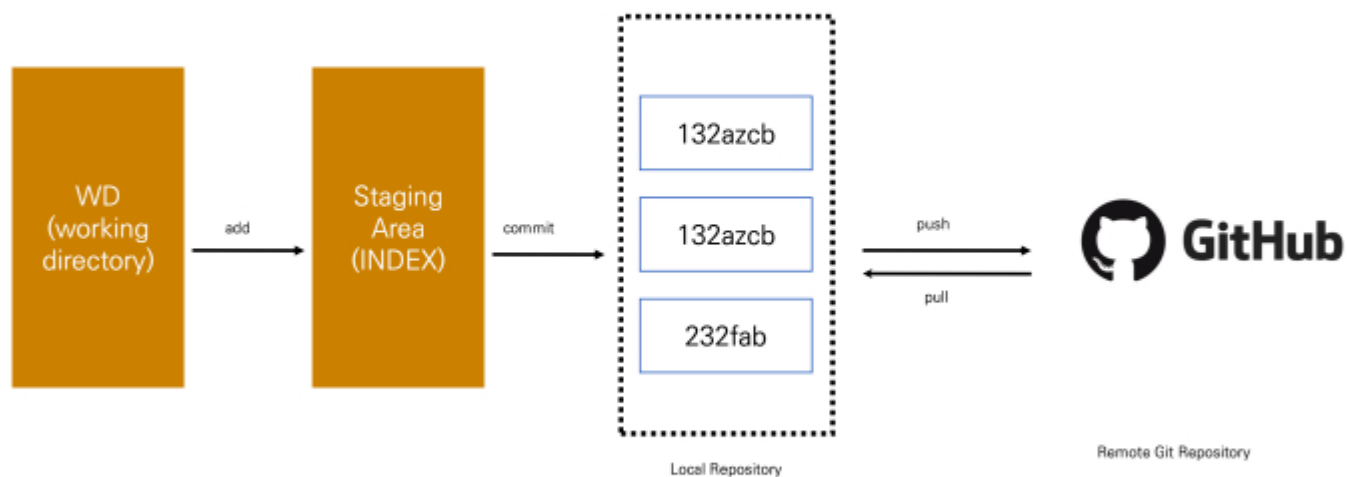


Branch의 목적을 알고 활용할 수 있다.

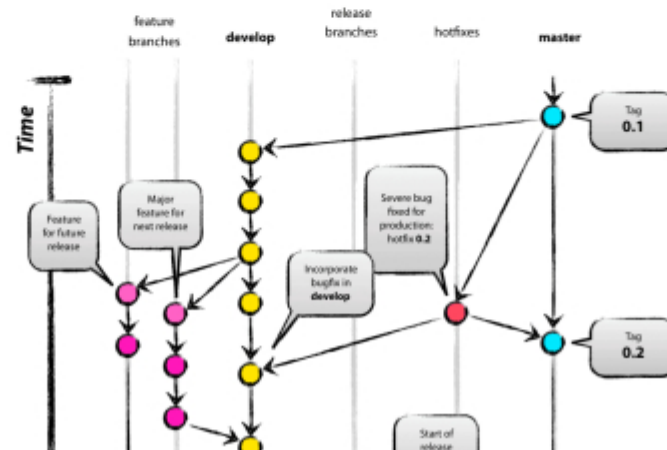
Branch 병합 과정의 상황을 이해하고 충돌을 해결할 수 있다.

## Git (이미 알고 있어야하는 개념 정리)



## branch

- 독립적인 작업흐름을 만들고 관리



## 브랜치 주요 명령어

### 1. 브랜치 생성

```
(master) $ git branch {branch name}
```

### 2. 브랜치 이동

```
(master) $ git checkout {branch name}
```

### 3. 브랜치 생성 및 이동

```
(master) $ git checkout -b {branch name}
```

### 4. 브랜치 목록

```
(master) $ git branch
```

### 5. 브랜치 삭제

```
(master) $ git branch -d {branch name}
```

## merge

- 각 branch에서 작업을 한 이후 이력을 합치기 위해 merge 명령어를 사용
- 병합을 진행할 때, 서로 다른 이력(commit)에서
  - 동일한 파일을 수정한 경우 충돌이 발생
    - 이 경우에는 반드시 직접 해당 파일을 확인하고 적절하게 수정
    - 수정한 이후에 직접 커밋 실행
  - 다른 파일을 수정한 경우
    - 충돌 없이 자동으로 Merge Commit이 생성됨

