

22기 정규세션

ToBig's 21기 김지은

GIT 기초 및 실습

CONTENTS

Unit 01 | GIT이란?

Unit 02 | GIT 기초 및 명령어

Unit 03 | 과제 안내

Unit 01

GIT이란?

Unit 01 | GIT이란?

- GIT이란 :

- 분산형 버전 관리 시스템(Version Control System)의 한 종류이며, 빠른 수행 속도에 중점을 둠



Unit 01 | GIT이란? – 버전 관리

• 버전 관리 :

- 파일 변화를 시간에 따라 기록했다가 나중에 특정 시점의 버전을 다시 꺼내올 수 있는 시스템을 의미
 - 복사, 백업, 저장, 뒤로 가기 등과 같은 과정을 거치게 됨
- 내가 원하는 시점(버전)으로 이동할 수 있게 해주는 것이 버전 관리!
- 컴퓨터 파일의 변경사항을 추적해주는 툴 = 버전 관리 시스템 = GIT

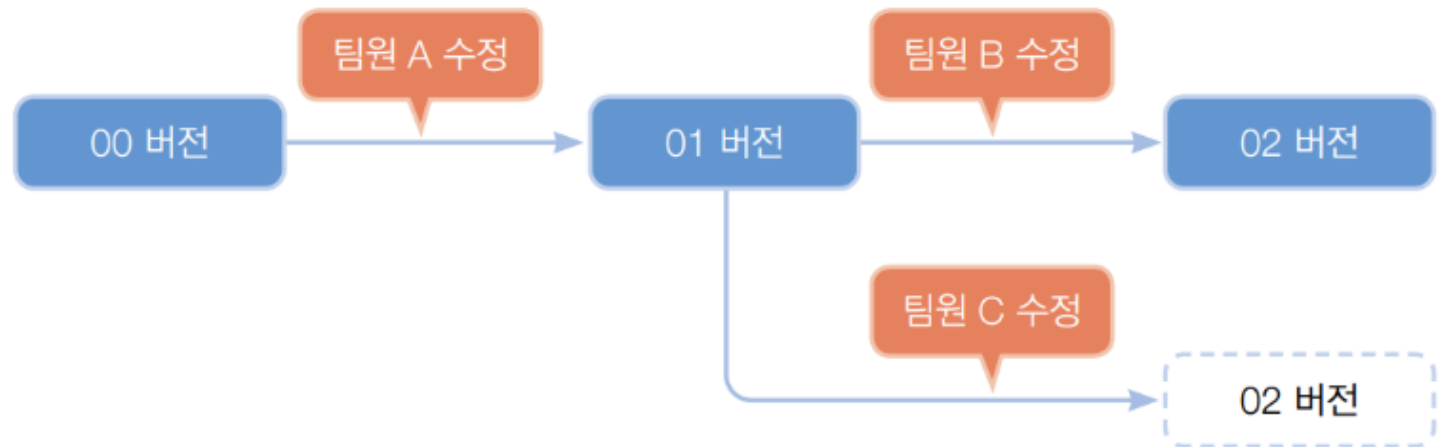
최종_시안.png
최종_시안_(1).png
진짜최종.png
Final.png
이게_마지막_시안_.png
이게_진짜_끝.png
끝.png
최종(2).png
진짜진짜_진짜_최종.png
위에거_다아니고_이게_최종.png



Unit 01 | GIT이란? - GIT의 필요성

GIT의 필요성:

- 01 버전 → 팀원 B, 팀원 C, 팀원 D + a 동시에 수정하게 된다면?..
 - 버전을 동시에 업로드 하게 되면,,? → 지워지거나, 겹치는 일 발생
 - 최종 업데이트 파일이 무엇인지 확인하기 어려움+복잡함



Unit 01 | GIT이란? - GIT의 장점

• GIT을 사용하면 좋은 점 :

- 위의 문제점들을 사전에 방지
- 같은 파일을 여러 명이 동시에 작업하는 병렬 개발이 가능
- 충돌을 최소화 할 수 있음

→ 대부분의 개발자가 Git으로 협업을 진행



Unit 01 | GIT이란? GIT vs GitHub

- **GIT vs GitHub :**

- **GIT** : 버전 관리 시스템
- **GitHub** : Git으로 관리하는 프로젝트를 올려둘 수 있는 사이트 중 하나

Ex. 블로그를 만들 수 있는 사이트 : 네이버, 다음 등등

→ **Git으로 관리하는 프로젝트를 올릴 수 있는 사이트들** 중 하나가 바로 **GitHub**(+GitLab, BitBucket 등)

