

리서치 코드 품질 관리

☰ 태그

리서치 코드 품질 문제

1. 각자의 컴퓨터에 저장
2. 코드 중복(복붙으로 개발)
 - a. 너무 많은 전역 변수
 - b. 너무 긴 코드
 - c. 꼬인 import
 - d. 추상적인 변수
3. 연구 결과 재현 불가
4. dirty code(하나의 python 파일에 로딩, 전처리, 병렬, 인퍼런스 모두 다....)



깨진 유리창의 법칙 : 한 건물의 유리창이 깨어진 채로 방치되어 있다면, 다른 유리창들도 깨어질 것.

▼ 코드의 재사용성을 증가시키면 → 코드 중복 해결

라이브러리를 통한 추상화


- 컴파일 시간 단축
- 인지 부하 감소
- 인적 오류 감소

린트와 유닛 테스트

린트

소스 코드를 분석하여 의심스럽거나, 에러를 발생하기 쉬운 코드에 표시를 달아놓는 것을 말한다. 원래는 C 용어였으나 일반적으로 사용하게 되었다.

python - flake8, Black 도입, pre-commit & clean code-style 실천하기
PEP 8 기준 python code style tool - Linter로 사용할 수 있는 flake8 그리고 조합이 좋은 black을 살펴보고 pre-commit 까지 설정해보자.

 <https://velog.io/@qlgks1/Python-flake8-Black-도입-clean-code-실천하기>

**Python
Linter**
flake8, black,
pre-commit

black과 flake8을 잘 사용한 예가 있어 가져와봤다.

- flake8: 정적 코드 검사
- black: 코드 수
- mypy: 타입 명시
- 파이썬 인덴트, 네이밍 컨벤션, 카멜 케이스, 스네이크 케이스 체크할 것

CI 지속적 통합

SW 공학에서, CI는 지속적으로 퀄리티 컨트롤을 적용하는 프로세스이다. 작은 단위의 작업, 빈번한 통합을 통해 SW의 질적 향상과 배포 시간 단축에 초점이 맞추어져있다.

- Circle CI
- Jenkins
- **Github Action**