

SETUP

<div>Git config --global user.name "[firstname Last name]"</div> <div>설명:버전 기록을 검토할 때 신용으로 식별할 수 있는 이름을 설정합니다.</div>
<div>git config --global user.email "[valid-email]"</div> <div>설명: 각 기록 마커와 연결할 전자 메일 주소 설정</div>
<div>git config --global color.ui auto</div> <div>설명: 쉽게 검토할 수 있도록 Git에 대한 자동 명령줄 색상 설정</div>

SETUP & INIT

<div>git init</div> <div>설명: 기존 디렉터리를 Git 저장소로 초기화</div>
<div>git clone [url]</div> <div>설명: URL을 통해 호스트 위치에서 전체 리포지토리 검색</div>

STAGE & SNAPSHOT

<div>git status</div> <div>설명: 다음 커밋(단계)에 파일을 지금처럼 추가</div>
<div>git add [file]</div> <div>설명: 작업 디렉터리의 변경 사항을 유지하면서 파일 스테이징 해제</div>
<div>Git diff</div> <div>설명 : WD와 SA의 차이</div>
<div>git diff diff</div> <div>설명: 저장되었지만 아직 커밋되지 않은 것의 차이</div>

BRANCH & MERGE

<div>git branch</div> <div>설명: 분기를 나열합니다. *가 현재 활성 분기 옆에 나타납니다.</div>
<div>git branch [branch-name]</div> <div>설명: 현재 커밋에서 새 분기 생성</div>
<div>git checkout</div> <div>설명: 다른 지점으로 전환하여 작업 디렉토리로 체크아웃합니다.</div>
<div>git merge [branch]</div> <div>설명: 지정한 분기의 기록을 현재 기록으로 병합</div>
<div>git log</div> <div>설명: 현재 분기 기록의 모든 커밋 표시</div>

Inspect & Compare

git log

설명 : 현재 활성 분기에 대한 커밋 기록 표시

```
git log branchB..branchA
```

설명 : 지점 B에 없는 지점 A의 커밋을 표시합니다.

```
git log --follow [file]
```

설명 : 변경된 파일의 커밋 표시(이름 변경 포함)

```
git diff branchB...branchA
```

설명 : 지점 A에 있는 것과 지점 B에 없는 것의 차이를 보여줍니다.

```
git show [SHA]
```

설명 : 사람이 읽을 수 있는 형식으로 Git에 개체를 표시합니다.

Tracking path Changes

git rm [file]

설명 : 프로젝트에서 파일을 삭제하고 커밋을 위해 제거 단계를 수행합니다.

```
git mv [existing-path] [new-path]
```

설명 : 기존 파일 경로 변경 및 이동 준비

```
git log --stat -M
```

설명 : 이동한 경로를 나타내는 모든 커밋 로그 표시

Share & update

```
git remote add [alias] [url]
```

설명 : Git URL을 별칭으로 추가합니다.

```
git merge [alias]/[branch]
```

설명 : 원격 지점을 현재 지점에 병합합니다.
그것을 최신 상태로 만듭니다.

```
git push [alias] [branch]
```

설명 : 로컬 분기 커밋을 원격 리포지토리 분기로 전송

```
git pull
```

설명 : 병합

Rewrite history

```
git rebase [branch]
```

설명 : 지정된 분기보다 앞서 현재 분기의 커밋 적용

```
git reset --hard [commit]
```

설명 : 준비 영역 지우기, 지정된 커밋에서 작업 트리 다시 쓰기

Temporary commits

git stash list

설명 : 스택 순서 저장된 파일 변경 나열

```
git stash pop
```

설명 : 스택에 남아 있는 stash는 위의 명령어를 사용하여 제거할 수 있다.

```
git stash drop
```

설명 : 저장 스택의 맨 위에서 변경 사항을 삭제합니다.