- **GitHub의 장점 :**

- 오픈 소스
- 시간, 공간의 제약 없이 협업 가능



Unit 02

GIT 기초 및 명령어

Unit 02 | GIT 기초 및 명령어 – Git Repository

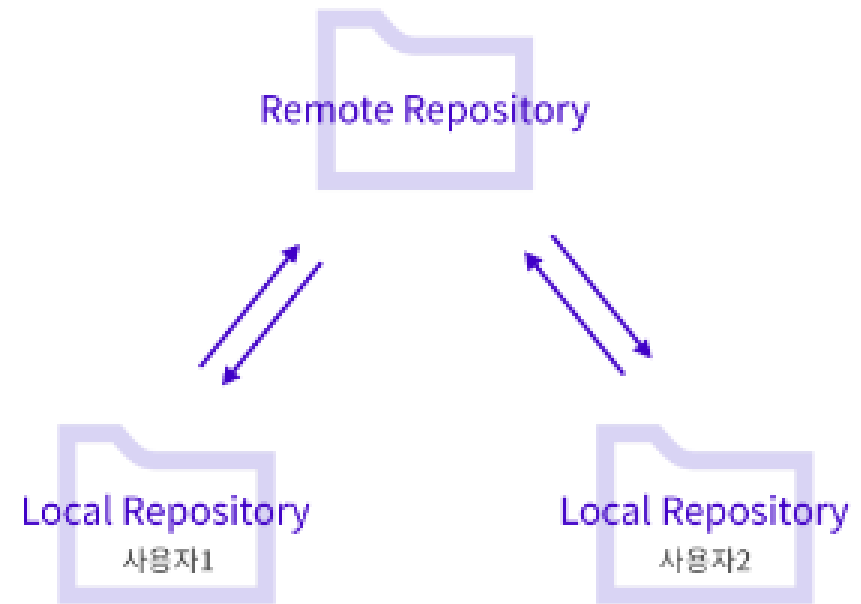
• Git Repository :

- Git으로 관리되는 폴더 = 파일이나 폴더를 저장해두는 곳
- Remote Repository와 Local Repository 두 종류의 저장소를 제공

