



버전 관리 도구



- 졸업논문.hwp
- 졸업논문수정.hwp
- 졸업논문수정1.hwp
- 졸업논문수정2.hwp
- 졸업논문완성본.hwp
- 졸업논문완성본1.hwp
- 졸업논문완성본2.hwp
- 졸업논문최종완성본.hwp
- 졸업논문최종완성본1.hwp
- 졸업논문최종완성본2.hwp
- 졸업논문최종완성본final.hwp
- 졸업논문최종완성본final1.hwp
- 졸업논문최종완성본final2.hwp
- 졸업논문최종완성본final최종.hwp
- 졸업논문최종완성본final최종1.hwp
- 졸업논문최종완성본final최종2.hwp
- ㅠ유서.hwp

코드 저장소



깃허브 왜 배워야 하죠 ?

1. 원활한 협업

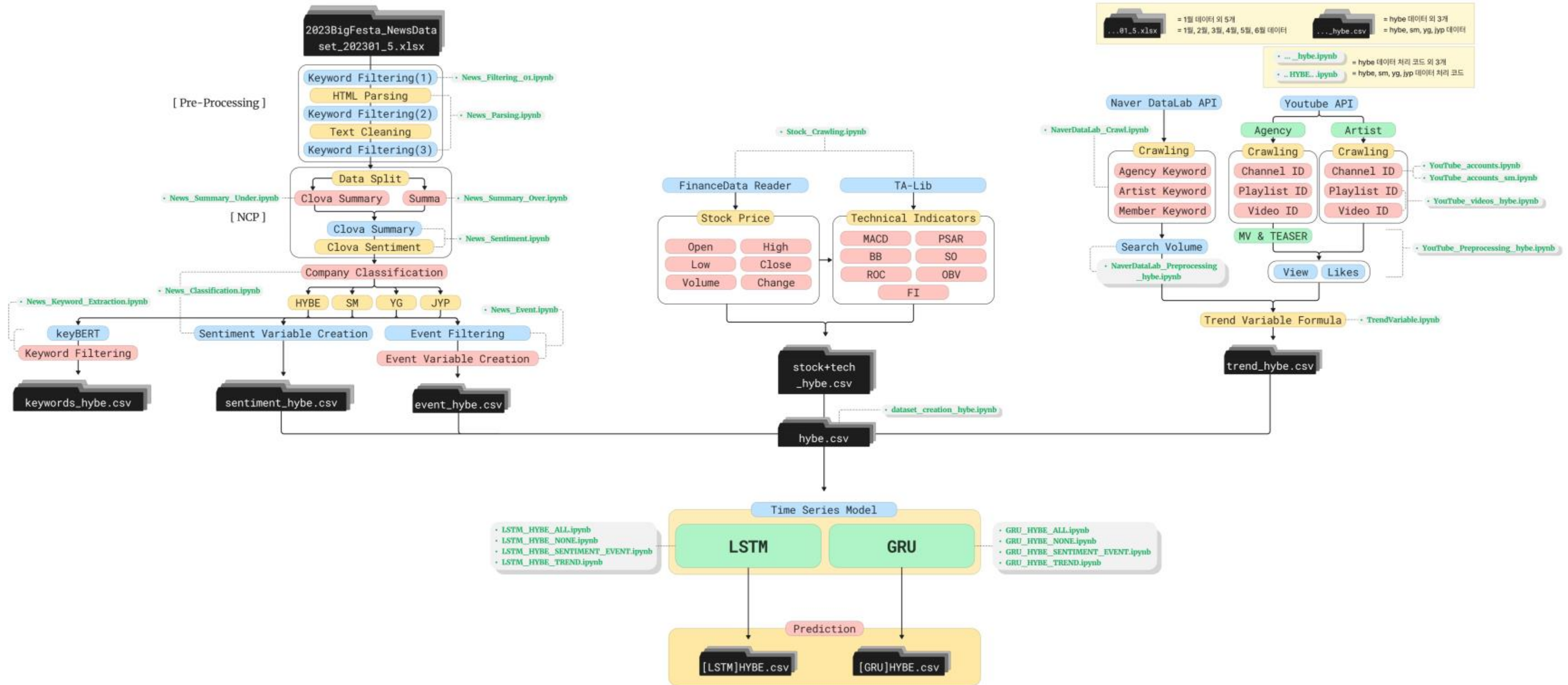
2. 포트폴리오

3. 간지

```
├── YouTube
│   ├── channel+pl_sm.csv
│   ├── channel_sm.csv
│   ├── data_aespa.csv
│   ├── data_boa.csv
│   ├── data_exo.csv
│   ├── data_nct.csv
│   ├── data_shinee.csv
│   ├── data_sm.csv
│   ├── item_aespa.csv
│   ├── item_boa.csv
│   ├── item_exo.csv
│   ├── item_nct.csv
│   ├── item_shinee.csv
│   ├── item_sm.csv
│   ├── video_aespa.csv
│   ├── video_boa.csv
│   ├── video_exo.csv
│   ├── video_nct.csv
│   ├── video_shinee.csv
│   └── video_sm.csv
├── mirae
│   ├── 2023BigFesta_NewsDataset_202301_5.xlsx
│   ├── 2023BigFesta_NewsDataset_202302_5.xlsx
│   ├── 2023BigFesta_NewsDataset_202303_5.xlsx
│   ├── 2023BigFesta_NewsDataset_202304_5.xlsx
│   ├── 2023BigFesta_NewsDataset_202305_5.xlsx
│   ├── 2023BigFesta_NewsDataset_202306_5.xlsx
│   └── mkt_idx.csv
└── result
```

```
32 directories, 152 files
(base) yusolcho@joyusol-ui-MacBookPro ~ %
```


Entertainment Stock Price Prediction Analysis Map



깃허브 왜 배워야 하죠 ?

1. 원활한 협업

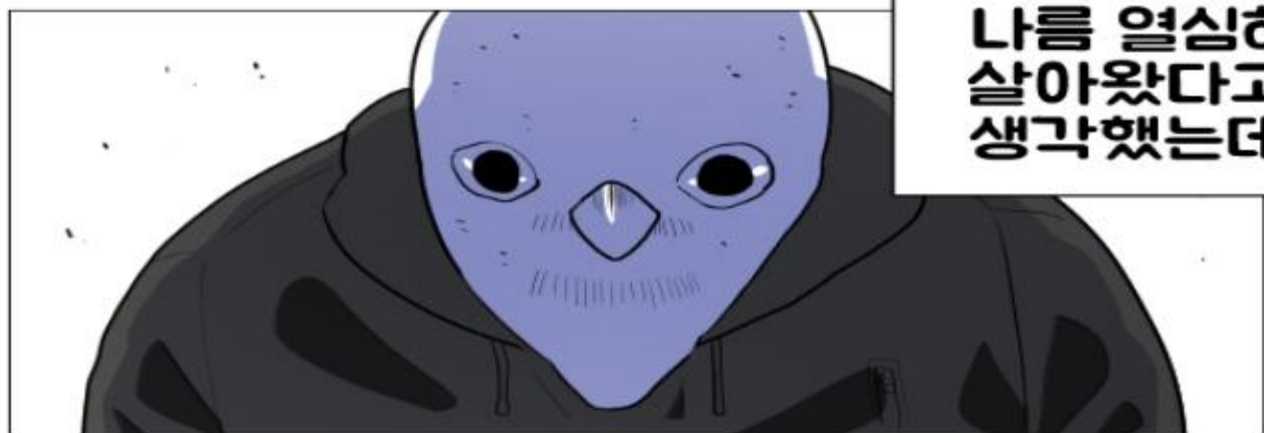
2. 포트폴리오

3. 간지

포트폴리오는
위험하다



나를 열심히
살아왔다고
생각했는데



포폴을 적다보면
난 무엇일까...
라고 생각이 든다



나....
진짜 뭐지.....



포트폴리오는
위험하다



2018-DataSciencePortfolio Public

Watch 0

Fork 0

Star 1

master

1 Branch 0 Tags

Go to file

Add file

Code

About

R, python, SPSS

Readme

Activity

1 star

0 watching

0 forks

Report repository

Releases

No releases published

Packages

No packages published

Languages



limsooyeon Delete SideProject directory

d8ccc63 · 2 weeks ago 93 Commits

MovieLens의 메타데이터를 이용한 추천 시스템	Add files via upload	6 years ago
bakery transactions 데이터를 이용한 장바구니 ...	Delete README.md	5 years ago
네이버 영화 리뷰 감성분석	Add files via upload	6 years ago
딥러닝 기반 시간 단위 미세면지 예보	Add files via upload	6 years ago
범죄 발생 건수에 대한 회귀분석 및 추정	Add files via upload	6 years ago
서울특별시 공공자전거 수요예측과 편중 지역 ...	Rename RnnLstm.ipynb to 06_RnnLstm.ipynb	5 years ago
스타벅스 크리스마스 프로모션 분석을 통한 마...	Add files via upload	6 years ago
영화 추천시스템	Add files via upload	6 years ago
유기견 매칭 시스템	Add files via upload	5 years ago
자살률 현황 분석 및 대안 제시	Delete code.R	5 years ago
전국 대학생의 아르바이트 근로 현황 실태 조사...	Add files via upload	6 years ago
README.md	Update README.md	2 weeks ago

