4/6 강의

Class : 객체를 만들어내는 틀의 개념. 생성된 객체간에 형태는 다르지만 기본은 같음 (와플틀 – 여러 종류의 와플, 게임 캐릭터들은 동일한 부류의 스텟을 사용하지만 그 스텟의 값은 캐릭터별로 다르다)

텍스트이(가) 표시된 사진

자동 생성된 설명

함수로는 하나의 프로세스만 실행 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

Class 생성 : \_\_init\_\_ 문법이므로 언더바 양쪽 2개는 고정명 (노란색), self는 class의 고유 객체 개념으로 마찬가지로 문법 취급, 다른 입력은 불가(초록색)

텍스트이(가) 표시된 사진

자동 생성된 설명

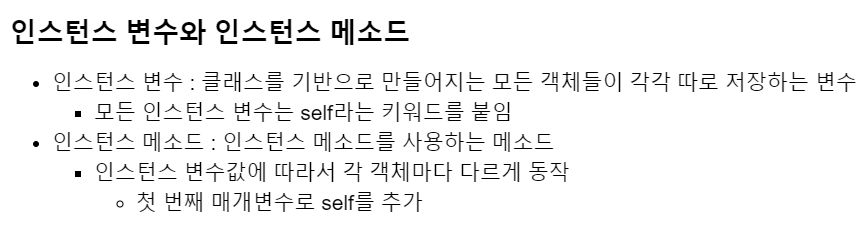
Pass : 아무 기능이 없으나 최소한의 형태만 갖추게 해줌 (입력 안했다는 오류를 안뜨게 해줌)

\_\_main\_\_ : 파이썬이 실행되는 영역 (큰 의미는 없음)

텍스트이(가) 표시된 사진

자동 생성된 설명

Type를 찍으면 어느 class에서 만들어진 객체인지 확인 가능



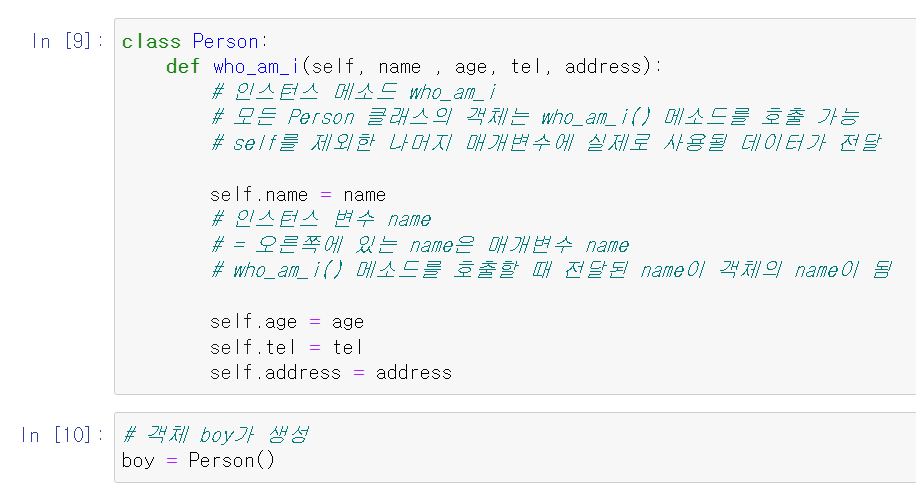
테이블이(가) 표시된 사진

자동 생성된 설명

Cal1, cal2 = 클래스로 만들어진 인스턴스 (= 객체)

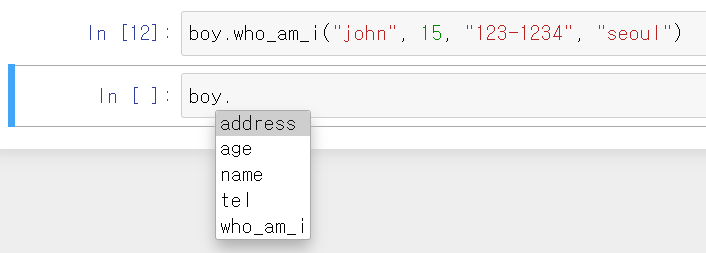
self.result = 인스턴스 변수 (self.는 인스턴스 변수를 선언하는 문법

