

GitHub의 기본적인 사용법과 브랜치를 다루는 방법을 초보자에게 설명할 때, 다음과 같은 순서로 설명하면 좋습니다:

1. GitHub란?

- GitHub는 코드 버전 관리를 돕는 도구로, 개발자들이 코드를 공유하고 협업할 수 있도록 도와줍니다.
- Git이라는 버전 관리 시스템 위에 웹 기반의 협업 기능을 추가한 플랫폼입니다.

2. Git과 GitHub 차이

- **Git**: 로컬 컴퓨터에서 버전 관리를 하는 도구.
- **GitHub**: Git으로 관리된 프로젝트를 웹에 업로드하고 협업하는 플랫폼.

3. GitHub 시작하기

1. **GitHub** 계정 만들기: [GitHub](#)에서 무료로 계정을 만듭니다.
2. **Git** 설치하기: [Git 설치](#) 후 로컬에서 사용할 수 있게 세팅합니다.
3. **Git** 설정:

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

4. Repository (저장소) 만들기

1. GitHub에서 "New Repository" 버튼을 클릭.
2. 저장소 이름을 입력하고 "Create Repository"를 누릅니다.

5. GitHub와 로컬 연결하기

```
git clone https://github.com/사용자명/저장소명.git
```

6. Branch 사용법

Branch는 코드를 수정하거나 새로운 기능을 추가할 때 사용하는 독립된 작업 공간입니다.

- 기본 브랜치: `main` 혹은 `master` 로 불리며, 프로젝트의 주요 버전입니다.

Branch 따는 방법:

```
git checkout -b 새로운브랜치이름
```

- `checkout -b` 명령어는 새로운 브랜치를 만들고 그 브랜치로 이동하는 역할을 합니다.

7. Branch 간 이동

- 브랜치를 전환할 때는 다음 명령어를 사용합니다:

```
git checkout 브랜치이름
```

2. 브랜치 목록 조회

```
git branch
```

현재 로컬에 존재하는 모든 브랜치를 보여줍니다. 현재 체크아웃된 브랜치는 `*`로 표시됩니다.

```
git branch -r
```

원격 저장소의 브랜치 목록을 조회합니다.

```
git branch -a
```

로컬과 원격의 모든 브랜치를 조회합니다.

예시:

```
git branch      # 로컬 브랜치 목록 조회
git branch -r   # 원격 브랜치 목록 조회
git branch -a   # 모든 브랜치 목록 조회
```

8. 브랜치 삭제

```
git branch -d 브랜치이름
```

병합된 브랜치를 삭제할 때 사용합니다. 만약 병합되지 않은 브랜치를 삭제하려고 하면 오류가 발생합니다.

```
git branch -D 브랜치이름
```

병합되지 않은 브랜치도 강제로 삭제할 때 사용합니다.

```
git push origin --delete 브랜치이름
```

원격 저장소의 브랜치를 삭제할 때 사용합니다.