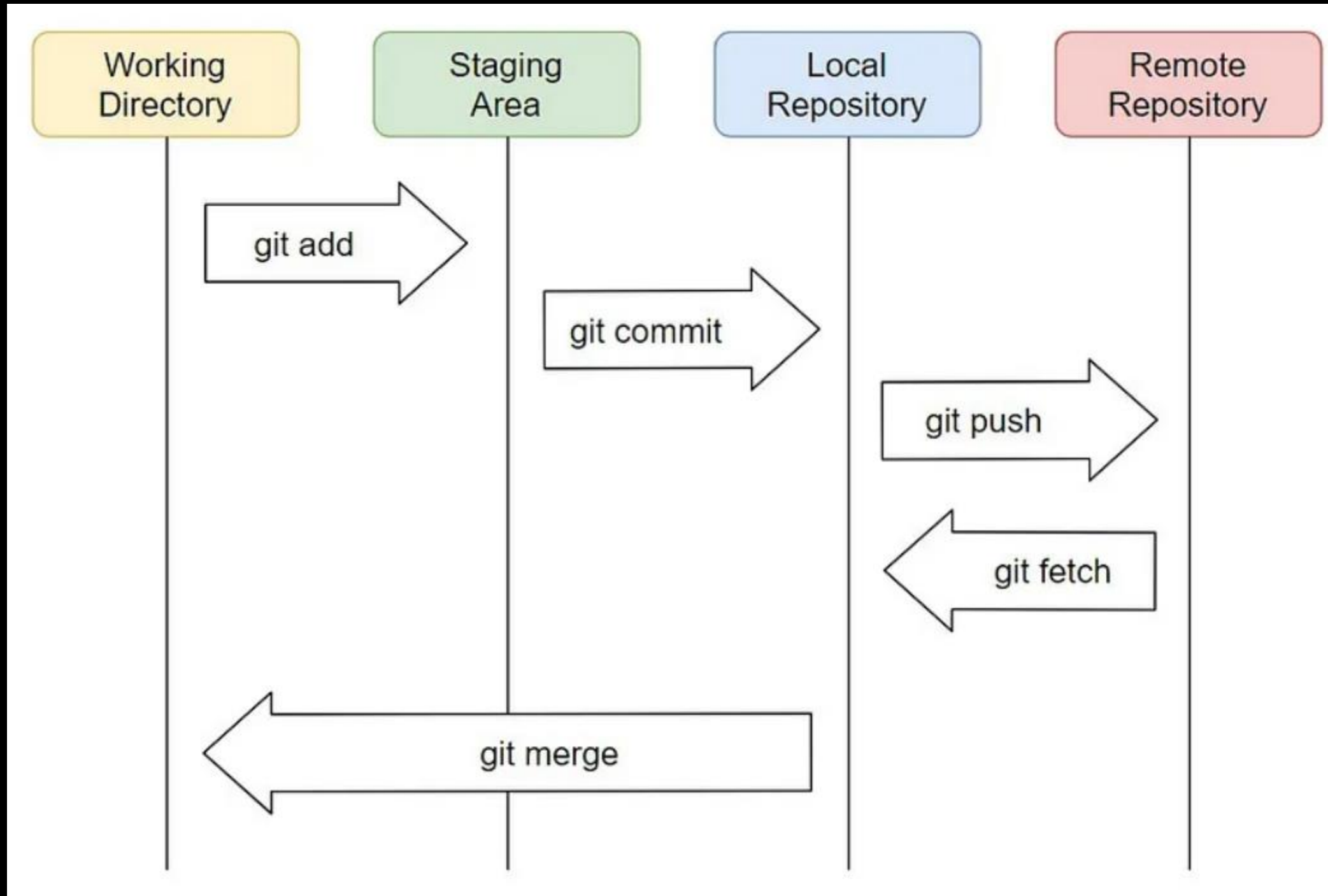
깃허브 왜 배워야 하죠 ?

1. 원활한 협업
2. 포트폴리오
3. 간지





git의 용어 및 동작 방식



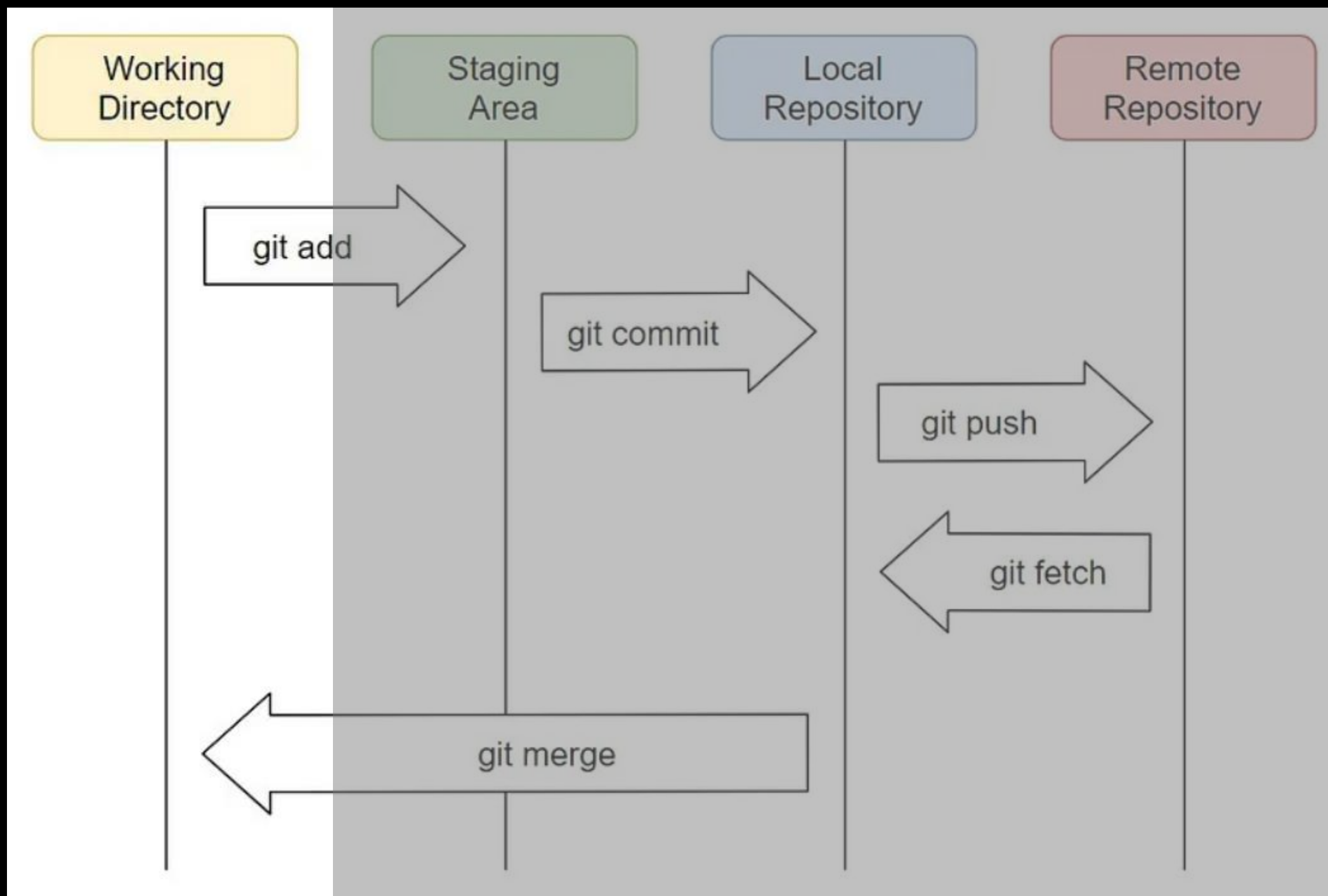
working Directory :
작업 폴더

Staging Area :
스테이징

Local Repository :
로컬 저장소

Remote Repository :
원격 저장소 (깃허브)