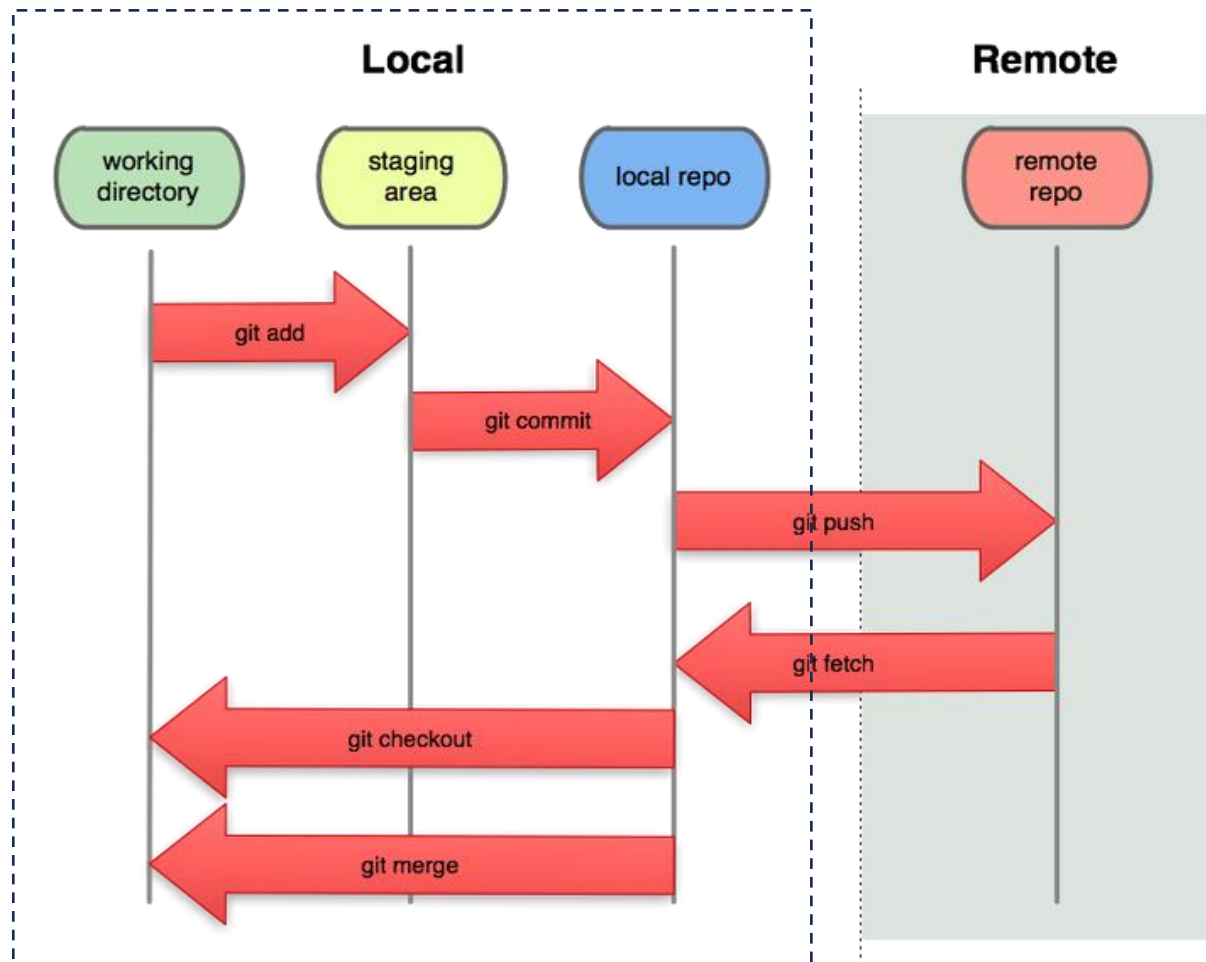


Unit 02 | GIT 기초 및 명령어 – Git Repository

- **Remote Repository (원격 저장소)**
 - 원격 서버(GitHub)에 저장된 저장소로, 여러 사람이 함께 공유
- **Local Repository (로컬 저장소)**
 - 사용자가 직접 관리하는 저장소로, 내 PC에 저장되어 있음



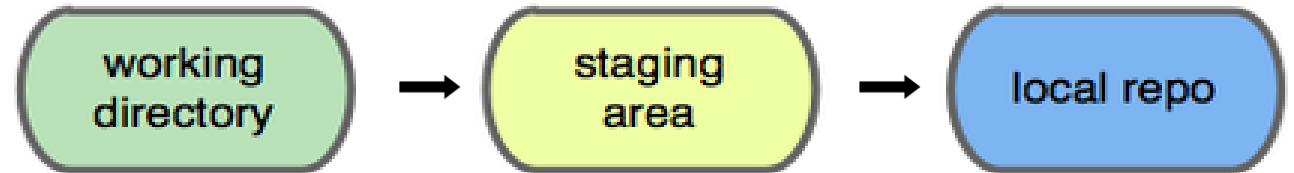
Unit 02 | GIT 기초 및 명령어 - Framework



Unit 02 | GIT 기초 및 명령어 – 로컬 저장소

- **Working Directory(작업공간)**

- 파일 수정, 저장 등의 작업을 하는 디렉토리



- **Staging Area(대기실)**

- 버전으로 만들 파일이 대기하는 곳

ex. Working Directory에서 n개의 파일을 수정했는데, 2개의 파일만 버전으로 만들려면 2개의 파일만 스테이지로 넘겨주면 된다.

- **local Repository(저장소)**

- 스테이지에서 대기하고 있는 파일들을 버전으로 만들어 저장하는 곳
- 버전 관리가 이루어지는 곳

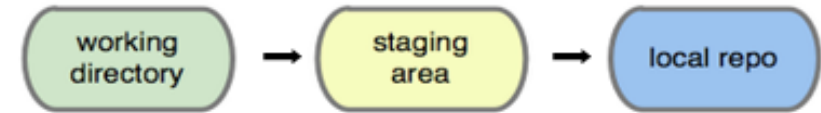
✓ 스테이지와 저장소는 눈에 보이지 않음.

✓ 깃을 초기화했을 때 만들어지는 .git 디렉터리 안에 숨은 파일 형태로 존재하는 영역



Unit 02 | GIT 기초 및 명령어 – 로컬 저장소

<로컬 저장소 만들기>



• 저장소 새로 만들기

- `git init`
- 저장소를 만들 디렉토리 위치에서 실행

```
$ git init
Initialized empty Git repository in D:/Jupyter_Notebook/루빅스/git_실습/.git/
```

• 수정할 코드를 본인의 레파지토리 폴더로 fork 후, `git clone`

- 원격 저장소를 로컬로 복사해오기
- `git clone '원격 저장소 주소'`
- 실행한 디렉토리에 원격 저장소 폴더가 복사됨

```
$ git clone https://github.com/Ji-eun-Kim/tobigs-22nd.git
Cloning into 'tobigs-22nd'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 32 (delta 4), reused 19 (delta 0), pack-reused 0
Receiving objects: 100% (32/32), 10.66 KiB | 10.66 MiB/s, done.
Resolving deltas: 100% (4/4), done.
```