9. 변경 사항 커밋하고 푸시하기

1. 변경 사항을 스테이징:

```
git add .
```

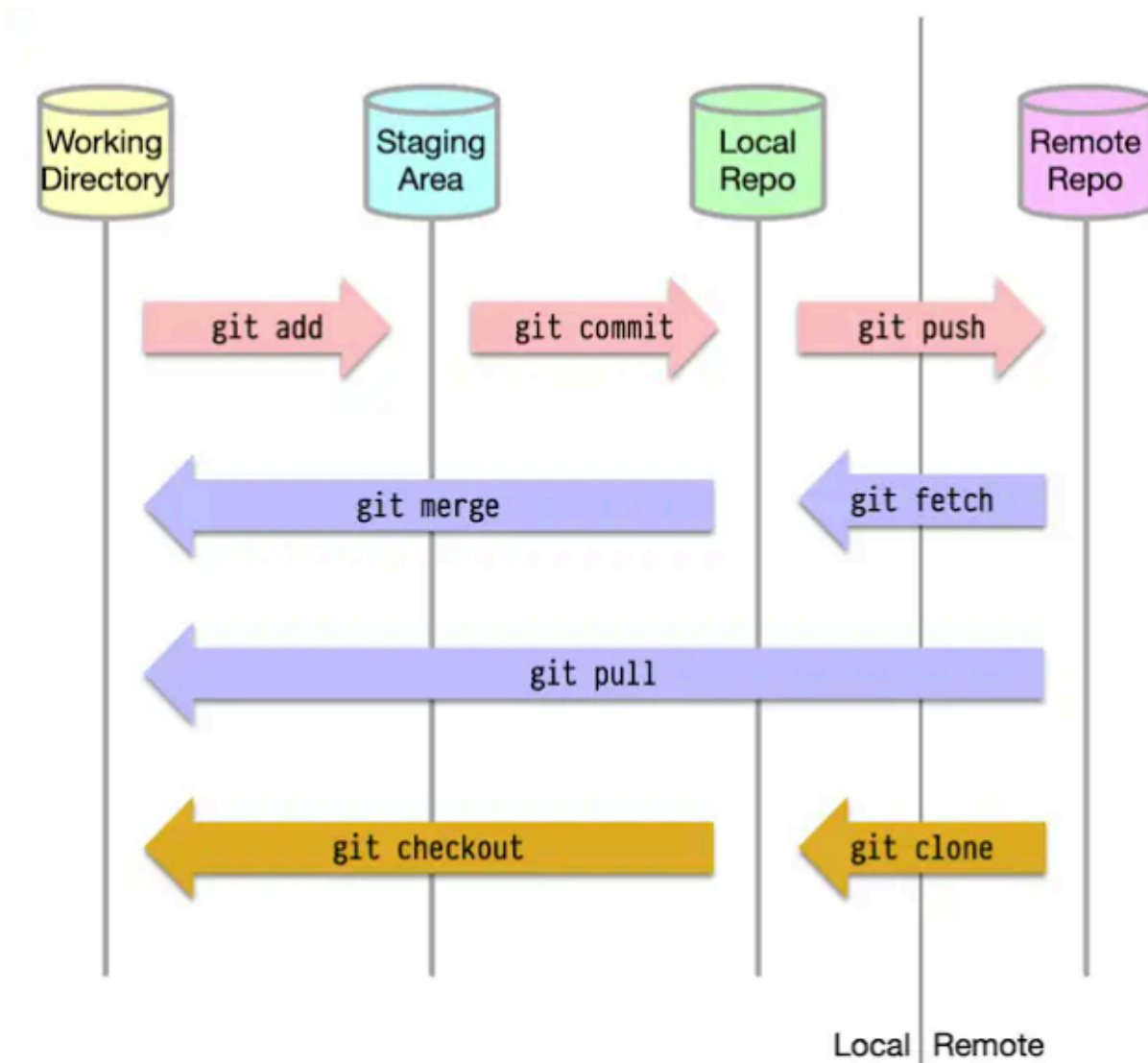
2. 커밋하기:

```
git commit -m "커밋 메시지"
```

3. GitHub로 푸시:

```
git push origin 브랜치이름
```

10. 아키텍처



11. Pull Request (PR)

- Pull Request는 자신이 작업한 브랜치를 `main` 브랜치에 병합 요청하는 기능입니다.

1. GitHub 웹사이트에서 Pull Request 생성.
2. 코드 리뷰가 완료되면 병합(Merge).

12. 기타 기본 명령어

- 상태 확인:

```
git status
```

- 기록 확인:

```
git log
```

GIT 협업

1. 저장소로 이동

- GitHub에 로그인한 후, 협업하려는 저장소로 이동합니다.

2. Settings(설정) 선택

- 저장소 상단 메뉴에서 **Settings** 탭을 클릭합니다.

3. Collaborators and Teams 선택

- 왼쪽 사이드바에서 **Collaborators and Teams**를 선택합니다.

4. Collaborators(협업자) 추가

- 페이지 중간 부분에 있는 **Add people** 버튼을 클릭합니다.
- 협업자로 추가하려는 사용자의 GitHub 사용자명을 입력합니다.

5. 권한 선택

사용자를 추가할 때 권한을 선택할 수 있습니다:

- **Read**: 저장소를 읽을 수 있지만, 수정은 불가능.
- **Write**: 저장소를 읽고 쓸 수 있으며, 커밋하고 푸시할 수 있음.
- **Admin**: 저장소의 모든 권한을 가짐, 설정 변경 가능.

6. 초대 보내기

- 사용자명을 입력한 후, **Add** 버튼을 클릭하여 초대를 보냅니다.
- 초대받은 사용자는 GitHub 알림에서 초대를 수락하면, 해당 저장소에 대한 권한을 부여받게 됩니다.