git의 용어 및 동작 방식



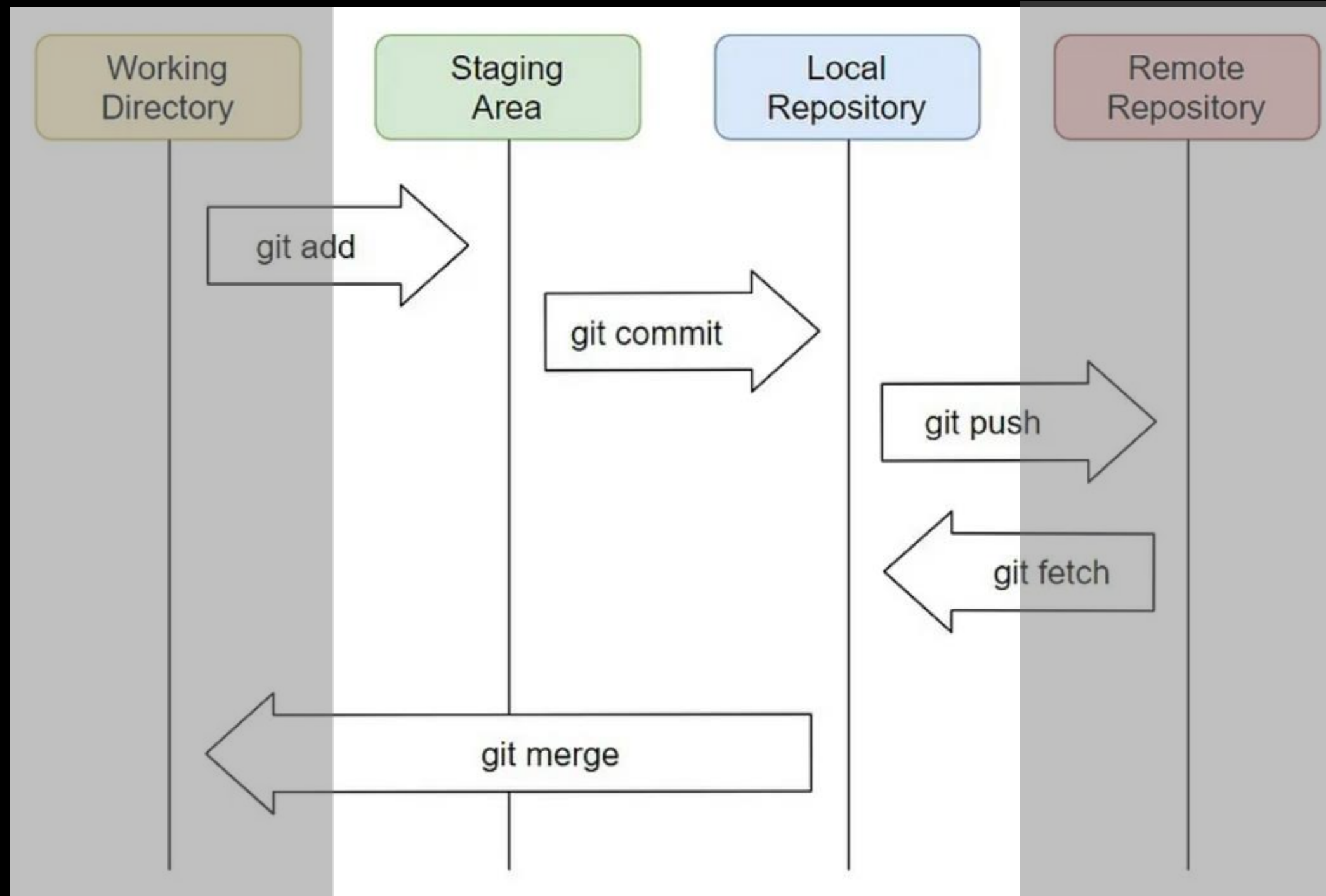
working Directory :
작업 폴더

Staging Area :
스테이징

Local Repository :
로컬 저장소

Remote Repository :
원격 저장소 (깃허브)

git의 용어 및 동작 방식



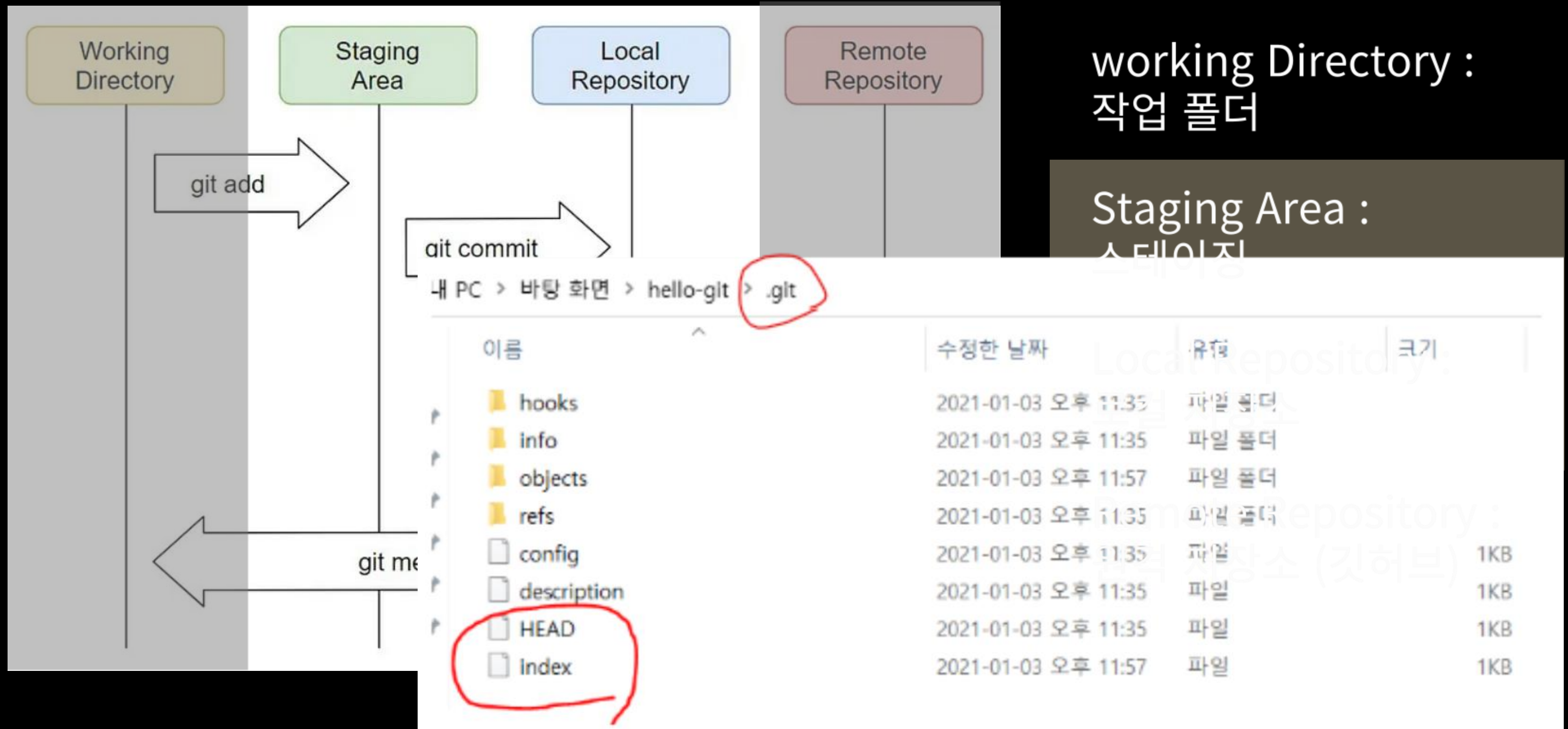
working Directory :
작업 폴더

Staging Area :
스테이징

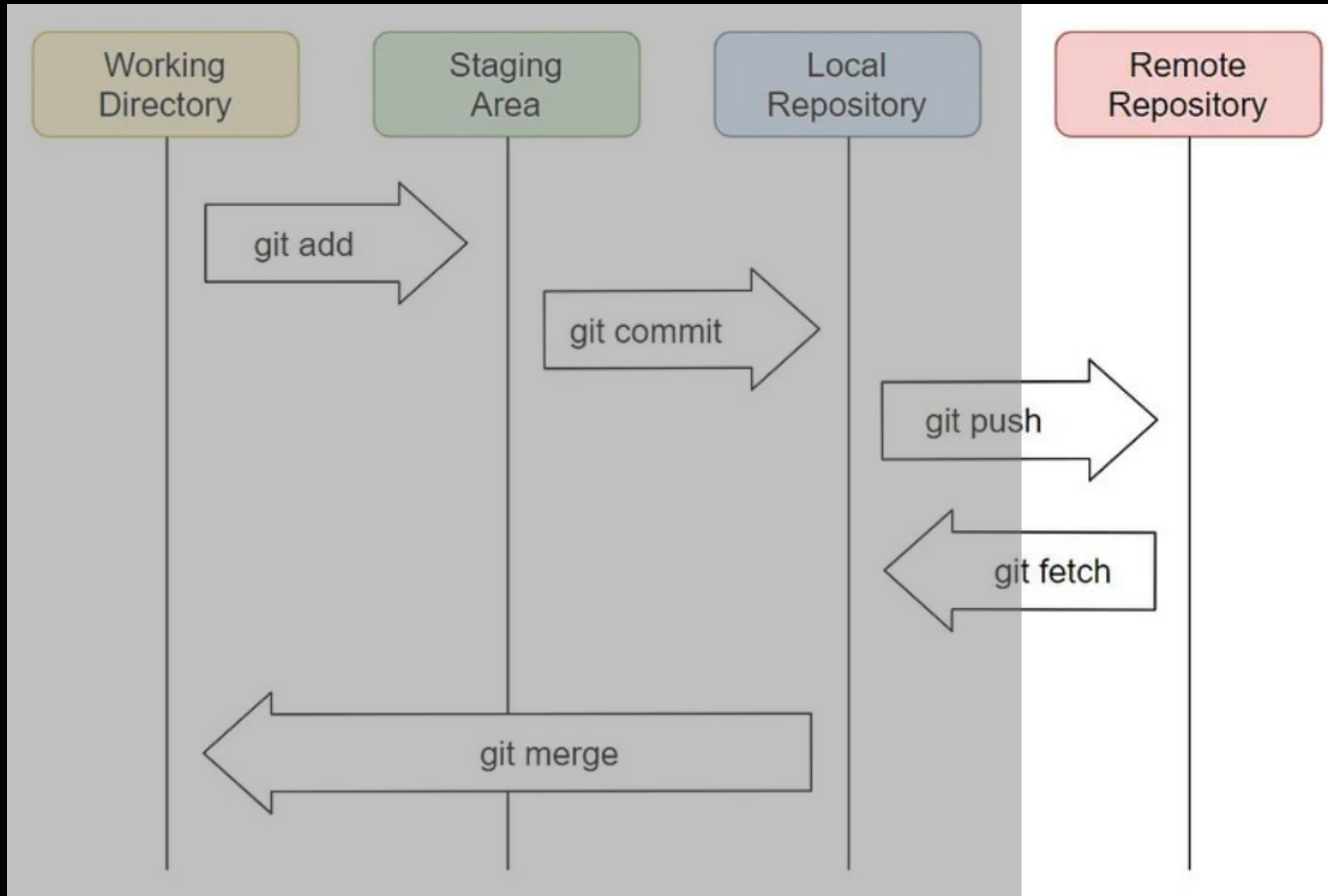
Local Repository :
로컬 저장소

Remote Repository :
원격 저장소 (깃허브)

