

GitHub

GIT CHEAT SHEET

Git은 컴퓨터에서 로컬로 발생하는 GitHub와 관련된 모든 작업을 담당하는 무료 오픈 소스 분산 버전 제어 시스템이다. 이 치트 시트에는 쉽게 참조할 수 있도록 가장 중요하고 일반적으로 사용되는 Git 명령이 포함되어 있다.

설치 & GUI

플랫폼에 맞는 Git 설치 프로그램을 통해 GitHub는 최신 CLI 버전을 손쉽게 업데이트할 수 있는 편의성을 제공하며, 일상적인 작업, 리뷰 및 저장소 동기화를 위한 GUI도 제공한다.

Windows용 GitHub

<https://windows.github.com>

Mac용 GitHub

<https://mac.github.com>

Linux 및 Solaris 플랫폼의 경우 공식 Git 웹 사이트에서 최신 릴리스를 사용할 수 있다.

모든 플랫폼을 위한 Git

<http://git-scm.com>

사용자 설정

모든 로컬 저장소에 공통으로 사용되는 사용자 정보 설정

```
git config --global user.name "[이름]"
```

변경 이력을 확인할 때 사용될 이름 설정

```
git config --global user.email "[유효한_이메일]"
```

각 변경 이력에 연결될 이메일 주소 설정

```
git config --global color.ui auto
```

Git 명령어 결과를 쉽게 확인할 수 있도록 자동 색상 설정 활성화

설정 & 초기화

저장소 초기화 및 복제

```
git init
```

기존 디렉터리를 Git 저장소로 초기화

```
git clone [url]
```

URL을 통해 호스팅된 저장소를 복제하여 로컬에 저장

스태이징 & 스냅샷

스냅샷 작업 및 Git 스태이징 영역

```
git status
```

다음 커밋을 위해 스태이징된, 작업 디렉터리에서 수정된 파일 표시

```
git add [파일명]
```

현재 상태의 파일을 다음 커밋에 추가(스태이징)

```
git reset [파일명]
```

파일의 스태이징을 취소하되, 작업 디렉터리의 변경 사항은 유지

```
git diff
```

변경되었으나 아직 스태이징 되지 않은 내용 비교

```
git diff --staged
```

스태이징 되었으나 아직 커밋되지 않은 변경 사항 비교

```
git commit -m "[커밋_내용]"
```

스태이징된 내용을 새 커밋으로 스냅샷하여 기록

브랜치 & 병합

브랜치를 통한 작업 분리, 컨텍스트 변경 및 변경 사항 병합

```
git branch
```

브랜치를 목록으로 표시 (활성 브랜치 옆에 * 표시)

```
git branch [branch-name]
```

현재 커밋에서 새 브랜치 생성

```
git checkout
```

다른 브랜치로 전환하고 해당 내용을 작업 디렉터리로 가져오기

```
git merge [branch]
```

지정된 브랜치의 변경 이력을 현재 브랜치로 병합

```
git log
```

현재 브랜치의 모든 커밋 이력 표시



검사 & 비교

로그, 변경 사항(diff), 객체 정보 검사

git log
현재 활성 브랜치의 커밋 이력 표시
git log 브랜치_B..브랜치_A
브랜치 B에는 없고 브랜치 A만 있는 커밋 표시
git log --follow [파일명]
파일 이름 변경 이력을 포함하여 파일이 변경된 커밋 표시
git diff 브랜치_B...브랜치_A
브랜치 A에만 있고 브랜치 B에는 없는 변경 사항(diff) 표시
git show [SHA]
Git 객체를 사람이 읽을 수 있는 형식으로 표시

경로 변경 추적

파일 삭제 및 경로 변경의 버전 관리

git rm [파일명]
프로젝트에서 파일을 삭제하고, 제거를 커밋할 수 있도록 스테이징
git mv [기존_경로] [새_경로]
기존 파일 경로를 변경하고, 변경 사항 스테이징
git log --stat -M
경로가 이동된 모든 커밋 로그 표시

패턴 무시

의도치 않은 파일 스테이징 또는 커밋 방지

logs/ *.notes pattern*/
원하는 패턴이 포함된 파일을 직접 문자열 일치 또는 와일드카드로 .gitignore에 저장
git config --global core.excludesfile [파일명]
모든 로컬 저장소에 대한 시스템 전체 무시 패턴

공유 & 업데이트

다른 저장소에서 업데이트 가져오기 및 로컬 저장소 업데이트

git remote add [별칭] [url]
Git URL을 별칭으로 추가
git fetch [별칭]
지정된 원격 저장소의 모든 브랜치를 가져오기
git merge [별칭]/[branch]
원격 브랜치의 변경 사항을 현재 브랜치로 통합
git push [별칭] [branch]
로컬 브랜치의 커밋을 원격 저장소의 브랜치로 전송
git pull
추적 중인 원격 브랜치에서 커밋을 가져와 병합

이력 재작성

브랜치 재작성, 커밋 업데이트 및 이력 정리

git rebase [branch]
현재 브랜치의 모든 커밋을 지정한 브랜치 앞에 적용
git reset --hard [commit]
스테이징 영역 초기화, 지정한 커밋으로 작업 디렉터리 재작성

임시 커밋

브랜치 변경을 위해 수정된 파일 임시 저장

git stash
수정되고 스테이징된 변경 사항 임시 저장
git stash list
임시 저장된 변경 사항의 스택 순서를 목록으로 표시
git stash pop
스택의 최상위 임시 저장 항목을 다시 작업 디렉터리로 가져오기
git stash drop
스택의 최상위 임시 저장 항목 삭제

GitHub Education

Teach and learn better, together. GitHub is free for students and teachers. Discounts available for other educational uses.

✉ education@github.com
🌐 education.github.com