

Git & GitHub

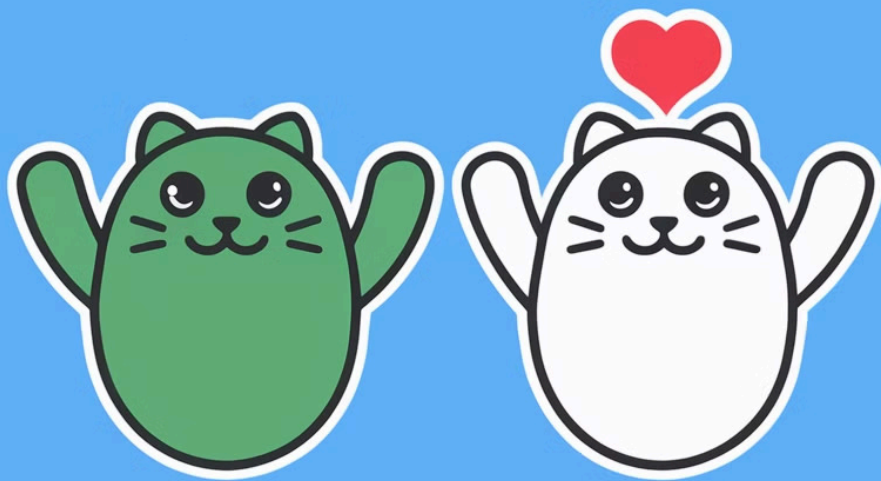
Git & GitHub 사용법 가이드

Git

분산 버전 관리 시스템으로, 파일 변경 기록을 관리하고 협업을 지원합니다.

GitHub

Git 저장소를 호스팅하는 클라우드 플랫폼으로 협업 도구를 제공합니다.



기본 용어

1 저장소(Repository)

프로젝트의 파일과 변경 이력을 저장하는 곳.

3 커밋(Commit)

파일 변경사항을 저장소에 기록.

5 풀(Pull)

원격 저장소의 변경사항을 로컬 저장소로 가져오기.

2 스테이징(Stage)

커밋 전 파일 변경사항을 준비하는 단계.

4 푸시(Push)

로컬 저장소의 커밋을 원격 저장소로 업로드.

6 브랜치(Branch)

독립적으로 작업할 수 있는 분기.



기본 설치 및 초기 설정

1

Git 설치

[Git 공식 사이트](#)에서 다운로드 후 설치.

2

Git 초기 설정

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

기본 명령어

저장소 생성

로컬에서 새 저장소 생성:

```
git init
```

기존 프로젝트를 깃허브에 연결:

```
git remote add origin <원격 저장소 URL>
```

변경사항 추적

변경된 파일 확인:

```
git status
```

변경사항 스테이징:

```
git add <파일명>
```

모든 파일 추가: `git add .`

커밋

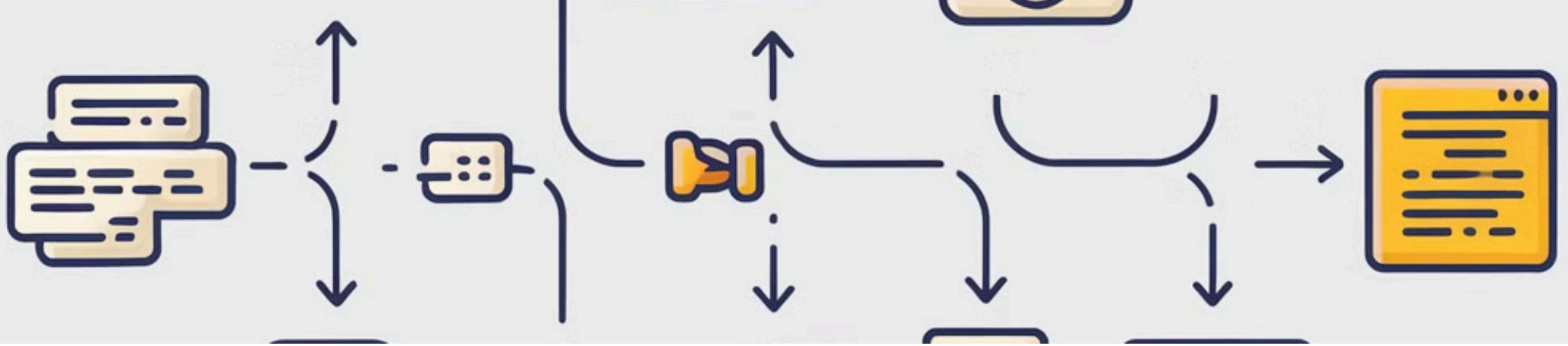
변경사항을 로컬 저장소에 저장:

```
git commit -m "커밋 메시지"
```