git의 용어 및 동작 방식



git의 용어 및 동작 방식



working Directory :
작업 폴더

Staging Area :
스테이징

Local Repository :
로컬 저장소

Remote Repository :
원격 저장소 (깃허브)

git의 용어 및 동작 방식

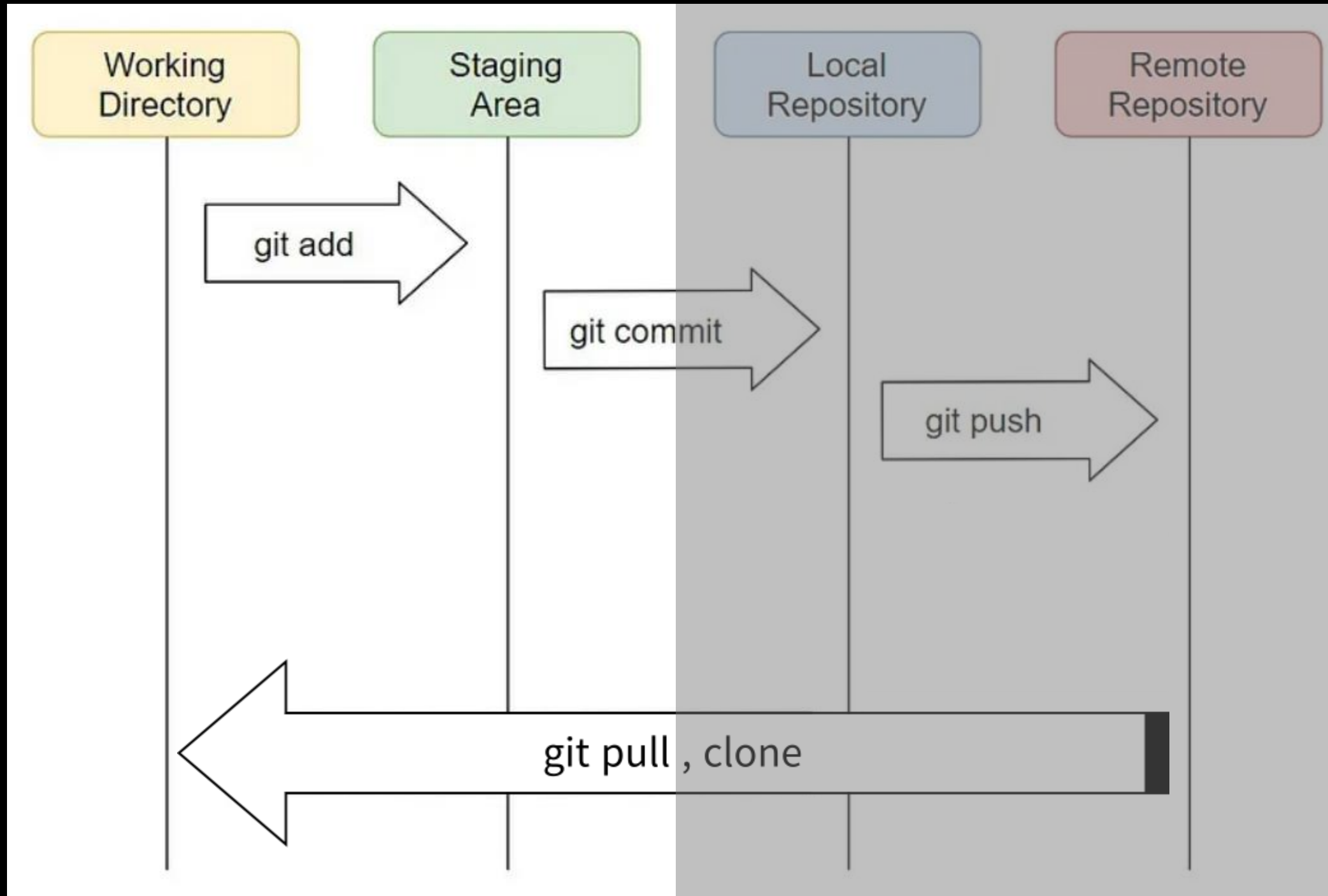
working Directory : 나의 pc에서 **작업자의 현재 시점**

