

0. 목차

1. Git 개요와 설치

- 버전 관리의 필요성과 종류 (로컬 vs 분산)
 - Git이란? Git의 철학 (작은 단위, 분산 구조)
 - Git vs SVN
 - Git 설치 및 설정 (`git config`)
 - 사용자 정보 설정 (`user.name`, `user.email`)
 - 에디터, 라인 끝 처리 등 기타 설정
 - Git GUI 도구 (SourceTree, GitKraken 등)
-

2. 기본 Git 명령어

- 저장소 생성 및 클론
 - `git init`, `git clone`
 - 스테이징과 커밋
 - `git add`, `git commit`
 - 커밋 메시지 컨벤션 (예: Conventional Commits)
 - 상태 확인
 - `git status`, `git diff`
 - 로그 확인
 - `git log`, `git log --graph`, `git show`
-

3. 파일 추적 및 무시 설정

- `.gitignore` 작성법
 - `.gitkeep`의 의미
 - `git clean`, `git rm`, `git mv`
 - 대소문자 변경 추적 이슈 (`core.ignorecase`)
-

4. 브랜치와 병합

- 브랜치 생성과 전환
 - `git branch`, `git checkout -b`
- 병합(Merge)
 - Fast-forward vs 3-way merge
- 충돌(Conflict) 해결

- 브랜치 삭제
 - `git log --oneline --graph` 로 브랜치 시각화
-

5. Rebase와 히스토리 조작

- `git rebase`란 무엇인가?
 - vs `git merge`
 - `git rebase -i`로 커밋 정리 (squash, fixup 등)
 - `git commit --amend`
 - `git reset` (soft, mixed, hard)
 - `git reflog`로 복구
-

6. 원격 저장소(GitHub 등) 연동

- `git remote` 명령어
 - `git push`, `git fetch`, `git pull`
 - 원격 브랜치 추적
 - 브랜치 푸시/삭제 (`git push origin branch-name`, `--delete`)
 - origin 외 다른 remote 설정 (upstream 등)
-

7. GitHub의 핵심 기능

- GitHub 계정 생성 및 SSH 키 설정
 - Repository 만들기 (공개/비공개)
 - README, LICENSE, .gitignore 자동 생성
 - Issues, Discussions, Projects (칸반보드)
 - Pull Request 생성 및 리뷰
 - Fork와 Upstream Pull 구조
 - GitHub Actions 간단 소개
-

8. 협업 워크플로우

- Git Flow 전략
- GitHub Flow 전략
- trunk-based 개발 방식
- Fork 기반 협업 전략
- Feature 브랜치 전략
- Pull Request + 코드 리뷰 프로세스

9. 태그와 릴리즈

- 버전 태그 달기 (`git tag`)
 - lightweight vs annotated tag
 - 이전 커밋에 태그 추가
 - GitHub 릴리즈 페이지 연동
 - `changelog` 작성 전략
-

10. 서브모듈과 서브트리

- 서브모듈 개념 및 설정
 - `git submodule add`, `init`, `update`
 - 의존성 관리
 - 서브트리 방식과 비교
 - `git subtree add`, `pull`, `push`
-

11. Git 내부 구조 및 원리

- `.git` 디렉토리 구조
 - Blob, Tree, Commit 객체
 - HEAD 포인터의 의미
 - SHA-1 해시 구조
 - staging area 동작 방식
-

12. 충돌 해결 및 디버깅

- `git mergetool` 사용
 - 충돌 해결 전략 및 커뮤니케이션
 - `git bisect` 로 문제 커밋 추적
 - `git blame`, `git annotate` 로 이력 추적
-

13. CI/CD와 GitHub 연동

- GitHub Actions 기본 구조
 - workflow, job, step
- push/pull request 트리거 자동화
- 테스트/빌드/배포 자동화
- secret 관리

- 외부 서비스 연동 (Slack, AWS 등)
-

14. 보안과 접근 제어

- GitHub 공개/비공개 저장소 설정
 - Collaborator 권한 관리
 - GitHub Team, Organization 관리
 - GPG 서명 커밋 (Verified 배지)
 - SSH 키와 Personal Access Token 관리
-

15. 고급 기능 및 자동화

- `git stash` 로 변경사항 임시 저장
 - `git cherry-pick`
 - `git filter-branch`, `git rebase --root`
 - `.gitattributes` 설정
 - eol 처리
 - diff/merge 전략
 - Git Hooks (`pre-commit`, `commit-msg`, `post-merge`)
 - husky, lint-staged와의 연동
-

16. GitHub Copilot 및 AI 도구

- GitHub Copilot 개요 및 설치
 - 코드 자동 완성
 - Pull Request 자동 요약
 - GitHub CLI 활용법
-

17. GitHub Pages

- 정적 사이트 배포
 - 커스텀 도메인 연결
 - Jekyll로 블로그 만들기
 - React 앱 배포 (gh-pages 브랜치)
-

18. 실전 예제 및 프로젝트

- React 프로젝트 Git 버전 관리 적용
- 협업용 Repository 템플릿 설계

- 블로그 템플릿 배포 with GitHub Pages
- GitHub Actions를 활용한 자동 테스트/배포 구성