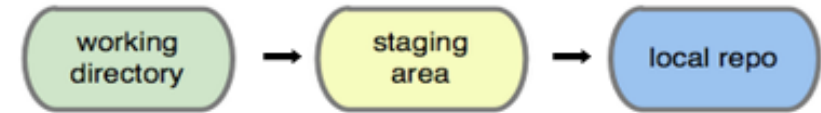
코드 수정

- `Code .` 명령어 후, 코드 수정

```
$ code .
```



Unit 02 | GIT 기초 및 명령어 – 로컬 저장소



<Working Directory → Staging area>

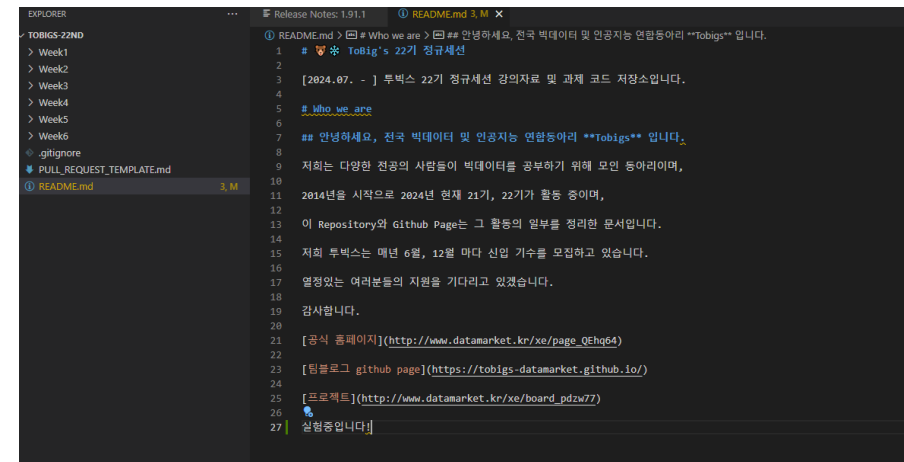
- 처음 코드 작성 시, working directory에 위치하는 파일들, 로컬에 올리기 전 대기시키기(Staging)

• 원하는 파일들만 인덱스에 등록 후, 커밋 진행

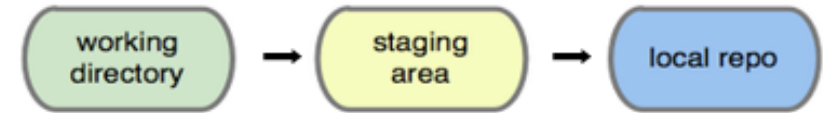
- `git add [파일명]` 실행 시, staging area로 이동
- `git add .` → 모든 파일이 이동

```
$ git add README.md
```

```
(김지은)  
$ git add .
```



Unit 02 | GIT 기초 및 명령어 – 로컬 저장소



<Staging area → local repository>

- 로컬 저장소(local repository)에 올리기, 기록하기

• 로컬 저장소로 커밋

- `git commit -m '커밋 메시지'` 실행 시 로컬 저장소로 커밋
- 어떤 변경 사항이 있었는지 확인하기 위해 메시지를 함께 기록
- 이전 커밋 상태부터 현재 상태까지의 변경 이력이 시간 순으로 기록된 커밋(혹은 리비전)이 만들어짐
- 커밋은 이력을 남기는 중요한 작업, 커밋 메시지는 필수!
 - 명료하고 이해하기 쉽게 남겨야 본인 뿐만 아니라, 다른 사람이 커밋 이력을 확인하기 쉬움

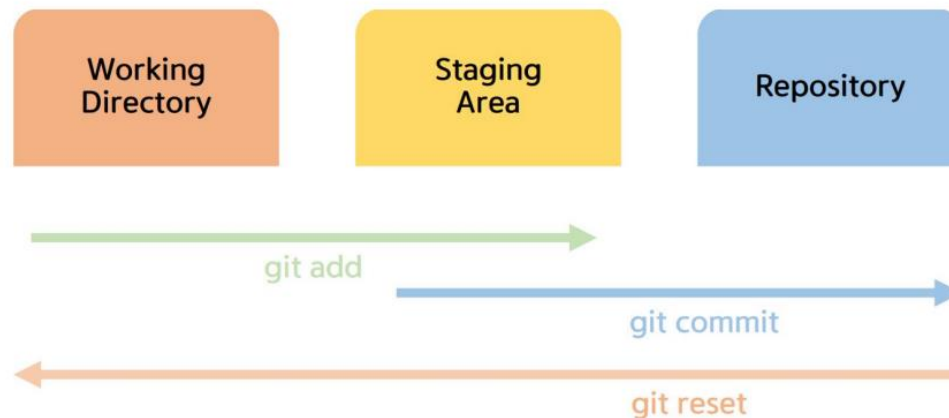
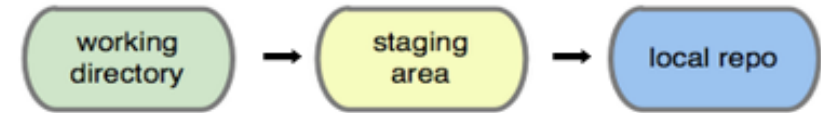
```
$ git commit -m 'README.md 파일 수정 진행'
[김지은 b6bb8f7] README.md 파일 수정 진행
1 file changed, 2 insertions(+)
```