Staging Area : 저장소에 커밋하기전에 **커밋을 준비하는 위치**

Local Repository : 내 pc에 파일이 저장되는 **개인 저장소**

Remote Repository : 깃허브

git의 용어 및 동작 방식



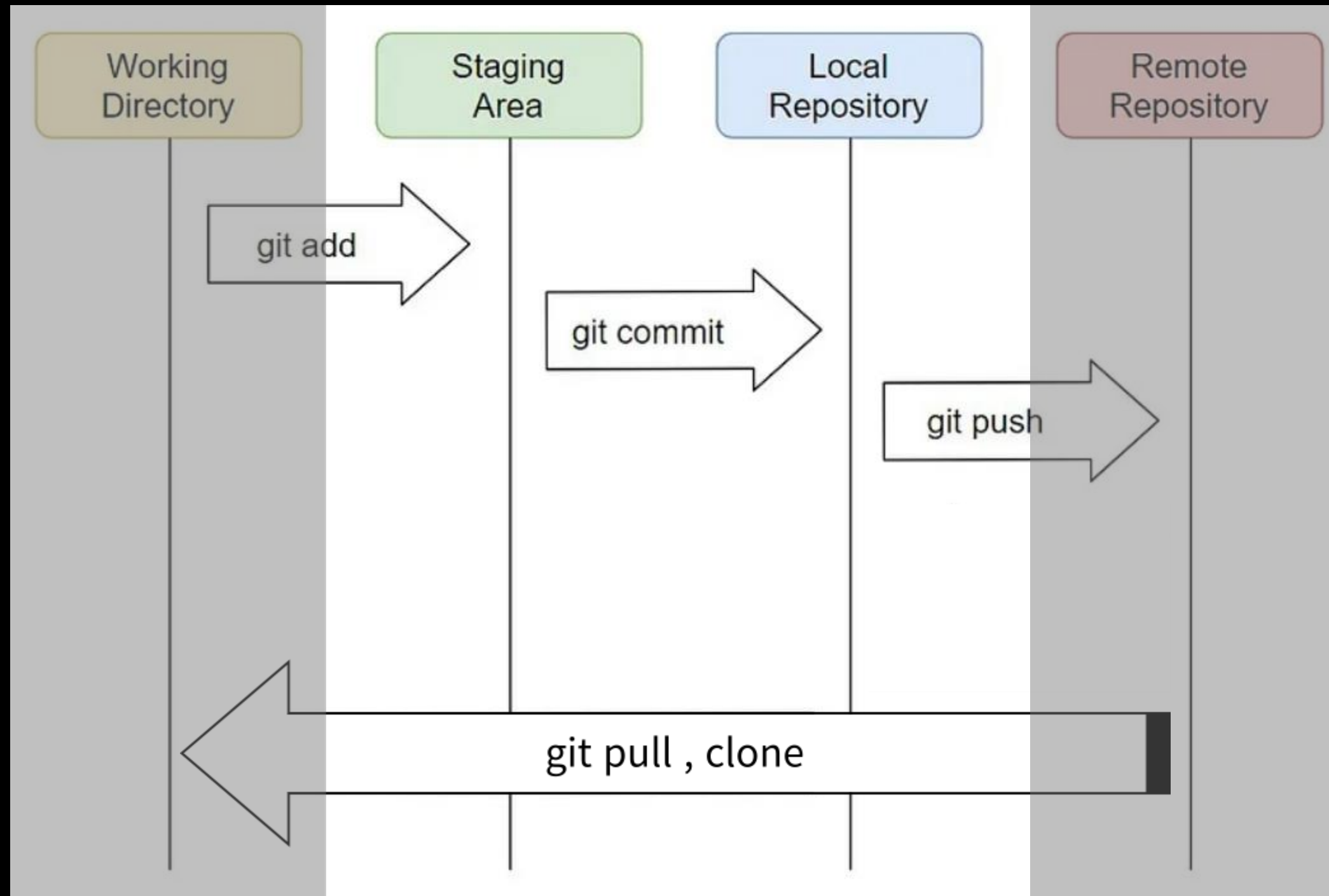
add :
Staging으로 올리기

commit :
이름 붙이기

push :
깃허브에 올리기

pull, clone :
깃허브에서 내려받기

git의 용어 및 동작 방식



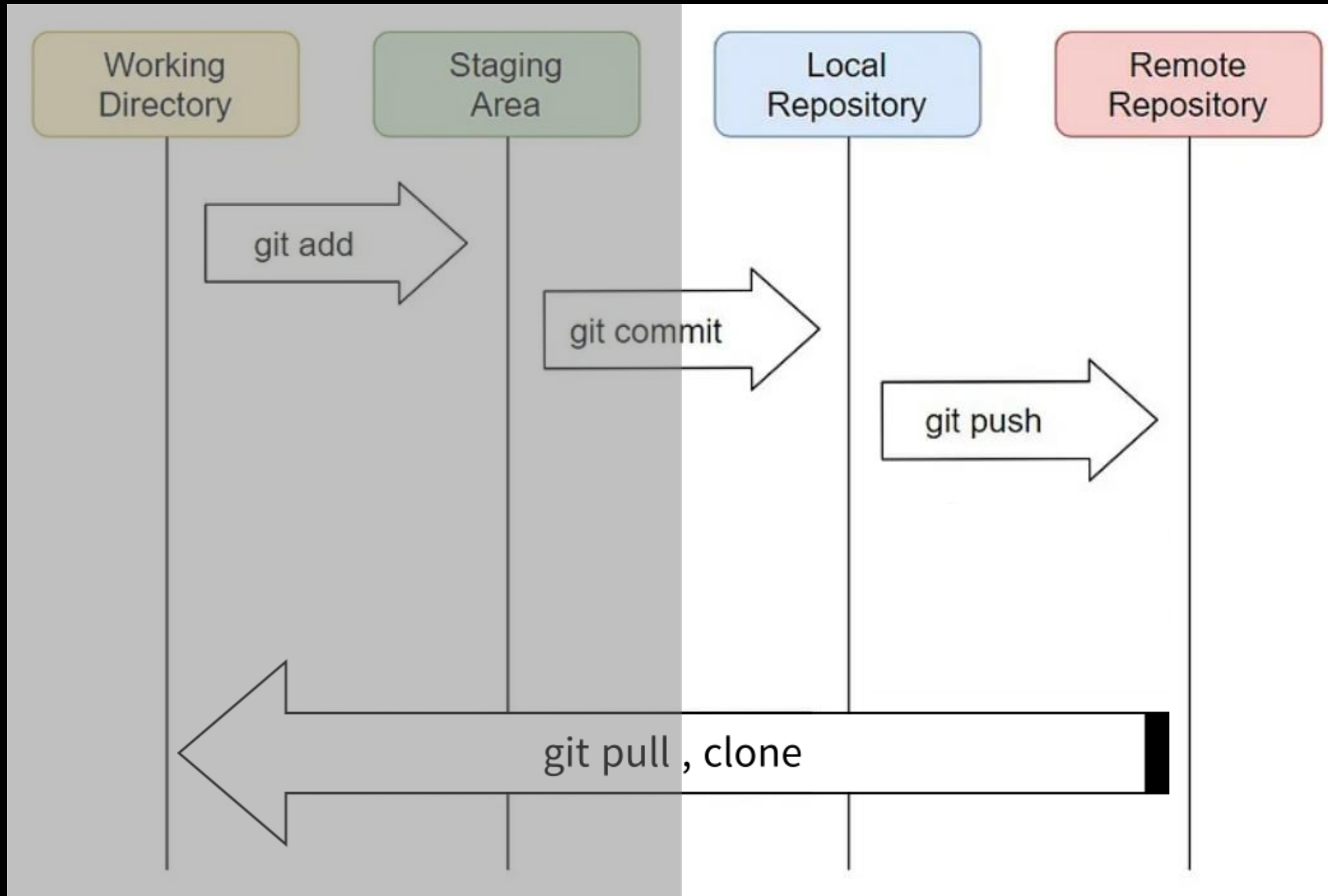
`add` :
Staging으로 올리기

`commit` :
이름 붙이기

`push` :
깃허브에 올리기

`pull, clone` :
깃허브에서 내려받기

git의 용어 및 동작 방식



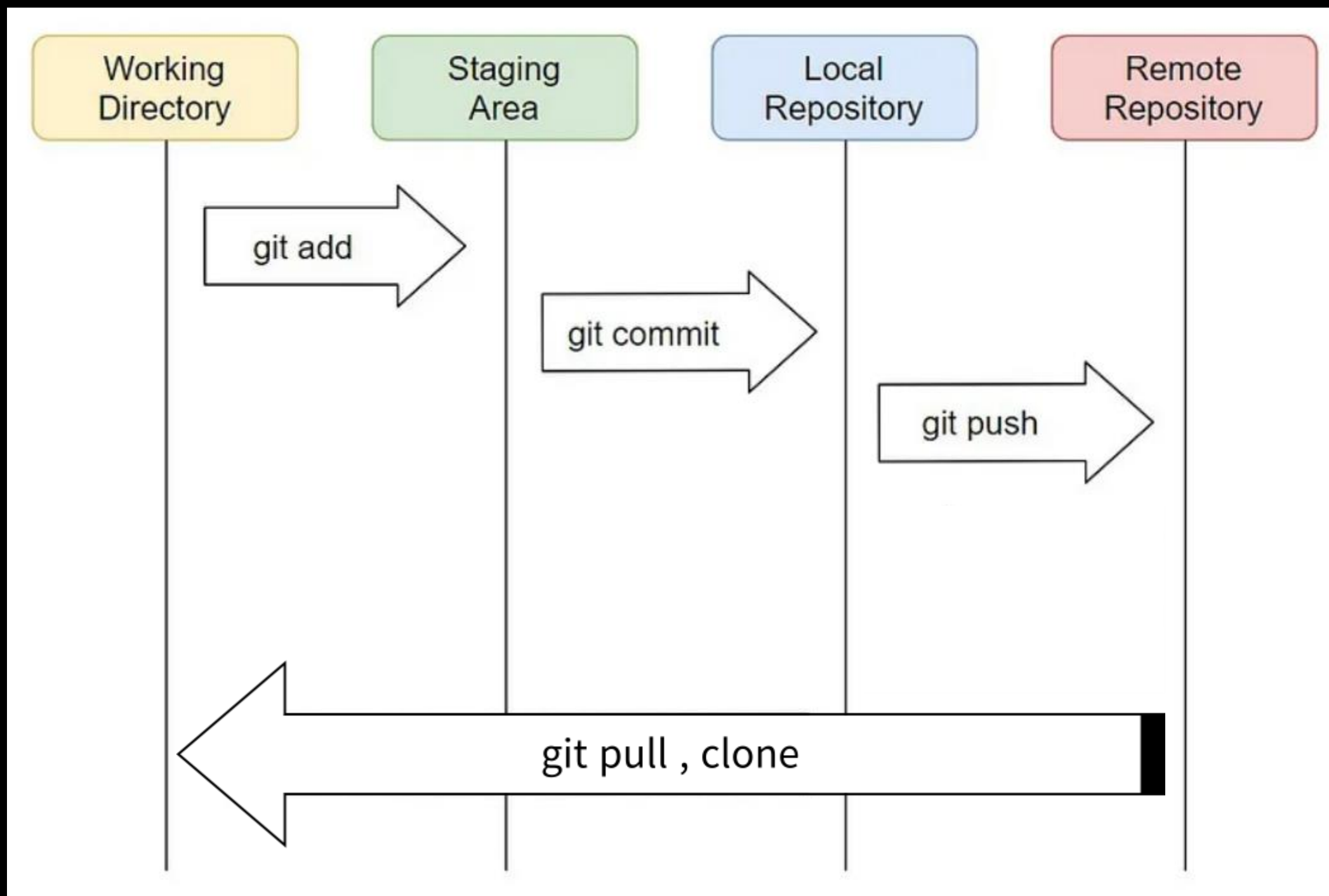
add :
Staging으로 올리기

commit :
이름 붙이기

push :
깃허브에 올리기

pull, clone :
깃허브에서 내려받기

git의 용어 및 동작 방식



add :
Staging으로 올리기

commit :
이름 붙이기

push :
깃허브에 올리기

pull, clone :
깃허브에서 내려받기

git의 용어 및 동작 방식



push

add

commit

나의 소중한 코드

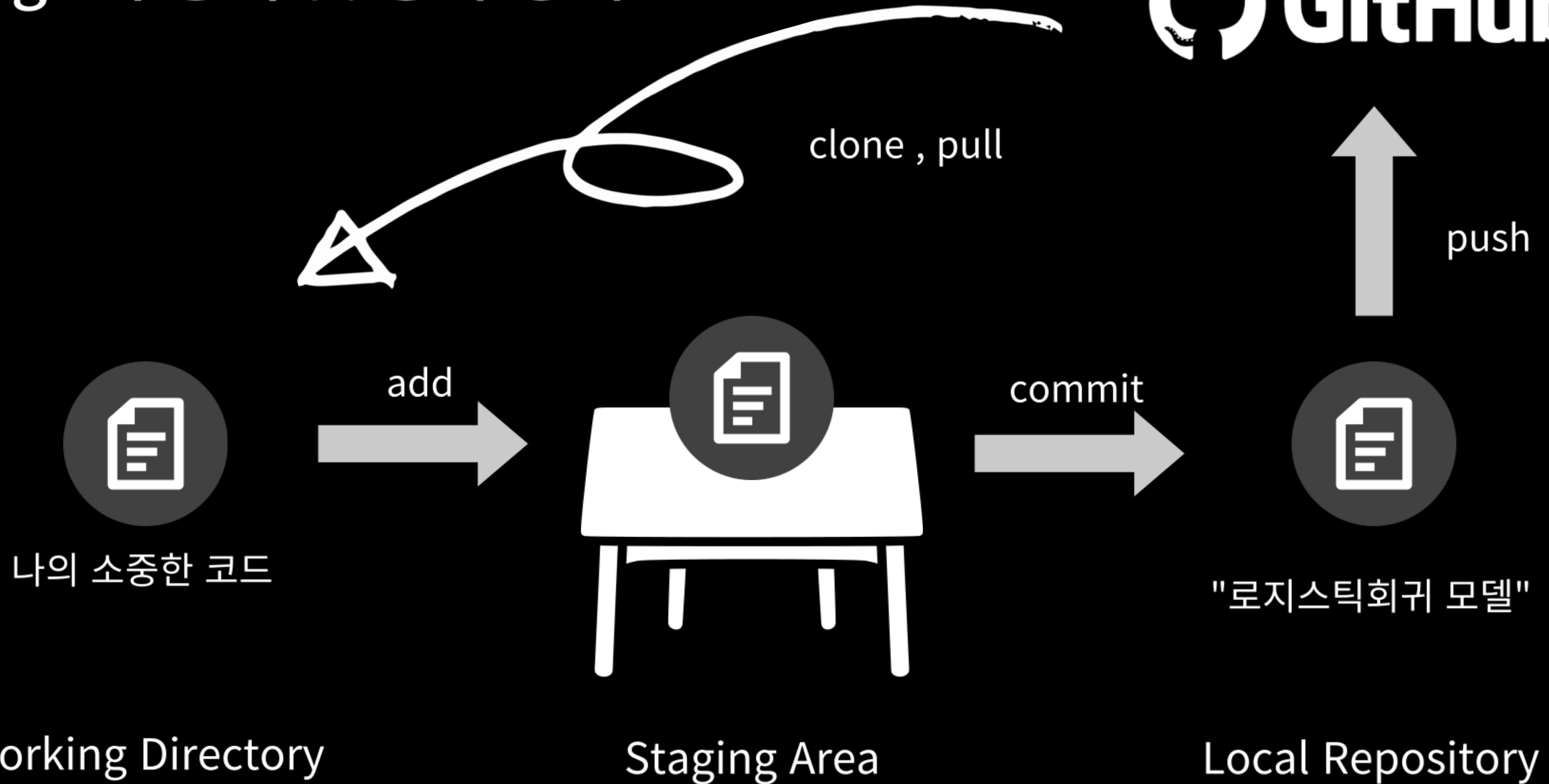
"로지스틱회귀 모델"

