

Git과 GitHub

2021.12



누구나 한번쯤은 겪었던 일

한달후에 최종본을 찾으려고 했는데 이름이 기억나지 않음



최종 Report.doc



최종 Report(0307).doc



최종 Report(진짜 최종).doc



최종 Report(최종).doc

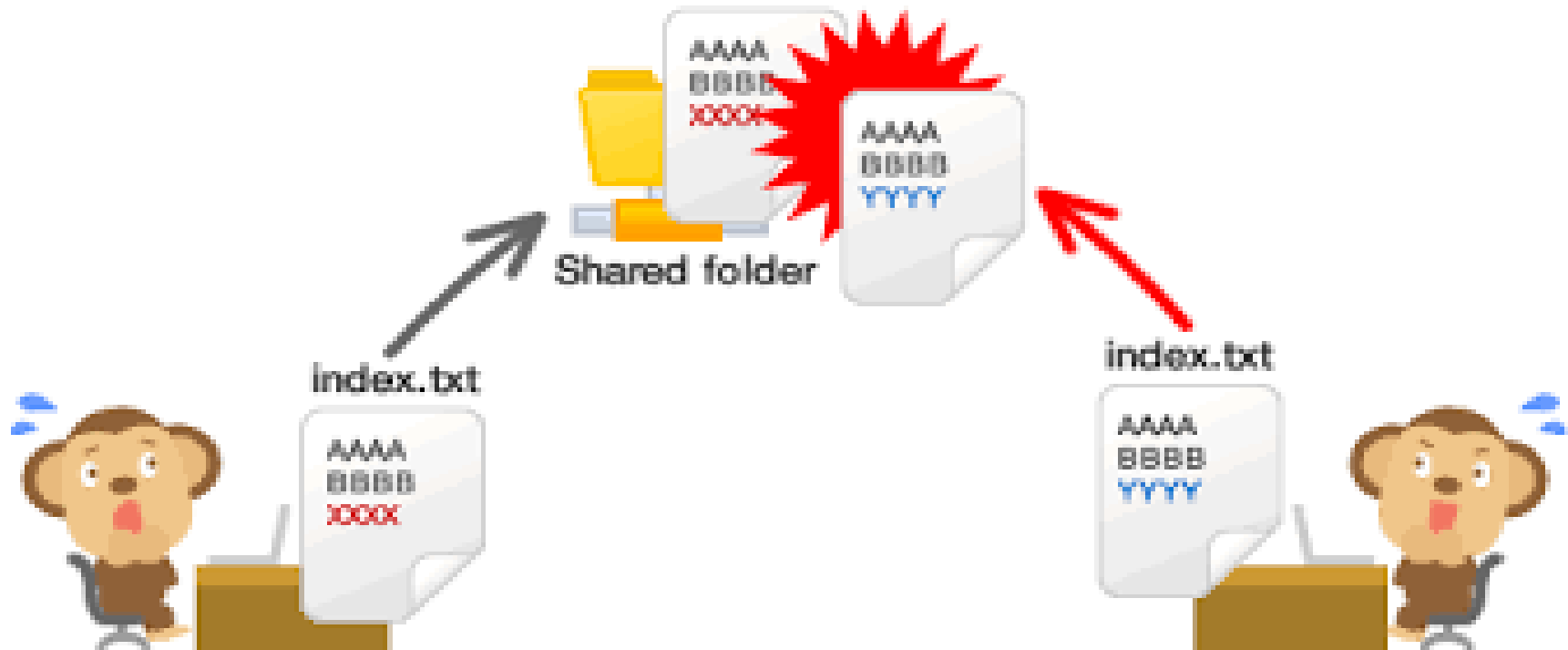


최종 Report(완성본).doc



누구나 한번쯤은 겪었던 일

여러명이 공동으로 같은 파일을 작업할 때 파일 내용을 덮어쓰는 문제가 생길 가능성



