

git

Index

1. git

- 개요
- 주요 개념

2. Git 명령어

3. git 관리

- .gitignore
- .gitkeep
- requirements.txt

- README.md

4. README

- 작성 요령



Version Control System

Git (소프트웨어)

- 컴퓨터파일의 변경사항을 추적하고 여러 사용자들 간의 작업 조율을 위한 분산 버전 관리 시스템
- 독립적인(인터넷 연결 X) 이력관리, 버전추적 등 기능을 갖춘 저장소

기본 개념

- 저장소 : 파일이나 폴더를 저장해두는 곳
- branch : 독립적인 작업을 진행하기 위한 개념
- commit : 변경사항을 기록



```
MINGW64:/c/Users/sosin/Desktop/Sesac/only-pull-me
sosin@DESKTOP-NLQAF93 MINGW64 ~/Desktop/Sesac/only-pull-me (master)
$ git log
commit 049ec76d1ecc8e7d8c6aeb4d88e6bdc3cee4a646 (HEAD -> master, origin/master, origin/HEAD)
Author: sosin <svstar94@naver.com>
Date: Fri Sep 16 13:52:52 2022 +0900

    add example6

commit efa83fc8396a7682c5c561ceeb6ee184a3f2d33d
Author: sosin <svstar94@naver.com>
Date: Fri Sep 16 11:13:25 2022 +0900

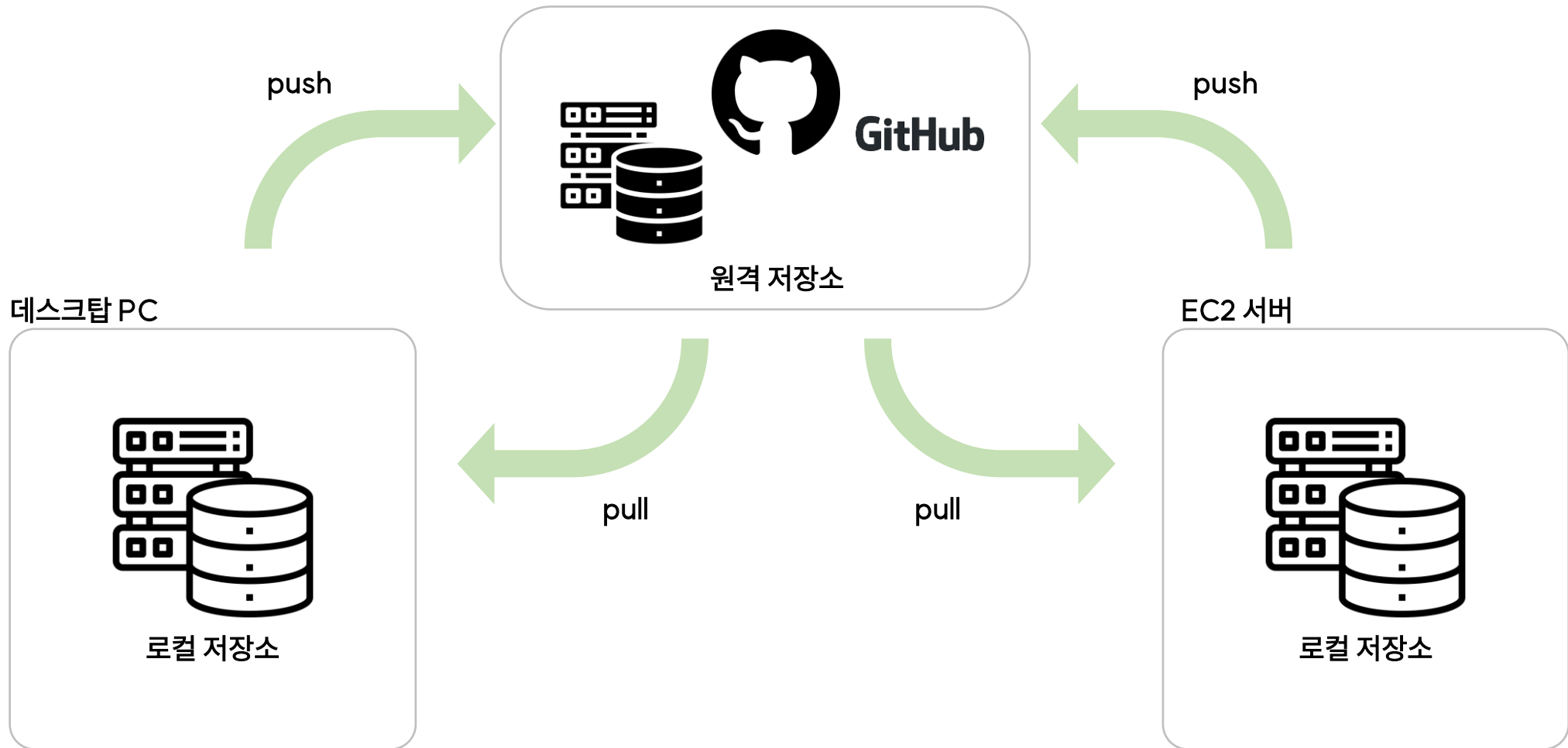
    update group message

commit 48f2facaf2f9b8e66184b64c94a2231d2b57b2fd
Author: sosin <svstar94@naver.com>
Date: Fri Sep 16 10:53:04 2022 +0900

    update message check code

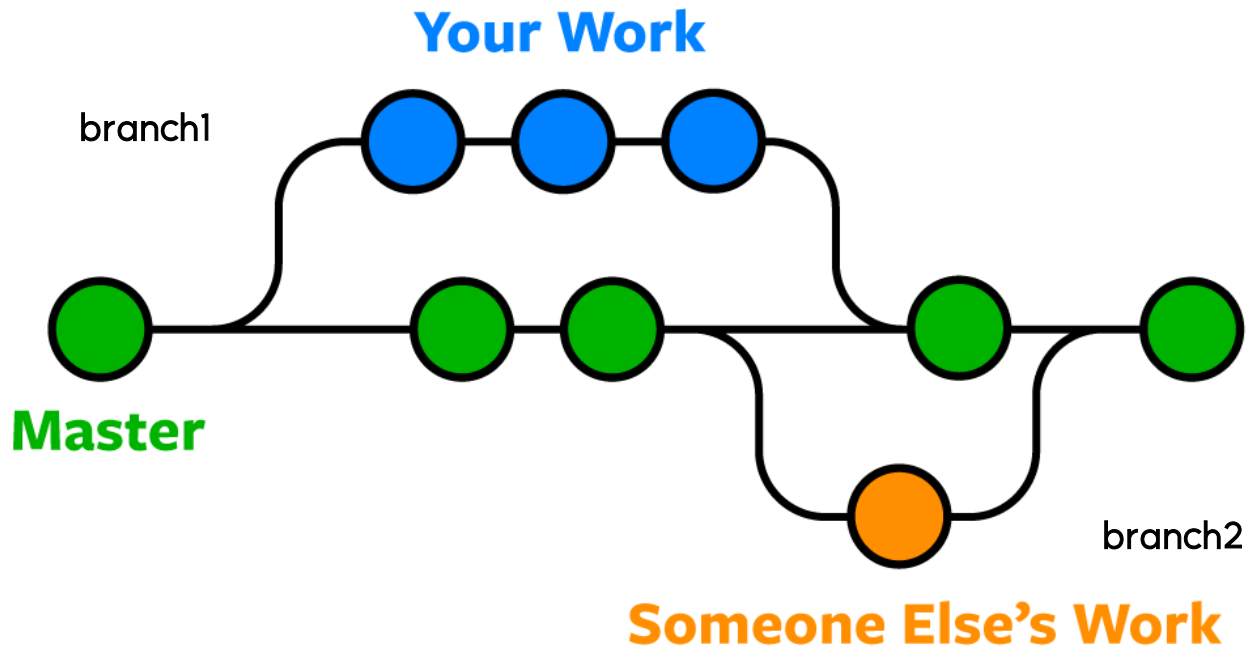
commit 3cfb0f0fa0d2343633edef4c8d26fe64ddb834dd
Author: sosin <svstar94@naver.com>
```