Working Directory

Staging Area

Local Repository

git의 용어 및 동작 방식



git의 용어 및 동작 방식

`git add 파일-이름` : '파일'을 staging area에 올림

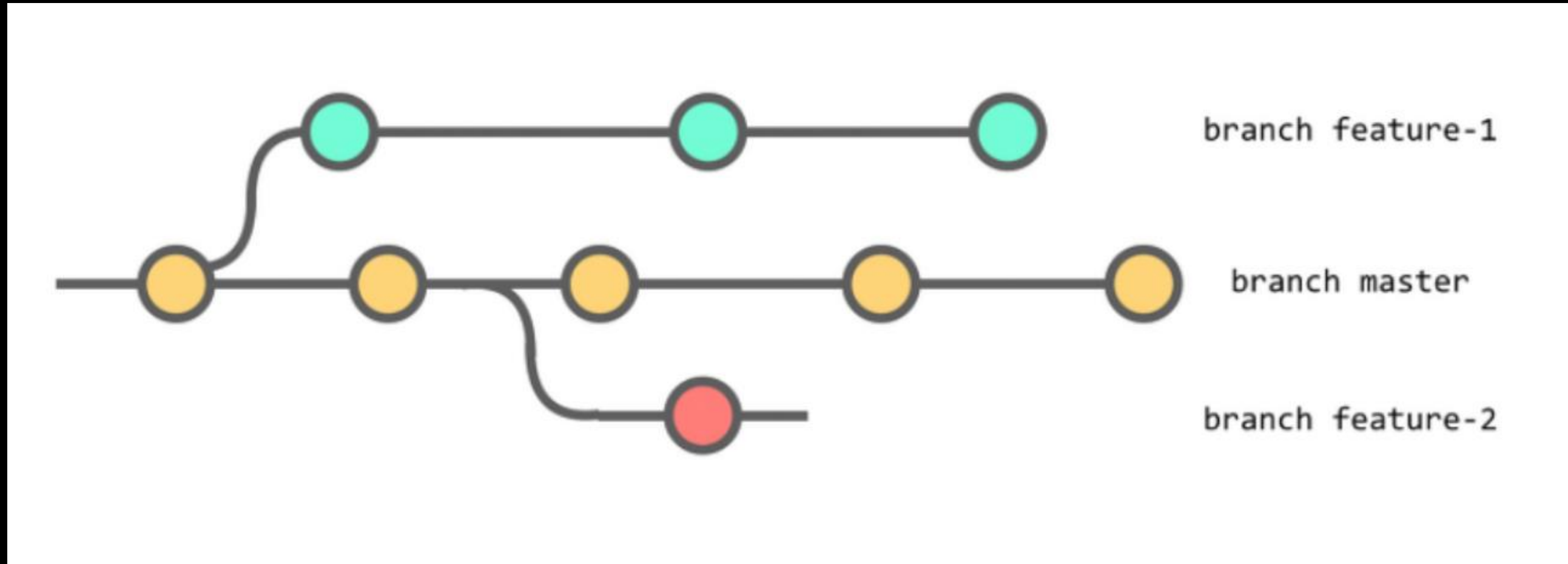
`git commit -m "설명"` : staging area에 올라간 파일에 설명을 붙임

`git push origin branch-이름` : origin은 깃허브를 의미하고 branch이름은 깃허브 브랜치를 의미한다.

`git pull origin branch-이름` : 깃허브의 코드로 내 기존 pc 코드를 업데이트함

`git clone 레포지토리주소` : 레포지토리의 코드를 다운 받음

branch



코드를 변경하고 싶거나 테스트 하고 싶을 때 사용

branch

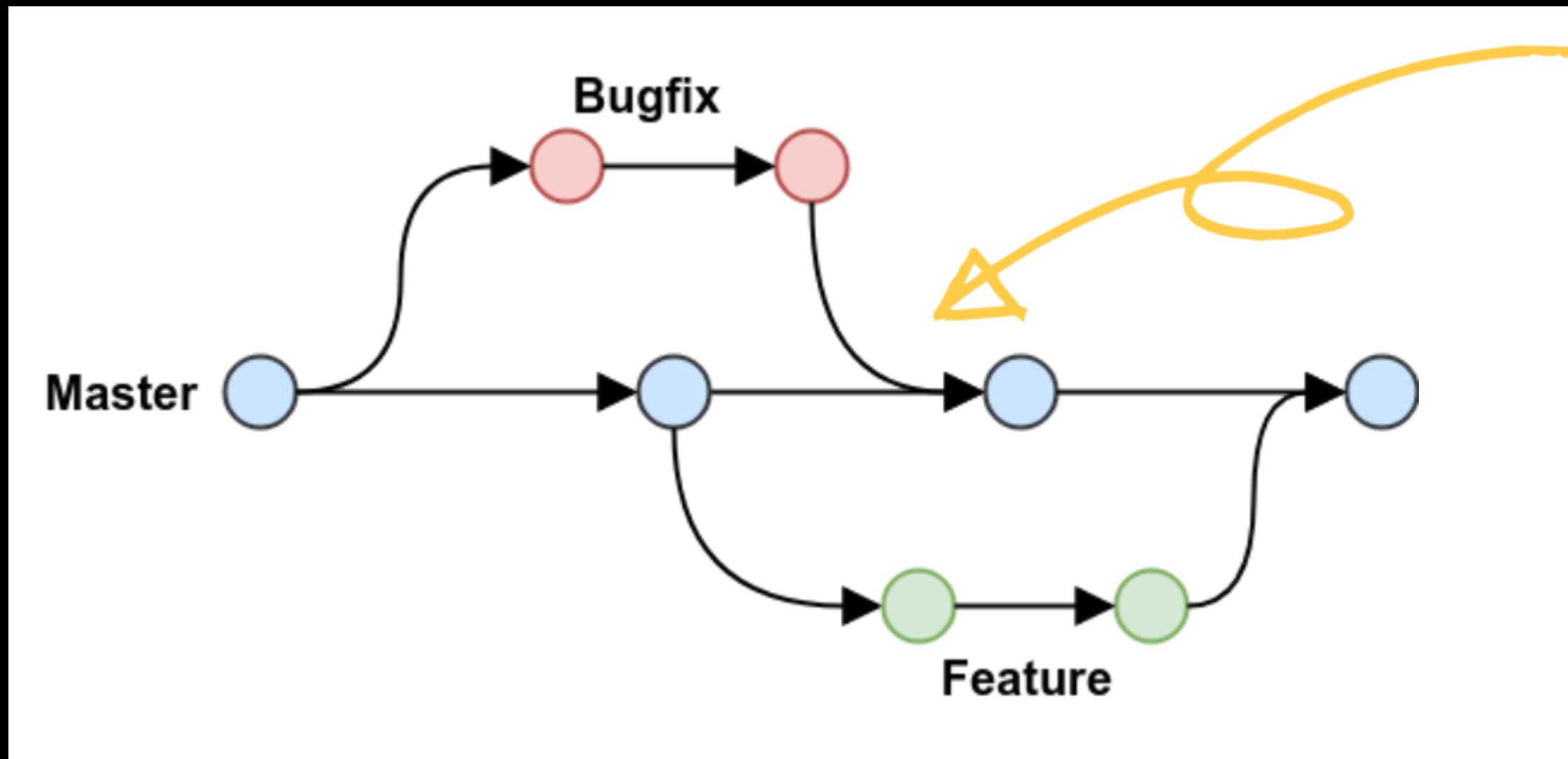
1. branch는 독립적인 작업 공간

2. branch에서 작업이 끝나면 main으로 합쳐준다.