1. GitHub 란?

■ 정의

분산 버전 관리 툴인 깃(Git)을 사용하는 프로젝트를 지원하는 웹호스팅 서비스이고 현재 가장 인기있는 오픈 소스 코드 저장소

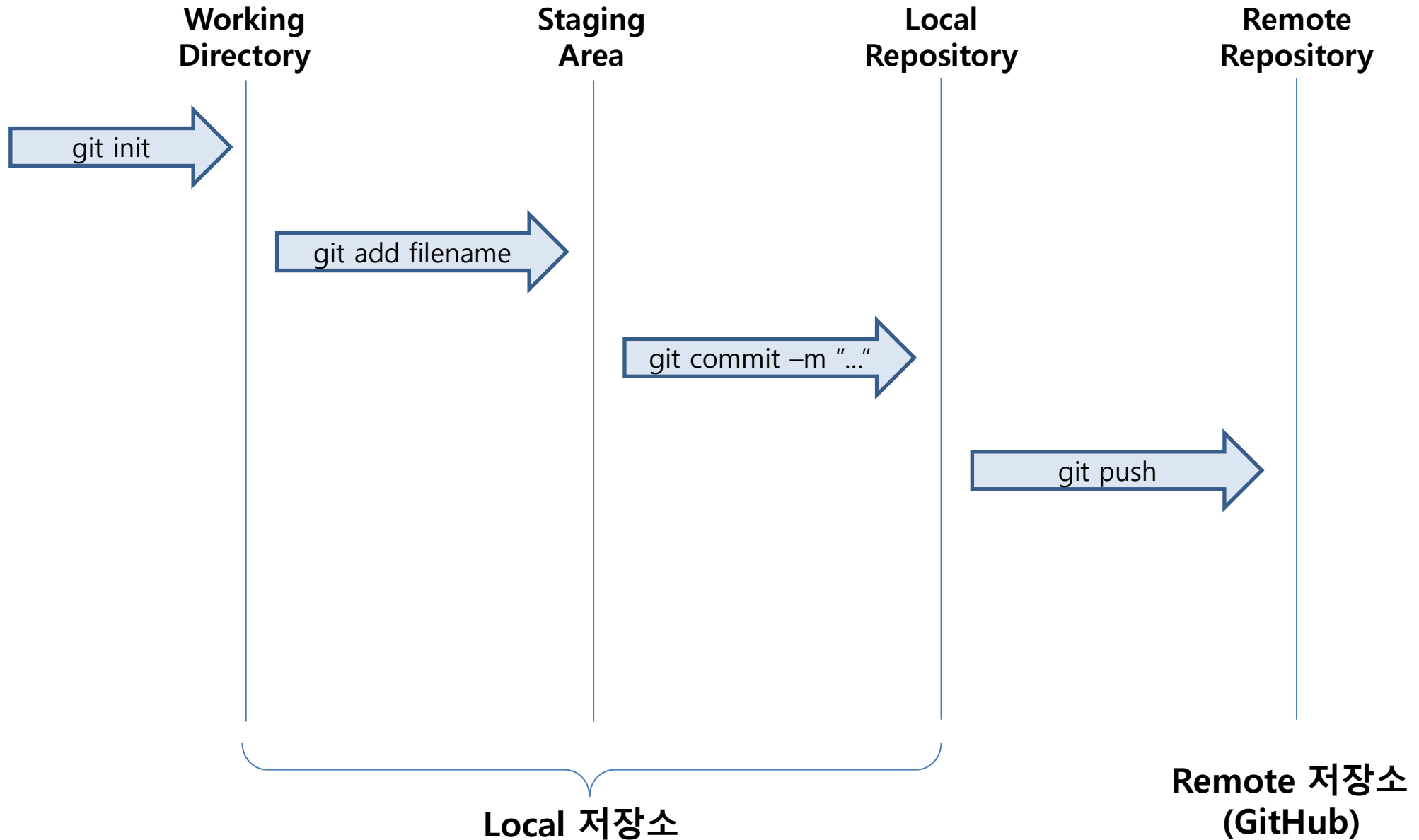
■ 특징

- Open 했을 때 무료
- 주제 별/프로젝트 별로 저장소를 만들 수 있음
- 2명 이상의 협력자(Collaborators)를 등록하여 하나의 프로젝트를 가지고 GitHub를 통해 공동 작업할 수 있음