Unit 02 | GIT 기초 및 명령어 – 로컬 저장소

<Local Repository → working directory>

- `git reset` : 과거 버전으로 되돌아감
- `git reset --hard` : 보다 강제로 되돌리기
- `git reset [고유이름]` or `git reset --hard [고유이름]`
 - 특정 버전을 지정하여 이동할 수 있음



Unit 02 | GIT 기초 및 명령어 – 원격 저장소

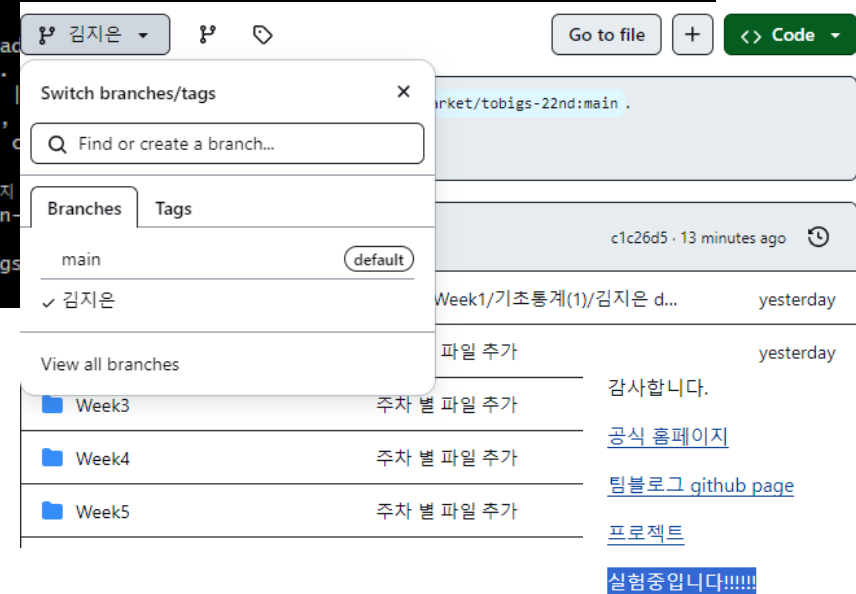
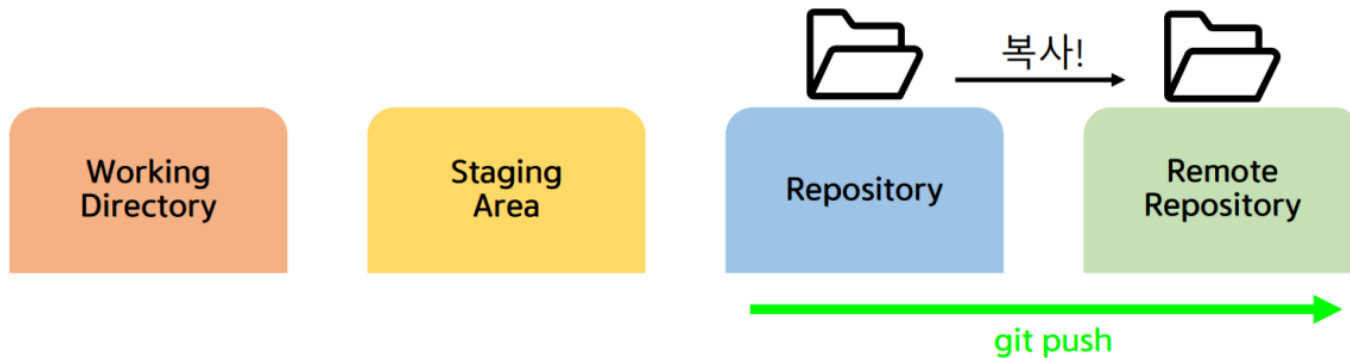
<Local Repository → Remote Repository>

- 내 로컬 저장소에 있는 내용을 원격 저장소 서버로 올리기
- `git push [Remote명] [Branch명]`
- `git push origin main`이 기본



```

JieunKim@DESKTOP-LLAK3MI MINGW64 /d/Jupyter_Notebook/두 박 스 /git_실 습 /tobigs-22nd (김 지 은)
$ git push origin 김 지 은
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 1.63 KiB |
Total 7 (delta 4), reused 0 (delta 0),
remote: Resolving deltas: 100% (4/4), done.
remote:
remote: Create a pull request for '김 지 은'
remote: https://github.com/Ji-eun-Kim/tobigs-22nd/pulls
remote:
To https://github.com/Ji-eun-Kim/tobigs-22nd
 * [new branch] 김 지 은 -> 김 지 은
  
```



Unit 02 | GIT 기초 및 명령어 – PR

<Compare & Pull Request>

- Pull Request 진행
- PR를 받은 원본 저장소 관리자는 코드 변경 내역을 확인하고 Merge 여부를 결정

had recent pushes 46 seconds ago

Compare & pull request

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#) or [compare across repositories](#).

base repository: tobigs-datamarket/tobigs-22nd base: main head repository: Ji-eun-Kim/tobigs-22nd

✓ Able to merge. These branches can be automatically merged into the base branch.