Add와 같이 (self가 들어간) 인스턴스 변수를 사용하는 함수가 인스턴스 메소드. 첫번째 매개변수로 self가 반드시 들어가야함

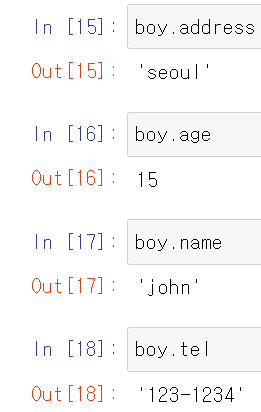


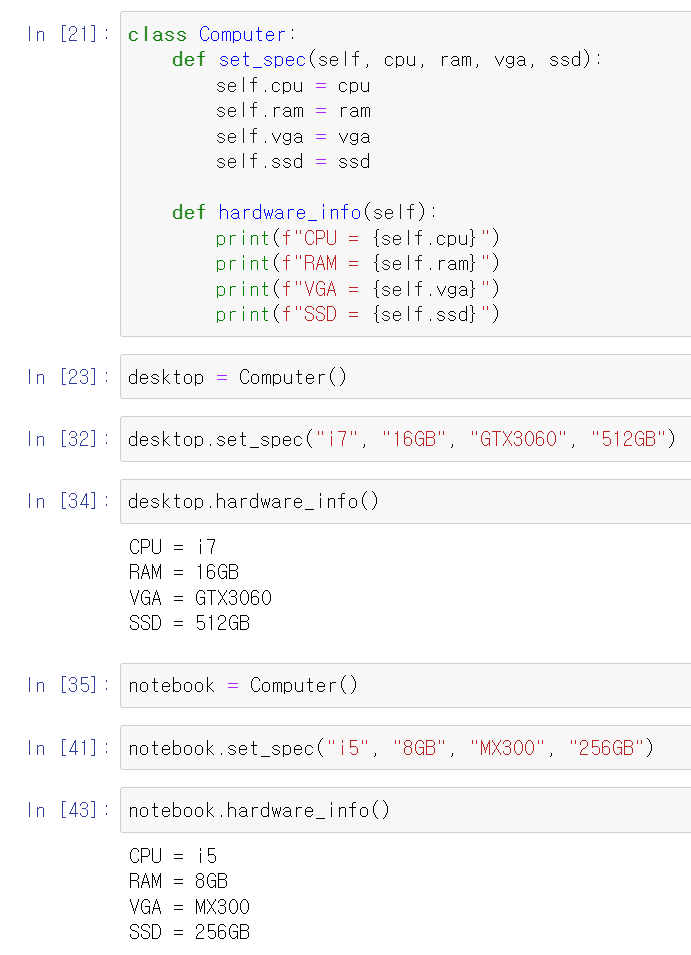
Person 클래스의 boy 객체(인스턴스) 생성

이 단계에서 boy는 who\_am\_i 메소드가 실행되지 않았으므로 name ~ address 변수가 없는 상태



Who\_am\_i 메소드를 실행시켜야 변수가 생김 (객체) + (.) 까지 입력하고 tab을 누르면 해당 객체가 사용할 수 있거나 가지고 있는 메소드와 인스턴스 변수를 선택할 수 있음 (주피터 기능)