저장소 (Repository)



branch

- 여러 사람과 협업할 경우 Master에서 바로 작업하는 것이 아닌 “기능별로” branch를 따로 만들어서 작업 후 Master에 병합하는 식으로 협업

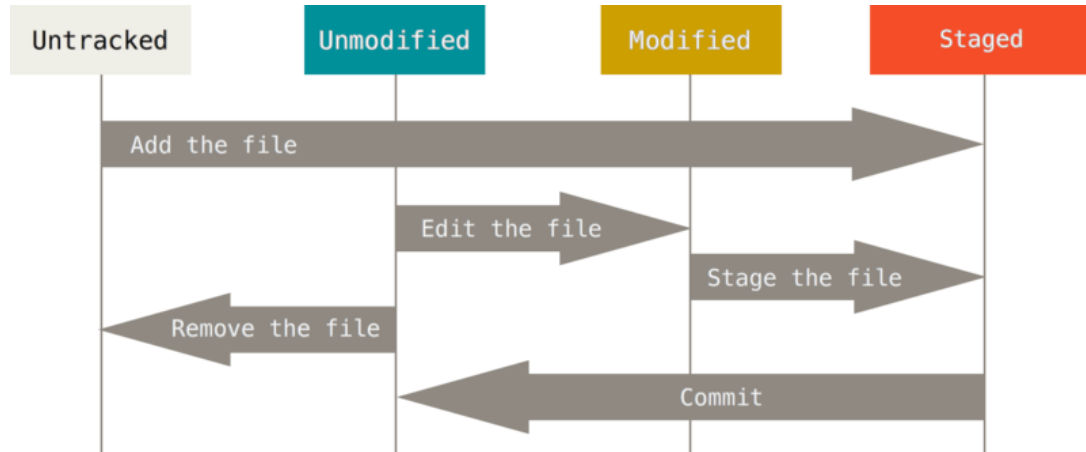


branch 이름 예시) 웹 서비스 개발

- develop : 개발 단계 코드를 merge(합치고), test
- front/dashboard : frontend 작업, dashboard 기능 branch
- back/login : backend 작업, login api 기능 branch

commit

- 파일 추가(add), 파일 삭제(remove), 파일 변경(modified) 사항을 통합하여 하나의 버전으로 추가



※ 커밋 이후 파일들은 unmodified(수정되지 않은) 상태

- 커밋은 파일의 최종 상태를 가진 특정 시점
- 즉, 하나의 버전이라고 할 수 있음

Commands

Git Commands

Git 주요 명령어

- git init : 현재 Working Directory(폴더)를 깃으로 관리 선언
 - ※ 프로젝트 단위로 git init
 - ※ git으로 관리되고 있는 폴더 안에서 다시 git init 하지 않음
- git add [파일목록] : 해당 파일을 변경사항 목록에 추가
- git commit -m "커밋 메시지" : 변경사항들을 하나의 버전으로 등록
- git push [원격저장소] [브랜치] : 원격저장소 브랜치에 현재 버전을 업로드
- git pull [원격저장소] [브랜치] : 원격저장소의 버전으로 로컬저장소 버전을 맞춤
- git remote add [이름] [원격저장소] : 이름으로 원격저장소 추가
- git remote remove [이름] : 해당 이름의 원격저장소 제거
- git clone [원격저장소] : 해당 원격저장소를 로컬저장소에 다운로드