Add a title

김투빅(mgi) - n주차 과제

Add a description

Write Preview H B I

김투빅(mgi) - n주차 과제

<!-- PR 오픈 전, 제목의 이름과 기수, 주차를 편집해주세요! -->

제출하기 전

> 만족하는 항목을 체크해주세요. 만족하지 못하는 항목이 있을 경우 사유를 작성해주세요.

- [] 과제 원본 파일을 변경하지 않았습니다.
- [] 과제 제출 대상 브랜치를 올바르게 선택했습니다.
- [] 과제 채점에 불필요한 데이터나 모델 파일을 포함하지 않습니다.
- [] 과제 파일 조회 및 실행에 문제가 없습니다.

Choose a base ref

Find a branch

Branches Tags

✓ main default

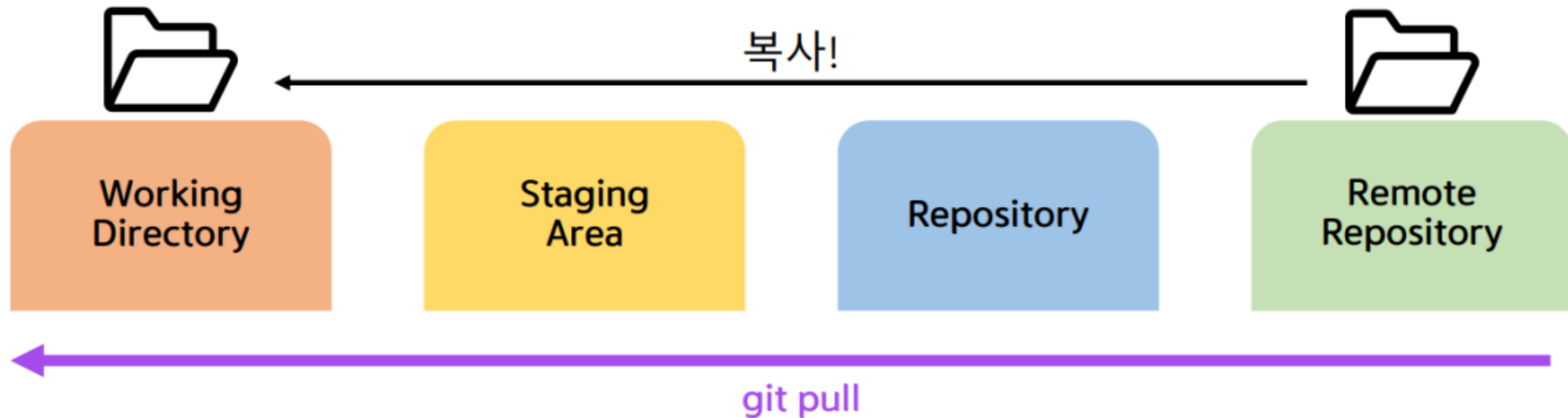
과제_제출



Unit 02 | GIT 기초 및 명령어 – 원격 저장소

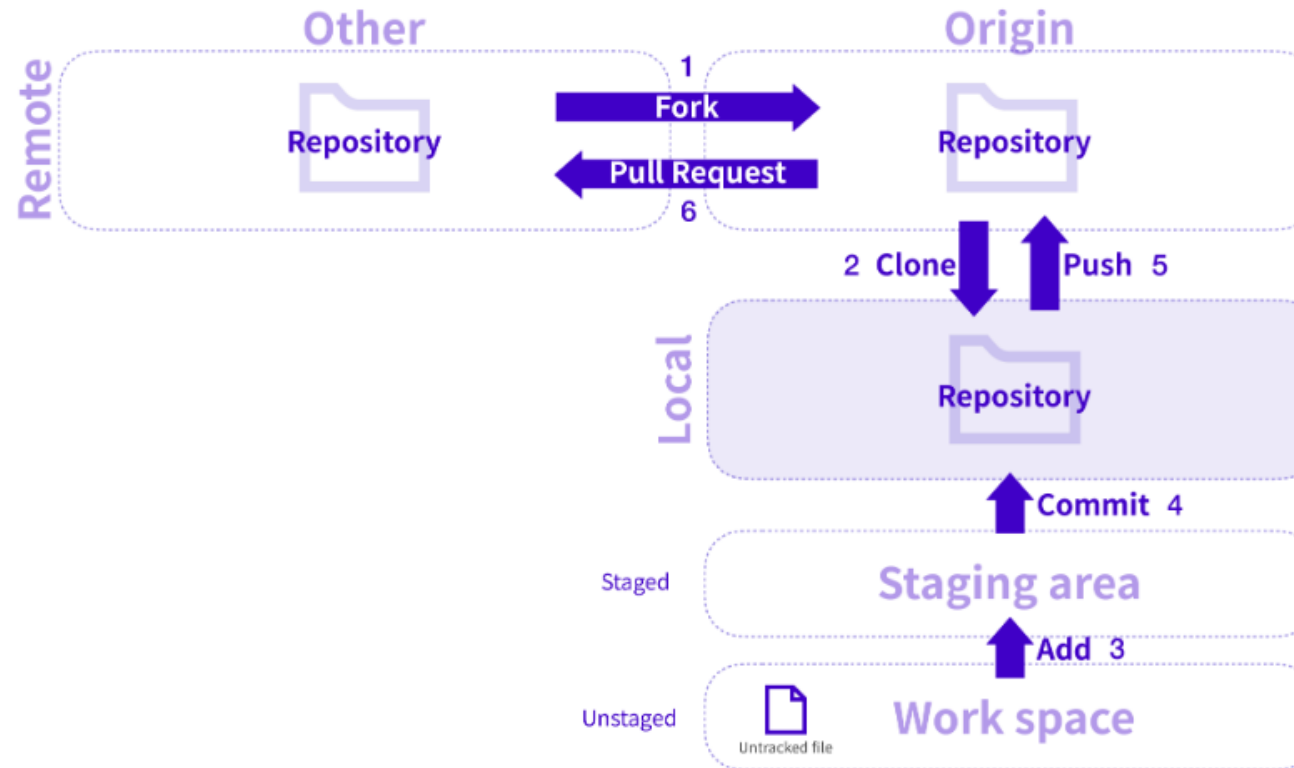
<remote repository → working directory>

- 원격저장소에서 최신 내용 가져와서 내 WD 업데이트
