파이썬은 인터프리터 언어

쉽고 쭉쭉 쓸 수 있는데, 컴파일러 없어서 보수/안정성 떨어짐.

파이썬은 실행되다가 오류나면 전까지는 돌아가다가 멈춤

클래스를 쓰는 가장 중요한 이유

1. 코드의 재활용성

반복되는 코드를 최소화 시키는 것이 좋다. 많이 쓸 것들은 클래스에 저장하고 부르면 코드가 짧아지고 재활용성이 높다

1. 모듈화

오류가 뜨면 걔만 보고, 유지 보수가 가능.

클래스 내의 함수와 변수를 가져다 쓰는 것. 클래스 내부의 함수들을 확인하는 방법은 ex) math.\_\_dir\_\_()

**방학 프로젝트 – 메타데이터 정리 자동화 프로그램 작성**

게시글이 올라오면 자동으로 작성되게….!

자동화할 수 있는 부분에 대한 정리(가능 / 불가능),

깃 사용법

1. Repository 만들기
2. 사용자 추가 – setting collaborator 아이디 검색 및 지정
3. 저장소 클론

Console에서 저장할 위치에 git clone 저장소 주소(깃헙 저장소에서 복사)

1. 애드, 커밋, 푸시

Status - 현재 add가 안 된 파일을 빨갛게 띄워서 보여줌

Add – local(내 컴퓨터) unstage와(add안된 것), stage(add된 것)가 존재하는데, unstage에 add가 안된 파일 중 git add \* 이렇게 하면 다 올리고, git add ‘파일명’이러면 하나만 올라가는 것

Commit – git commit -m “메시지” 메시지 적어서 보내거나, 그냥 git commit하면 cmder에서 vim이 떠서 명령어 입력하게 함. Vim은 i를 치면 입력모드, esc한 번 누르고, 쉬프트 + 세미콜론 누르면 커맨드 모드로 감. 여기서 w누르고 엔터 치면 저장. :q누르면 vim 종료

Push – 서버에 내 오리진에 local의 커밋들을 제출. Git push origin “브런치명”

Log – 저장소 변화 보여줌

Remote – remote -v 자기가 가진 서버의 주소들을 보여줌

1. 브런치 파기

Git branch “브런치 명”를 통해서 브런치 만들기

1. 브런치 변경

마스터에서 다른 브런치로 혹은 그 반대로 갈 때

Git checkout “가려는 브런치 명”