

# 3. Git 명령어

## 저장소 만들기

저장소로 사용할 폴더 생성

```
git init
```

저장소 하위의 모든 폴더를 깃이 관리 - 중첩해서 만들지 말 것

확인하기

```
git status
```

혹시 메인 브랜치 이름이 변경되지 않았다면

```
git branch -M main
```

## Commit

파일 생성, 내용 작성, 수정, 삭제 등의 작업 후, 체크 포인트 만들어 저장하기

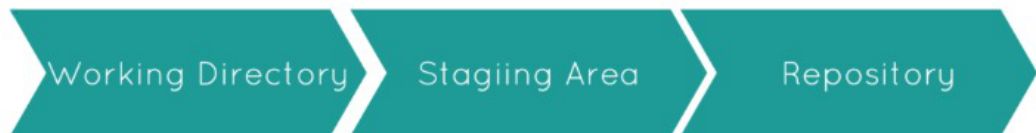
단순 파일의 저장이 아닌 현재 상태를 저장하는 개념, 포토샵의 스냅샷 혹은 피그마의 버전 관리

## 변경 사항 확인

```
git status
```



U (Untracked)



## 3 PARTS OF GIT PROJECT

### Working directory

- working tree라고 하기도 함
- 현재 작업하고 있는 영역. 즉, 작업을 하고 있는 프로젝트 디렉토리
- 아직 추적(track)하고 있는 않은 상태

### Staging area

- index라고 하기도 함
- git add 명령어로 전달한 파일들이 존재하는 영역
- 이 영역의 파일들만 commit에 반영됨

### Repository

- working directory의 변경 이력들이 저장되어 있는 영역
- commit들이 모여있는 저장소

## 변경 사항을 스테이지에 올리기

```
git add 파일이름 파일이름 파일이름
```

```
git add .
```

## commit 메시지 작성

```
git commit -m "메시지"
```

## VS Code 에서 commit 하기

### VS Code의 기본 터미널을 Git Bash로 설정

- VS Code에서 **Ctrl + Shift + P**
- **Select Default Profile** 검색하여 선택
- **Git Bash** 선택
- 터미널에서 **+**로 새 창을 열어서 기본으로 Git Bash가 설정된 것 확인

### 폴더와 파일 옆에 표시되는 글자의 의미

글자	의미	내용
<b>A</b>	Added	저장소에 새로 추가된 것들
<b>M</b>	Modified	수정된 것들
<b>D</b>	Deleted	삭제된 것들

글자	의미	내용
<b>U</b>	Untracked	새로 추가되거나 수정된 것들인데 아직 git add 되지않아 추적이 안되는 상태
<b>C</b>	Conflict	파일에 충돌이 발생했습니다.
<b>R</b>	Renamed	파일 이름이 변경되었습니다.
<b>S</b>	Submodule	다른 하위 리포지토리가 존재합니다.

git commit이 정상적으로 되면 이러한 표시들이 사라진다.

## commit 내역 확인하기

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

빠져나가기 :q

## 이전 커밋에서 잊었던 변경사항이나 파일을 포함시키기

```
git add 파일이름
git commit --amend
```