- `git pull [Remote] [Branch명]` 실행 시 원격 저장소에서 최신 내용 받아오기
- `git pull origin main`이 기본



Unit 02 | GIT 기초 및 명령어 – 마무리

Github workflow overview



Unit 03

과제 안내

Unit 03 | 과제 안내

• 과제 제출 안내

- 반드시 **본인의 이름으로 branch** 생성 후, 코드 과제 진행
- PR 진행 시, 반드시 **과제 제출 branch**로 PR
- 원본 파일 절대 수정/삭제 금지!!(main branch는 건들지 말아주세요)

김투빅(m기) - n주차 과제

제출하기 전

만족하는 항목을 체크해주세요. 만족하지 못하는 항목이 있을 경우 사유를 작성해주세요.

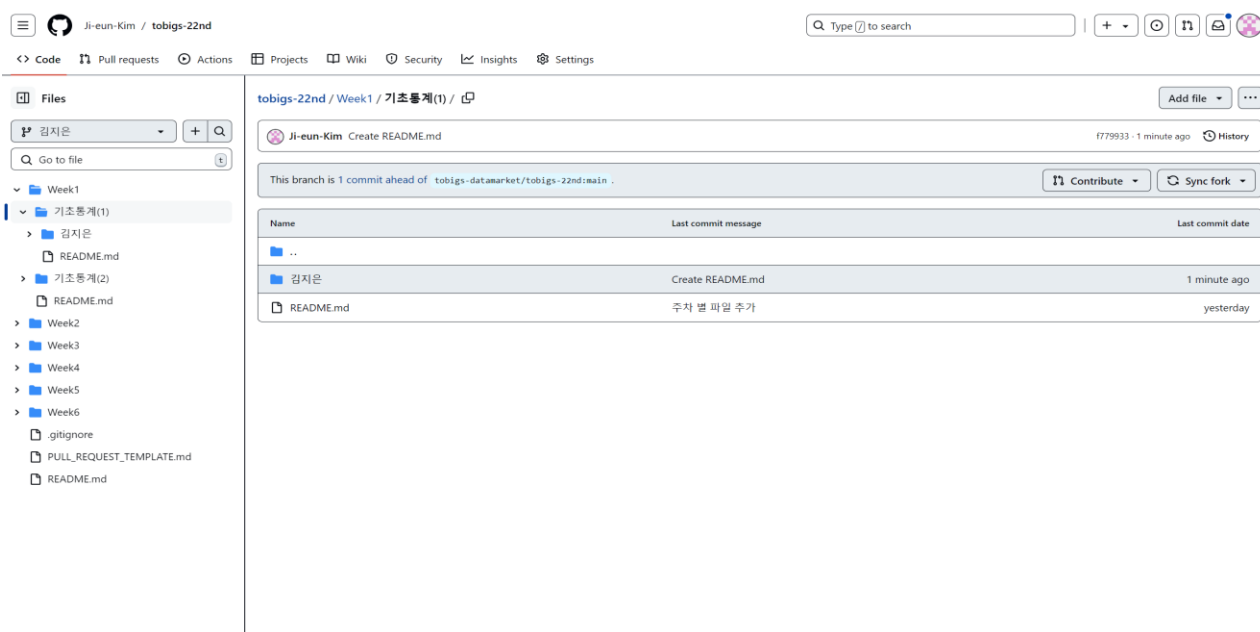
- ☐ 과제 원본 파일을 변경하지 않았습니다.
- ☐ 과제 제출 대상 브랜치를 올바르게 선택했습니다.
- ☐ 과제 채점에 불필요한 데이터나 모델 파일을 포함하지 않습니다.
- ☐ 과제 파일 조회 및 실행에 문제가 없습니다.