Git 설정

- git config --global user.name [이름] : 깃 유저 이름 설정
- git config --global user.email [이메일] : 깃 유저 이메일 설정

manage

git management

.gitignore

- 깃에서 관리하지 않을 파일들 목록 (데이터, 파일, 이미지, 비밀)
- 특정 파일 혹은 폴더까지의 상대경로를 지정
- 모든 문자열 : *
- .png로 끝나는 모든 파일을 제외 = *.png
- image 폴더의 모든 파일 제외 = image/*
- secret이 들어간 모든 파일 제외 = *secret*

```
# Created by https://www.toptal.com/developers/gitignore/api/macos
# Edit at https://www.toptal.com/developers/gitignore?templates=macos

### macOS ###
# General
.DS_Store
.AppleDouble
.LSOVERRIDE

# Icon must end with two \r
Icon

# Thumbnails
.*_

# Files that might appear in the root of a volume
.DocumentRevisions-V100
.fsevents
.Spotlight-V100
.TemporaryItems
.Trashes
.VolumeIcon.icns
.com.apple.timemachine.donotpresent

# Directories potentially created on remote AFP share
.AppleDB
.AppleDesktop
Network Trash Folder
Temporary Items
.apdisk

### macOS Patch ###
# iCloud generated files
*.icloud

# End of https://www.toptal.com/developers/gitignore/api/macos
```


.gitkeep

- 폴더 생성 후 특정 파일 없이 깃에 업로드할 시 사용 (폴더 구조를 잡을 시 주로 사용)

requirements.txt

- 파이썬 패키지 관리를 위한 파일
- 파이썬 패키지와 사용 버전들을 나열

```
backend > requirements.txt
1 asgiref==3.5.2
2 Django==4.1
3 django-cors-headers==3.10.0
4 django-environ==0.9.0
5 djangorestframework==3.12.4
6 djangorestframework-simplejwt==5.2.0
7 mysqlclient==2.1.1
8 PyJWT==2.4.0
9 pytz==2022.2.1
10 sqlparse==0.4.2
11 tzdata==2022.2
12
```

- 파이썬 환경에서 pip freeze > requirements.txt 로 생성
- pip install -r requirements.txt 로 패키지 설치