푸시

원격 저장소에 업로드:

```
git push origin <브랜치명>
```



깃 워크플로우 도면

- 1

Git 커밋 및 푸시

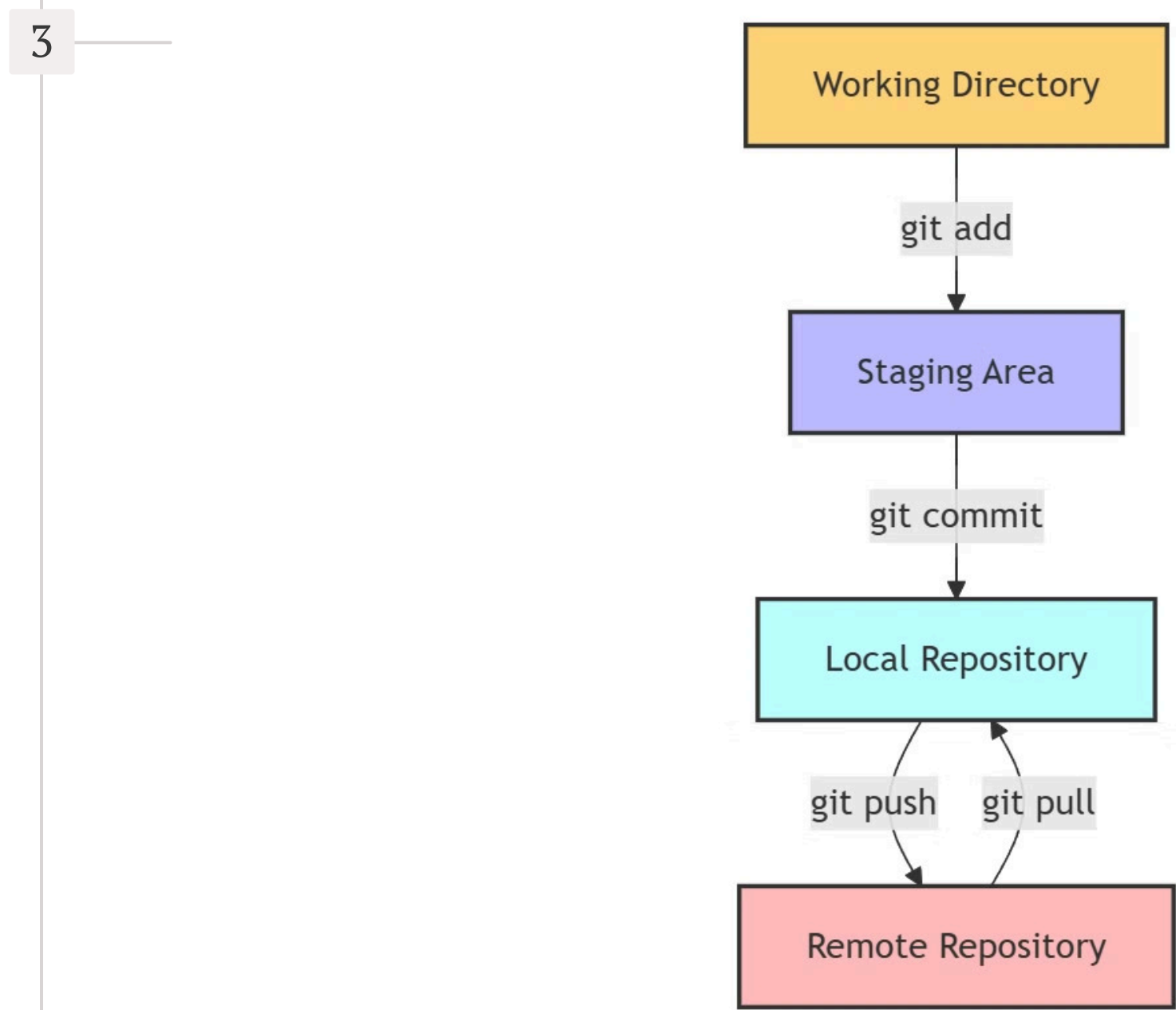
작업 파일 → (git add) → 스테이징 → (git commit) → 로컬 저장소

로컬 저장소 → (git push) → 원격 저장소

- 2

Git 풀

원격 저장소 → (git pull) → 로컬 저장소 → 작업 파일 업데이트



깃허브 기본 사용법

1

새 저장소 생성

GitHub에 로그인 → "New Repository" 클릭 → 이름 입력 및 생성.

2

원격 저장소와 로컬 저장소 연결

```
git remote add origin <깃허브 저장소 URL>
```

3

협업

다른 사용자와 브랜치를 나누어 작업 후 Pull Request 생성.