Unit 03 | 과제 안내

• 과제 제출 안내

- 본인의 Branch 생성 후, 과제를 제출할 폴더 안에 본인의 이름 명 폴더 생성
- 이후, 본인 이름 폴더 안에 과제 파일 업로드 후, 과제 제출 Branch에 PR
- 원본 파일 절대 수정/삭제 금지!!(main branch는 건들지 말아주세요)



```
tobigs-22st
├── Week1
│   ├── 기초통계1
│   │   ├── 김투빅
│   │   │   ├── example.md # 과제 파일
│   │   │   ├── PR 과정1.png
│   │   │   └── PR 과정2.png
│   └── 기초통계2
│       ├── 김투빅
│       └── PR 과정.pdf
```



Unit 03 | 과제 안내

1주차 GIT 과제 안내

- 기초 통계(1), 기초 통계(2)
- 과제 수행 후, GIT 명령어를 사용해 PR까지의 과정을 캡처
 - 각 과제 폴더에 png or pdf 형태로 업로드해서 제출하기
 - 관련 안내 사항은 7/21(일) 중으로 재공지 예정

```
tobigs-22st
└─ Week1
   │   ├── 기초통계1
   │   │   └─ 김투빅
   │   │       ├── example.md # 과제 파일
   │   │       ├── PR 과정1.png
   │   │       └─ PR 과정2.png
   └─ 기초통계2
       └─ 김투빅
           └─ PR 과정.pdf
```



Unit 03 | 과제 안내

과제 마감 기한

- 다음 주차 진행하는 주의 화요일 23시 59분까지(엄수)
 - 다음 세션 날짜 : 8/3(토)
 - 과제 마감 기한 : 7월 30일(화) 23시 59분까지

```
tobigs-22st
└─ Week1
   └─ 기초통계1
      └─ 김투빅
         └─ example.md # 과제 파일
            └─ PR 과정1.png
               └─ PR 과정2.png
└─ 기초통계2
   └─ 김투빅
      └─ PR 과정.pdf
```



Unit 03 | 과제 안내

GIT 관련 문의사항

21기 김지은, 21기 신기섭
운영진(교육부)



REFERENCE

https://m.hanbit.co.kr/channel/category/category_view.html?cms_code=CMS2036561776
<https://velog.io/@oseo/Git-%EB%82%B4%EA%B0%80-%EB%B3%B4%EB%A0%A4%EA%B3%A0-%EC%93%B0%EB%8A%94-%EA%B9%83-%EC%A0%95%EB%A6%AC-%EA%B8%80>
[https://inpa.tistory.com/entry/GIT-%E2%9A%A1%E2%8F%A0-%EA%B0%9C%EB%85%90-%EC%9B%90%EB%A6%AC-%EC%89%BD%EA%B2%8C%EC%9D%B4%ED%95%B4#%EC%9B%90%EA%B2%A9%EC%A0%80%EC%9E%A5%EC%86%8Cremote repository](https://inpa.tistory.com/entry/GIT-%E2%9A%A1%E2%8F%A0-%EA%B0%9C%EB%85%90-%EC%9B%90%EB%A6%AC-%EC%89%BD%EA%B2%8C%EC%9D%B4%ED%95%B4#%EC%9B%90%EA%B2%A9%EC%A0%80%EC%9E%A5%EC%86%8Cremote_repository)
<https://velog.io/@seunghyun/Git-GitHub-%EA%B0%9C%EB%85%90%EC%A0%95%EB%A6%AC>
<https://co-no.tistory.com/entry/Git-Git%EC%9D%98-%EA%B8%B0%EB%B3%B8-%EA%B0%9C%EB%85%90%EA%B3%BC-%EC%9A%A9%EC%96%B4-%EC%A0%95%EB%A6%AC>
<https://wayhome25.github.io/git/2017/07/08/git-first-pull-request-story/>
<https://velog.io/@jhp4986/Git-%EC%88%9C%EC%84%9C-%EC%99%80-%EB%AA%85%EB%A0%B9%EC%96%B4-%EC%A0%95%EB%A6%AC>
[투빅스 20기 양지우님 1주차 정규세션 내용 참고](#)



Thank
you

감사합니다.