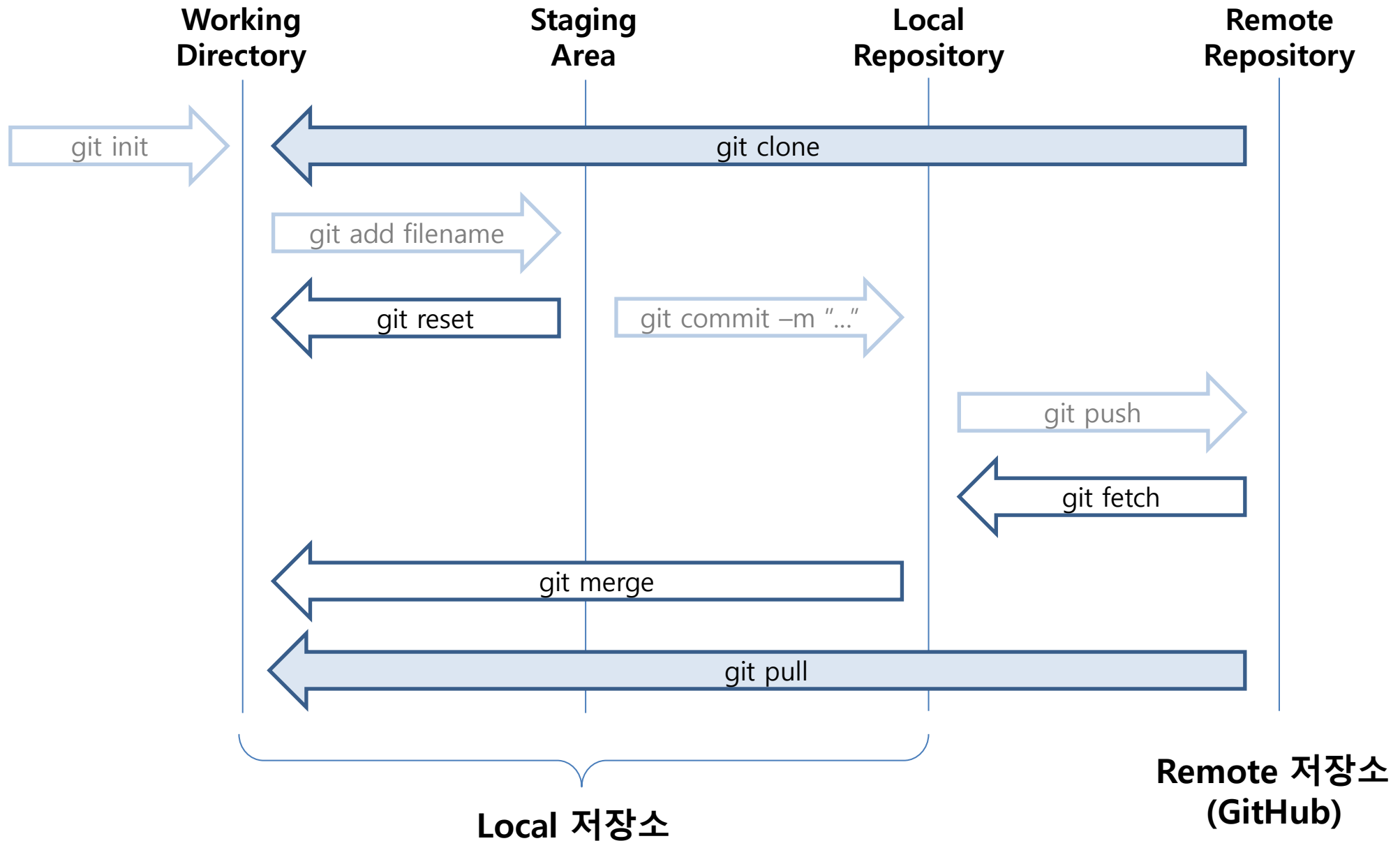
■ 현실적인 의미

- 유명한 오픈 소스 프로그램들의 저장소 및 협업 장소
- 개발자의 이력을 증명하기 위한 수단 (이력서/포트폴리오/Blog 대체)