객체별로 인스턴스 변수값이 다르므로 같은 메소드를 실행했을 때 나타나는 값도 다르다

텍스트이(가) 표시된 사진

자동 생성된 설명

add 함수의 result는 add 함수의 지역변수

텍스트이(가) 표시된 사진

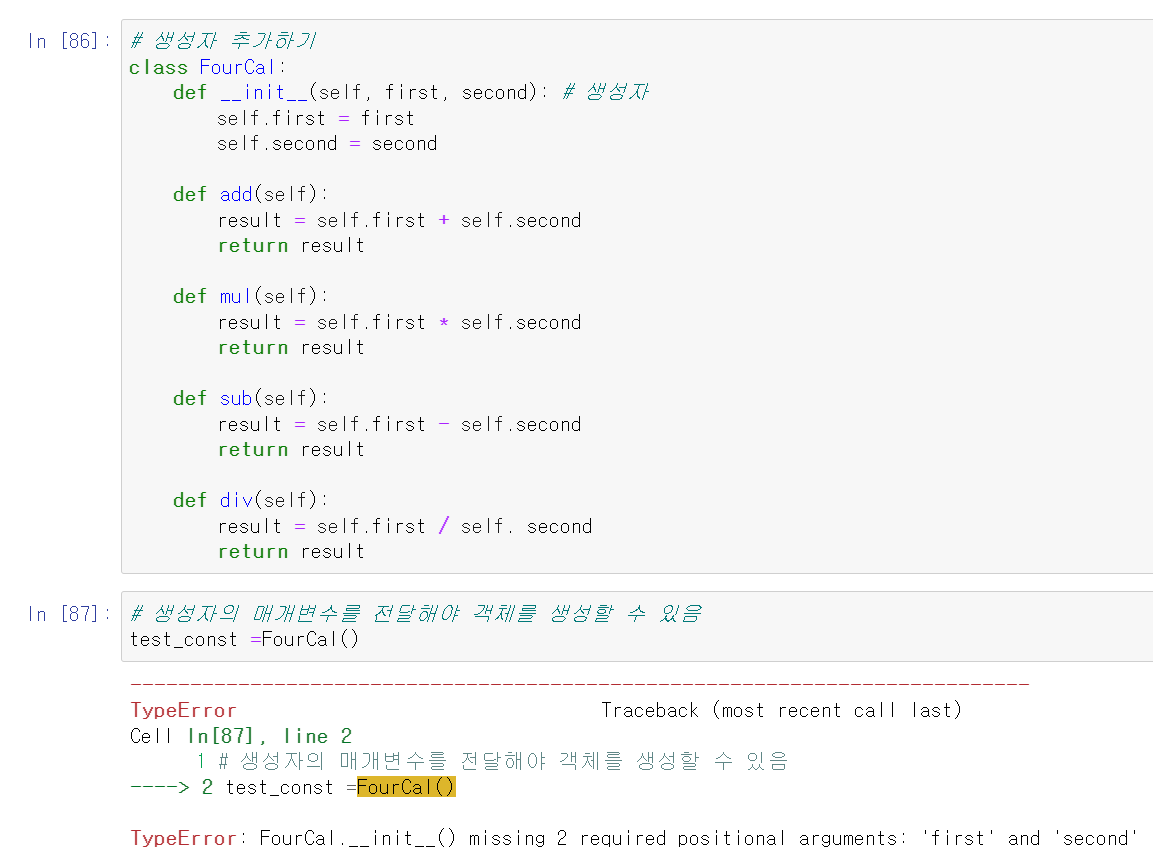
자동 생성된 설명

클래스 내의 지역변수 result는 클래스내 다른 함수나 클래스 외부에서 불러올 수 없다

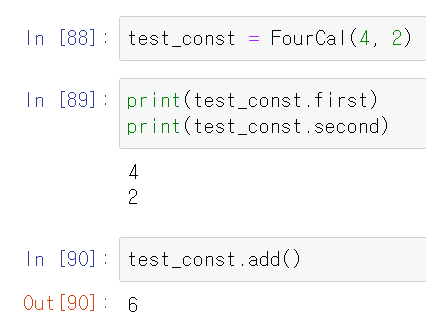
텍스트이(가) 표시된 사진

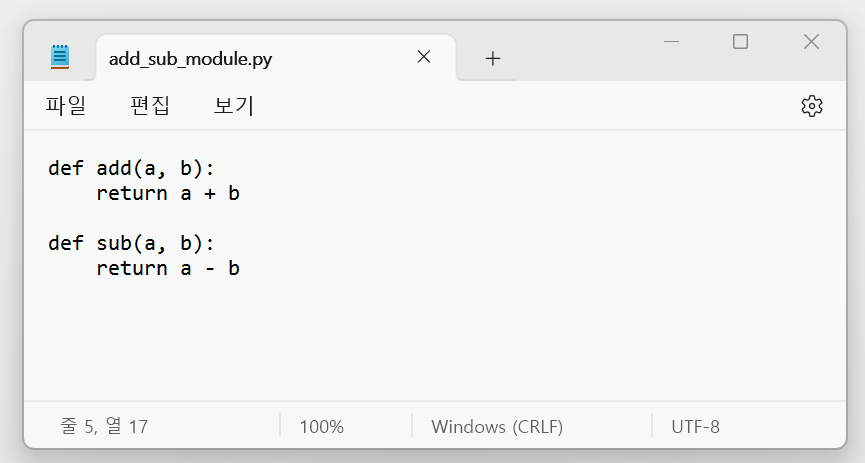
자동 생성된 설명

사칙연산을 수행하는 클래스 생성



생성자 : 객체가 생성될 때 즉시 실행되는 메소드 (\_\_init\_\_ 명의 함수) 즉시 실행되므로, 객체 생성 시에 생성자의 매개변수가 있을 경우 이를 입력해야만 객체가 생성됨





모듈 : 함수나 변수, 클래스를 모아둔 파일 다른 파이썬 프로그램에서 불러와서 사용할 수 있음

텍스트이(가) 표시된 사진

자동 생성된 설명

모듈을 불러와 해당 모듈 내의 함수 실행 from으로 불러오지 않으면 직접 실행은 불가함

텍스트이(가) 표시된 사진

자동 생성된 설명

From 모듈 import 함수를 사용하면 함수를 직접 사용할 수 있다

텍스트이(가) 표시된 사진

자동 생성된 설명

Import \*을 쓰면 모듈 내 모든 함수, 변수 등을 불러옴 (추천은 안함)



모듈명이 길 경우 as로 별명을 붙일수도 있음

텍스트이(가) 표시된 사진

자동 생성된 설명

모듈 내 함수도 별명을 붙일 수 있음