POST \</b> <b>-f title="Bug" -f body="Details"</b>
<b>gh api graphql -f query='{ viewer { login } }'</b>

### 출력 형식 지정

<b>gh api repos/owner/repo --jq '.stargazers_count'</b>
<b>gh api repos/owner/repo --template '{{.full_name}}'</b>
<b>gh pr list --json number,title --jq '[][.title]'</b>

### API 옵션

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

## 별칭

### 별칭 관리

<b>gh alias set co 'pr checkout'</b>
<b>gh alias set bugs 'issue list --label bug'</b>
<b>gh alias set last 'run list -L 1'</b>
<b>gh alias list</b>

### 고급 별칭

<b># Alias with shell command</b>
<b>gh alias set --shell pv 'gh pr view --json url --jq .url   pbcopy'</b>
<b># Delete alias</b>
<b>gh alias delete co</b>

### 별칭 팁

<b>gh alias set name 'cmd'</b>	단순 별칭 생성
<b>--shell</b>	셸을 통해 실행 (파이프 지원)
<b>gh alias list</b>	정의된 모든 별칭 표시
<b>gh alias delete name</b>	별칭 제거

## 일반 패턴

### 일상 워크플로우

<b>gh issue list --assignee @me</b>	# my open issues
<b>gh pr status</b>	# PRs needing attention
<b>gh pr checks 123</b>	# CI status for PR
<b>gh run list -b main -L 5</b>	# recent CI runs

### 검색

<b>gh search repos --language rust --stars "&gt;1000"</b>
<b>gh search issues --repo owner/repo "memory leak"</b>
<b>gh search prs --state open --review required</b>

# GitHub CLI 빠른 참조

---

## 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