클래스 형태로 데코레이터를 만들어서, my\_function에 적용해주는 과제였다. 함수를 호출하는 모양은 똑같지만 verbose라는 이름의 데코레이터가 적용이되어 다른 결과값까지 함께 출력을 해주게 된다. 현재 my\_function은 인자가 있는 함수이다. \_\_\_call\_\_(self, name)으로 객체 자신을 제외한 파라미터를 하나만 추가로 넣어주어도 과제의 문제는 해결할 수 있지만, 데코레이터의 의미는 범용성에서 오는것이라고 생각한다. 따라서, 어떠한 함수에 데코레이터를 입혀도 작동하는 코드를 작성하고 싶었기에, 인자의 종류나 유무에 관계없이 작동할 수 있도록 \_\_call\_\_(self, \*args, \*\*kwargs)로 가변인자를 받도록 코딩을 진행하였다. 이렇게 코드를 짜도 문제의 상황은 당연히 해결이 되고, 더 여러개의 인자가 필요한 함수나, 아예 인자가 필요하지 않은 함수의 경우에도 데코레이터를 적용할 수 있게 되었다.

텍스트이(가) 표시된 사진

자동 생성된 설명