README.md

- 가이드라인, 안내문, 설명서, 지침서
- 폴더별로 README 마크다운 파일을 작성하면 github에서 기본적으로 하단에 파싱하여 보여줌

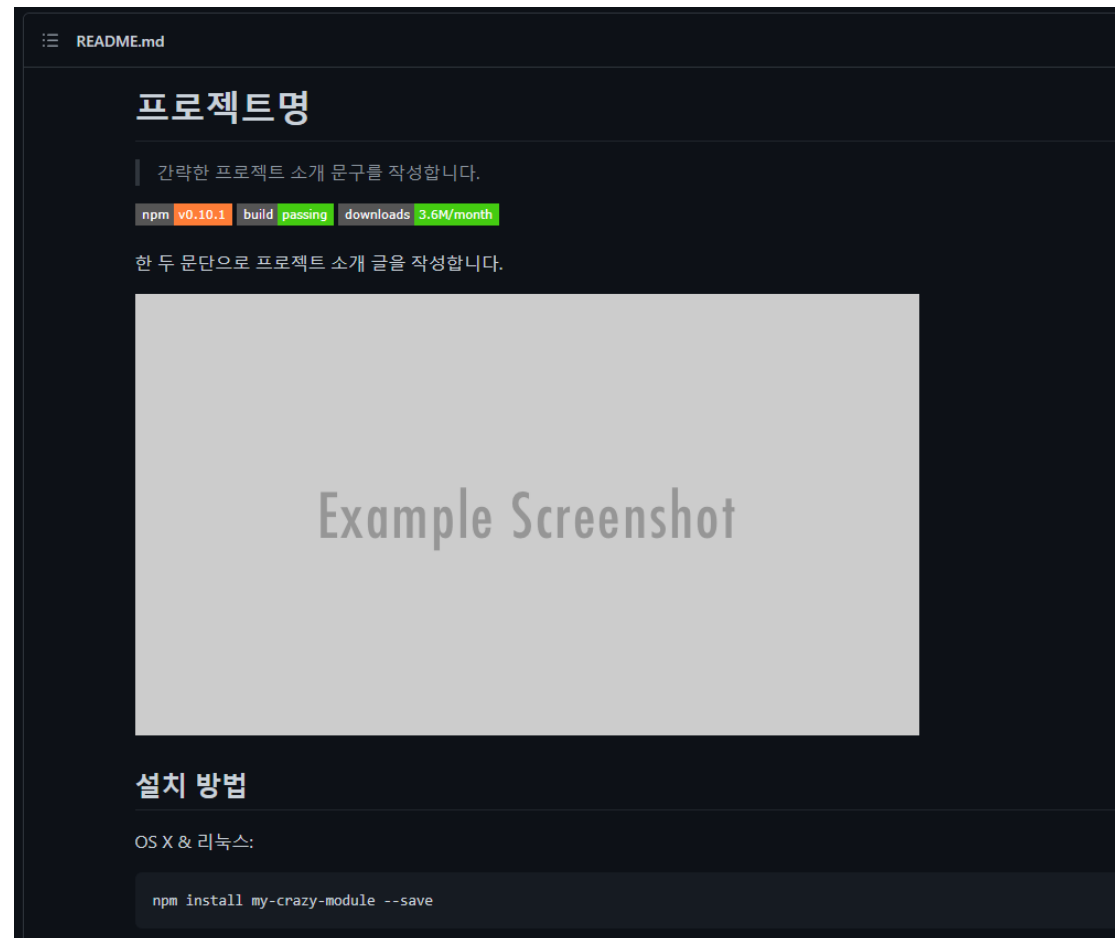


README

README 작성요령

README.md

- [마크다운 작성법](#)
- [깃허브 README 템플릿](#)
- [Awesome README](#)
- <https://shields.io/> (사용 언어, 프레임워크, 기술 배지)
 - 예) <https://img.shields.io/badge/python-3.10-blue>
- <https://www.emojiengine.com/en/> (이모지)



패키지 README 작성법

- 패키지명
- 패키지 한 줄 설명
- Dependency
- 설치
- 사용법
- 패키지 구조 (주요 기능, 패키지, 모듈)

- 개인 패키지는 README를 자유롭게 구성하되, 해당 패키지를 사용하기 용이하도록 작성

- 각 폴더별로 추가 작성
- 모듈 별 상세 설명, 사용방법, 함수, 클래스의 파라미터, 반환되는 값

소신의 금융 데이터 분석 라이브러리

금융 데이터 분석을 위한 라이브러리 입니다.

Dependency

python >=3.6 MariaDB >=10.3 Chrome

설치

패키지 다운로드

```
git clone https://github.com/sesac-finance/sosin_package.git
```

sosin_package 폴더로 이동

```
cd sosin_package
```

의존 라이브러리 설치

```
pip install -r requirements.txt
```

사용법

```
# 전체 패키지 임포트
from sosin_package import sosin

# 사용할 모듈만 임포트
from sosin_package import crawler, rpa, database, processing, rpa, utils
```

패키지 구조

- crawler: 크롤링 패키지
- database: 데이터베이스 관리 패키지
- processing: 데이터 처리 패키지
- rpa: 자동화 관리 패키지
- utils: 유틸 패키지

```
├── crawler
│   ├── news.py
│   ├── blog.py
│   ├── finance.py
│   └── twitter.py
├── database
│   ├── nosql.py
│   └── rdb.py
├── processing
│   ├── text.py
│   ├── data.py
│   └── image.py
├── rpa
│   ├── mail
│   │   └── gmail.py
│   ├── messenger
│   │   └── telegram.py
├── utils
│   └── file_manager.py
└── sosin.py
```