2. Git의 동작 원리



2. Git의 동작 원리



3. Git 사용법

1. Windows용 git 다운로드 후 설치 (Default Setting)

<https://git-scm.com/download/win>

2. 설치가 제대로 되었는지 확인 후 초기 세팅 / GitHub에 회원 등록

```
git --version (hyphen 기호가 2개)
```

```
git config --global user.name "Bigdata Kim"
```

```
git config --global user.email bigdata@naver.com
```

3. 작업할 디렉토리(예, d:\Workspace\python)에서 새로운 저장소를 생성

cmd (새로운 명령 프롬프트 앱 실행)

d:

```
cd \Workspace\python
```

```
git init (하나의 PC에 여러 개의 저장소 생성 가능)
```

3. Git 사용법

4. Working Directroy에서 작업 수행

`notepad test.txt`

5. 파일을 Staging Area(준비 상태)로 보내기

`git add filename`

(통상 `git add .` 으로 사용)

`git status` 로 상태 확인 가능

6. 파일을 확정함

`git commit -m "First Commit"` (따옴표 안에 쓰고 싶은 말 쓰면 됨)

`git log` 로 결과 확인 가능

3. Git 사용법

7. GitHub 사이트에 들어가서 새로운 저장소(Repository) 생성
이름(Test)과 간단한 설명(Testing GitHub Operation)을 입력

8. 기본 생성 branch 이름을 'master'에서 'main'으로 변경

```
git branch -M main
```

9. 필요한 URL과 Command 복사한 후 "명령 프롬프트" 앱에서 실행

```
git remote add origin https://github.com/ckiekim/Test.git
```

10. 로컬에서 확인한 파일을 원격저장소에 보관하고 배포함

```
git push -u origin main (최초 실행시)
```

```
git push (이후에는 git push만 하면 됨)
```

4. README.md

■ README.md 란?

- 저장소에 대한 소개글로 최대한 성의있게 작성할 것
- Markdown 문서로 쉽게 배울 수 있음 ("GitHub Readme 작성"으로 검색)
- GitHub 사이트에서 주로 편집함 (결과 미리보기 활용)

■ Local과 동기화

```
git pull
```

5. 선택적 저장

■ GitHub에 올리지 말아야 할 파일

- 임시 파일
- 100 MB 이상되는 대용량 데이터 파일
- 패스워드

■ .gitignore

```
# 임의로 파일이나 디렉토리를 지정
password.txt
largefile.csv
data_folder/

# Jupyter Notebook
.ipynb_checkpoints

# Byte-compiled
__pycache__/
```

6. 집과 사무실에서 사용하는 경우

■ 최초 작업시

- github.com 사이트에 repository(project) 만들기

- 사무실(집)에서

```
git init
```

```
git add .
```

```
git commit -m "Commit message"
```

```
git branch -M main
```

```
git remote add origin https://github.com/~project.git
```

```
git push -u origin main
```

- 집(사무실)에서

```
git clone https://github.com/~project.git project-folder
```

6. 집과 사무실에서 사용하는 경우

■ 이후 작업시

- 사무실(집)에서

```
git add .
```

```
git commit -m "Commit message"
```

```
git push
```

- 집(사무실)에서

```
git pull
```

7. 많이 하는 실수

■ 디렉토리(폴더) 주의

- 하위 폴더에서 git init을 하면 안됨
- 폴더별로 git init을 하여 GitHub와 연동할 것

C:\

D:\workspace --+-- .git



+-- Python --+-- .git

+-- DataAnalysis --+-- .git

+-- MachineLearning --+-- .git

+-- DeepLearning --+-- .git

- git init을 하기 전에 현재 디렉토리 구조를 파악할 것

■ 대형 파일(100MB 이상) 주의

일일 커밋

(Daily Commit)