

- add

파일의 변경사항을 다음 커밋에 반영하기
현재 directory의 모든 파일을 staging area로 이동

```
$ git add [file]
```

모든 변경 사항을 다음 커밋에 반영하기

```
$ git add . / $ git add -all
```

- log

commit history 확인 (마지막 커밋인 head부터 이전 모든 이력 로그 표시)

```
$ git log
```

한 줄로 commit history 보기

```
$ git log --oneline
```

한 줄로 그래프 형태로 commit history 보기

```
$ git log --oneline --graph
```

```
$ git log --oneline --graph --all
```

commit의 변경 사항까지 확인하기

```
$ git log -p [file]
```

- clone

기존에 사용 중인 저장소를 clone해서 가져오기

```
$ git clone [url]
```

```
$ git clone [url] [새로운 폴더명]
```

- commit

staging의 파일들을 메시지와 함께 커밋하기

```
$ git commit -m "commit message"
```

add하고 commit 동시에 하기

```
$ git commit -am "commit message"
```

마지막 커밋 수정하기

```
$ git commit --amend
```

- show

현재 branch의 가장 최근 커밋 정보 확인하기

```
$ git show
```

특정 커밋 정보 확인하기

```
$ git show [commit id]
```

특정 branch의 가장 최근 커밋 정보 확인하기

```
$ git show [branch]
```

- push

로컬 저장소의 커밋한 변경사항을 원격 저장소로 전송하기

```
$ git push
```

```
$ git push [remote] [branch]
```

- pull

원격 저장소의 변경사항을 로컬 저장소로 가져오고 병합하기

\$ git pull

\$ git pull [remote] [branch]

원격 저장소의 변경사항을 로컬 저장소로 가져오고 리베이스하기

\$ git pull --rebase

- fetch/merge

원격 저장소의 변경사항을 로컬 저장소로 가져오기 전 변경내용 확인하기

\$ git fetch