3. branch의 이름은 자유

4. branch는 기존 코드의 복사본

branch



Bugfix branch를
main(master)에 merge

branch

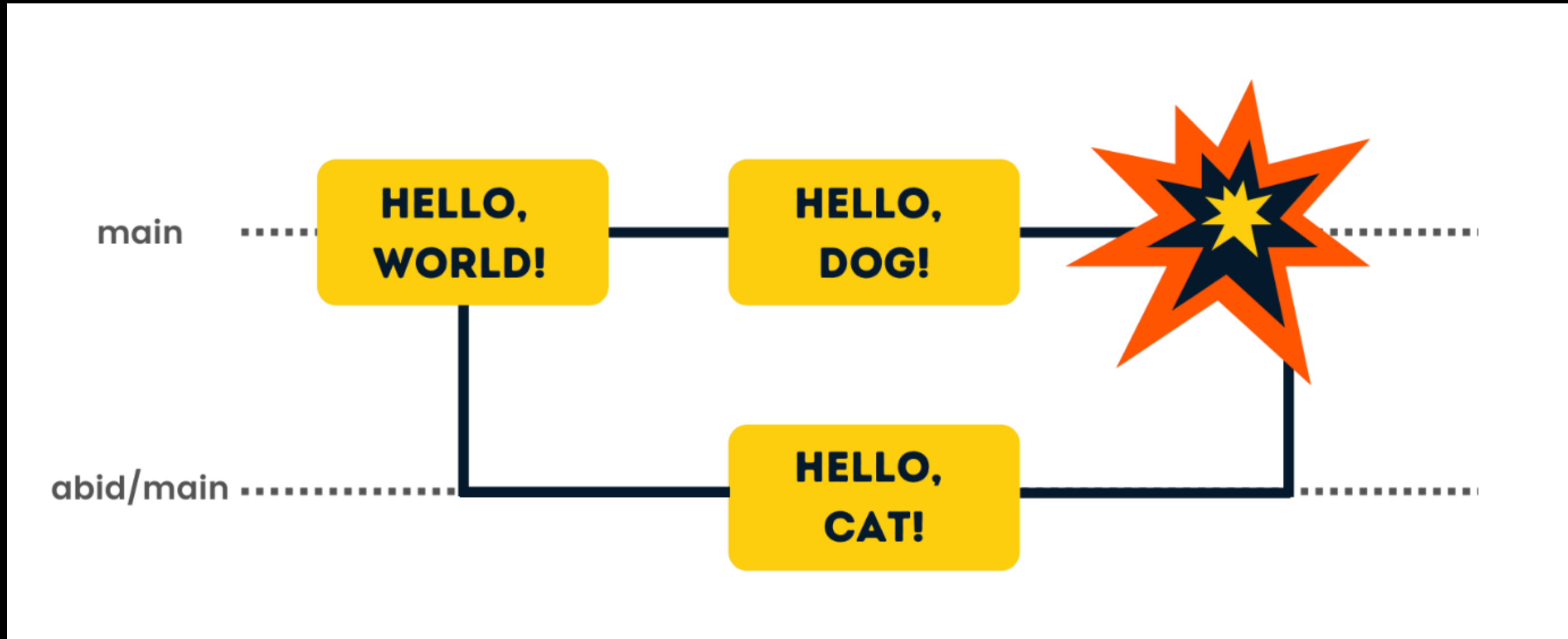
`git branch 브랜치-이름` : 브랜치 생성

`git checkout 브랜치-이름` : 브랜치 이동

`git checkout -b 브랜치-이름` : 브랜치 생성 및 이동

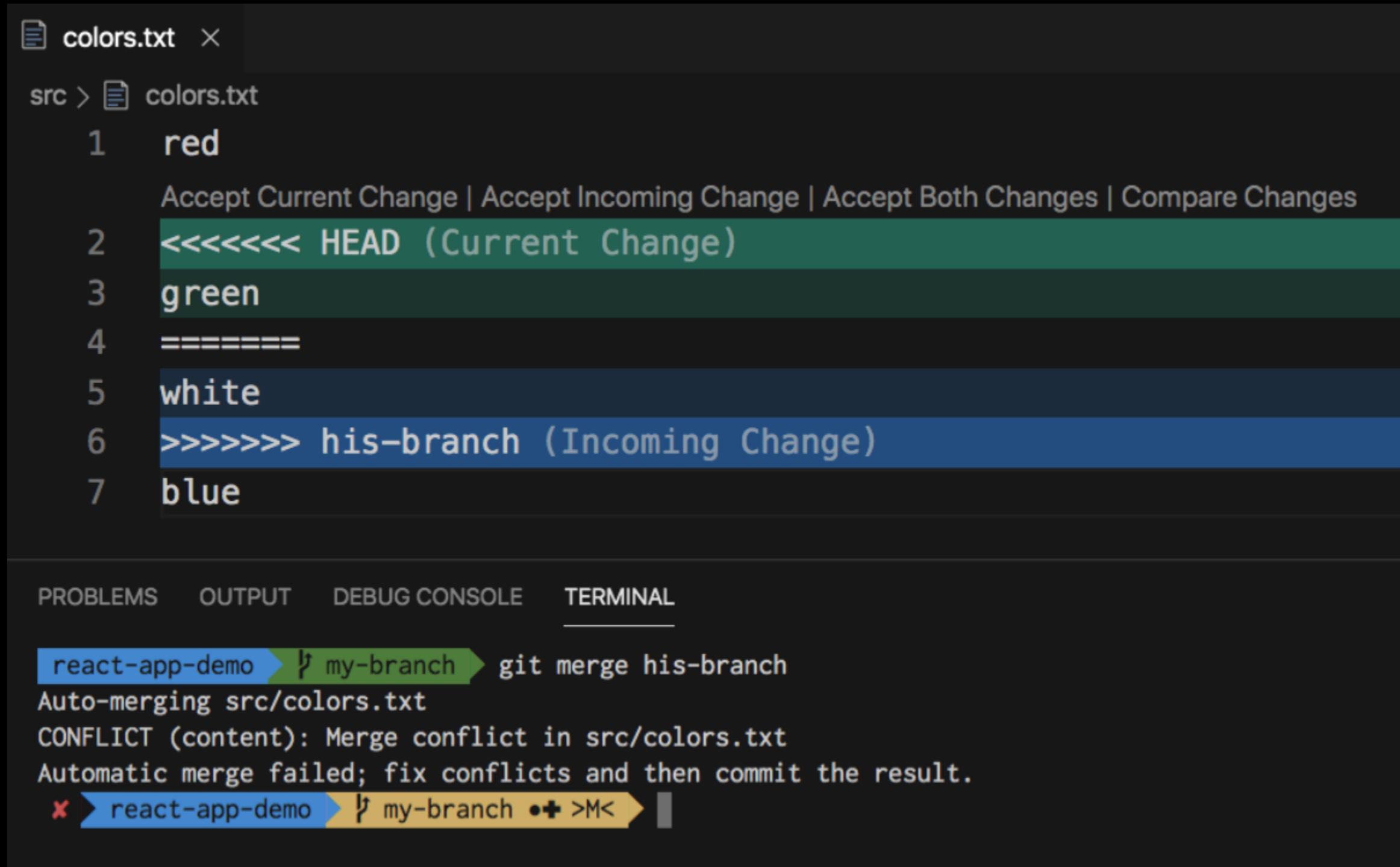
`git merge branch-이름` : branch 를 main에 병합

충돌 (conflict)



코드의 같은 줄에 서로 다른 변경사항이 생겼을 때 발생
코드 입장에서는 어떤 변경사항이 진짜인지 구분할 수 없으므로 사용자가 골라줘야한다.

충돌 (conflict)

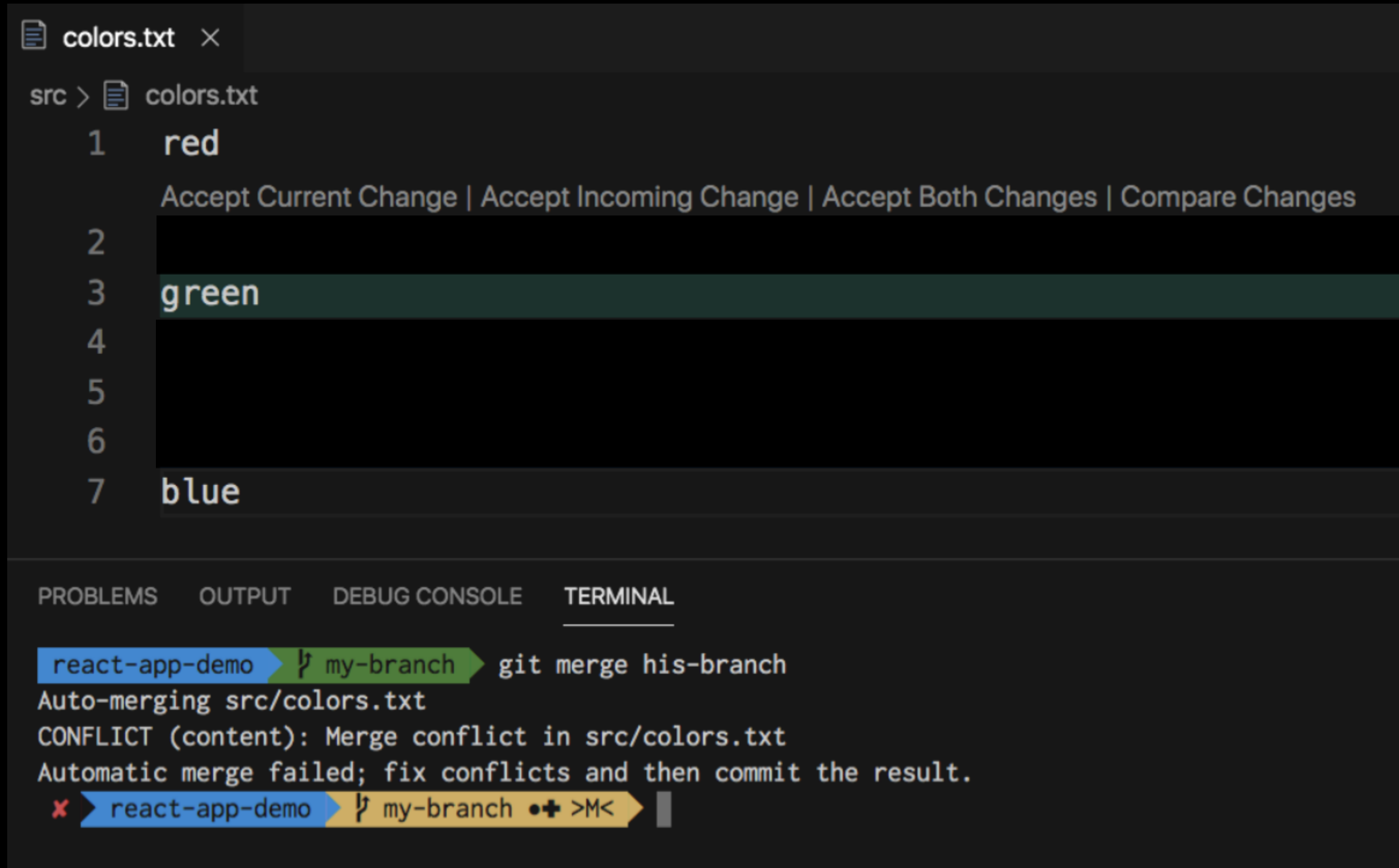


```
colors.txt x
src > colors.txt
1  red
   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
2  <<<<<< HEAD (Current Change)
3  green
4  =====
5  white
6  >>>>>> his-branch (Incoming Change)
7  blue

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
react-app-demo  my-branch  git merge his-branch
Auto-merging src/colors.txt
CONFLICT (content): Merge conflict in src/colors.txt
Automatic merge failed; fix conflicts and then commit the result.
x react-app-demo  my-branch  •+ >M<
```

green 과 white 중에 선택

충돌 (conflict)



```
src > colors.txt
1  red
   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
2
3  green
4
5
6
7  blue
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
react-app-demo my-branch git merge his-branch
Auto-merging src/colors.txt
CONFLICT (content): Merge conflict in src/colors.txt
Automatic merge failed; fix conflicts and then commit the result.
x react-app-demo my-branch >M<
```

green 과 white 중에 선택

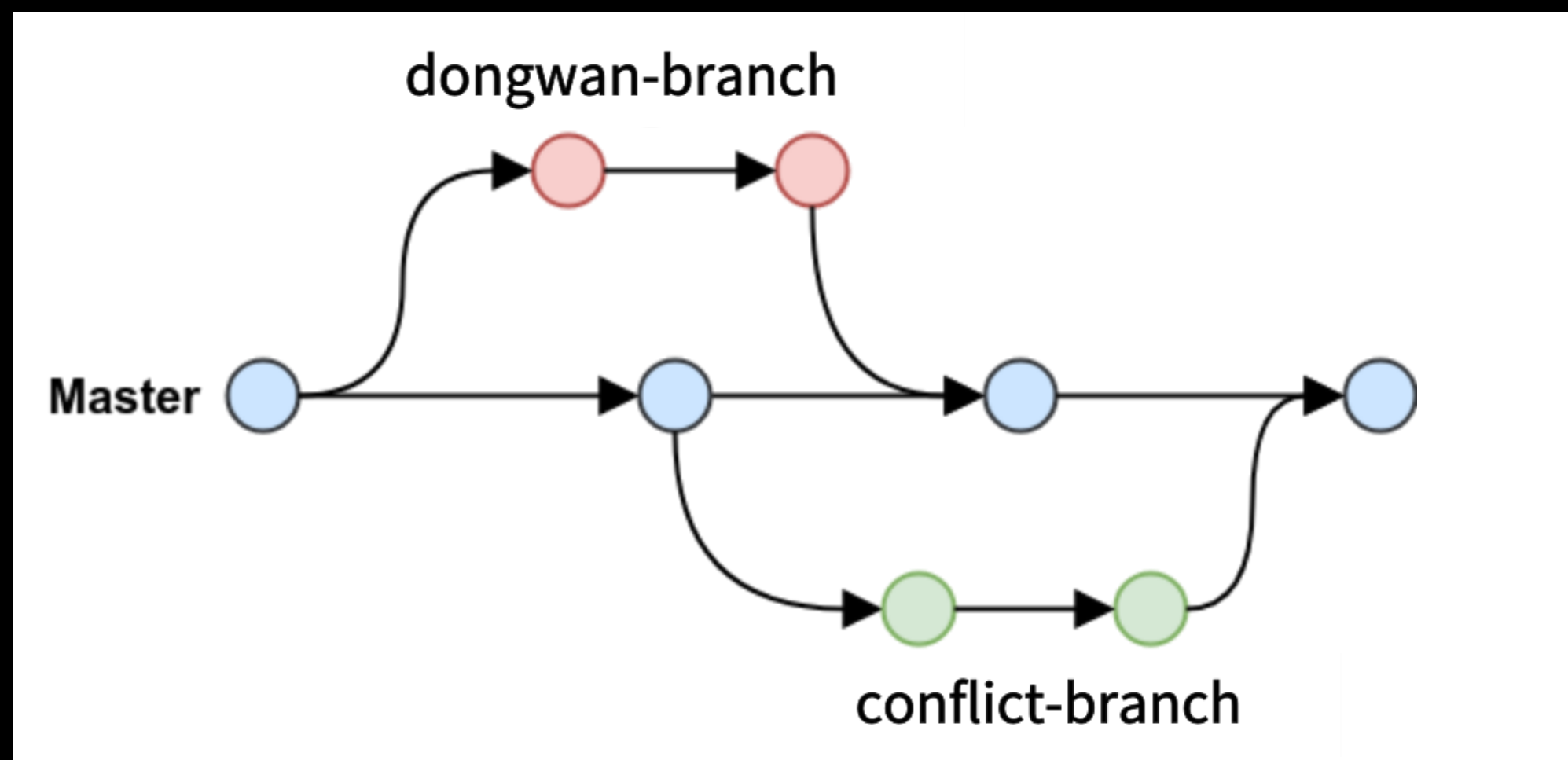
실습 순서

1. 깃허브에 레포지토리 만들기
2. 레포지토리를 내 pc로 clone 받기
3. 나의 코드 올려보기
4. 브랜치를 이용하기
5. 충돌 해결하기



실습 순서

1. 깃허브에 레포지토리 만들기
2. 레포지토리를 내 pc로 clone 받기
3. 나의 코드 올려보기
4. 브랜치를 이용하기
5. 충돌 해결하기



git 사용자 정보 설정

```
git config --global user.name "사용자이름"
```

```
git config --global user.email "이메일@abc.com"
```

```
git config --list
```




GitHub 협업 전략


Pull Request 란 ?

저장소의 내용을 내가 가져와서 업데이트 한 후에
이런 내용을 바꿨으니 main에 합쳐달라고 요청하는 의미

깃허브 branch에 코드를 올릴때 뜨게 된다




Pull Request

 **pprac** Public

 Pin

Unwatch 1

 **dongwan-branch** had recent pushes 4 seconds ago

Compare & pull request

 main ▼

2 Branches 0 Tags

Q Go to file


t

Add file ▾

<> Code ▾

dongwan97 branch용

7e958f0 · 10 minutes ago ⌚ 3 Commits

 README.md

first commit

13 minutes ago

branch.py


branch용

10 minutes ago

dongwan.py

first commit

12 minutes ago

 **README**



GitHub branch protection Rules



wandong
dongwan97

Edit profile

👤 2 followers · 4 following

📖 Dongguk Univ.

📍 Seoul

🕒 16:52 (UTC +09:00)

Find a repository...

Type ▾

Language ▾

Sort ▾

📁 New

Pethub-webProject Public

[웹프로그래밍 프로젝트] 강아지 건강관리 웹사이트

react html-css

🟡 JavaScript 🐙 1 Updated 2 weeks ago

☆ Star ▾

Smoking-Prediction-Model Public

공공데이터를 활용한 흡연 여부 예측

python machine-learning r eda

🟠 Jupyter Notebook Updated 2 weeks ago

☆ Star ▾

miraeAsset-PROTOTYPE-APP Public

Forked from [BigDataFestival-Team-kkirikkiri/PROTOTYPE-APP](#)

[미래에셋 빅데이터 페스티벌] 프로토타입 앱

prototype flutter

☆ Star ▾

branch 전략

The screenshot shows the GitHub repository settings for a repository named 'prac' owned by 'dongwan97'. The 'Settings' tab is selected in the top navigation bar. On the left sidebar, the 'General' section is expanded, and 'Branches' is selected under 'Code and automation'. The main content area is titled 'Branch protection rules' and displays a message: 'You haven't protected any of your branches'. It explains that a protected branch rule can be used to disable force pushing, prevent branch deletion, and require status checks before merging. A link 'Learn more about protected branches' is provided. Below the message is a button labeled 'Add branch protection rule'.

dongwan97 / prac

Search: Type / to search

Navigation: <> Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings

Left sidebar (General section):

- Access
 - Collaborators
 - Moderation options
- Code and automation
 - Branches**
 - Tags
 - Rules
 - Actions
 - Webhooks
 - Environments
 - Codespaces
 - Pages

Main content: Branch protection rules

You haven't protected any of your branches

Define a protected branch rule to disable force pushing, prevent branches from being deleted, and optionally require status checks before merging. [Learn more about protected branches](#)

Add branch protection rule

branch 전략

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Autolink references

Protect your most important branches

Branch protection rules define whether collaborators can delete or force push to the branch and set requirements for any pushes to the branch, such as passing status checks or a linear commit history.

Branch name pattern *

main

Protect matching branches

☒ Require a pull request before merging

When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.

☒ Require approvals

When enabled, pull requests targeting a matching branch require a number of approvals and no changes requested before they can be merged.

Required number of approvals before merging: 1

☐ Dismiss stale pull request approvals when new commits are pushed

New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

☐ Require review from Code Owners

Require an approved review in pull requests including files with a designated code owner.

☐ Require approval of the most recent reviewable push

Whether the most recent reviewable push must be approved by someone other than the person who pushed it.