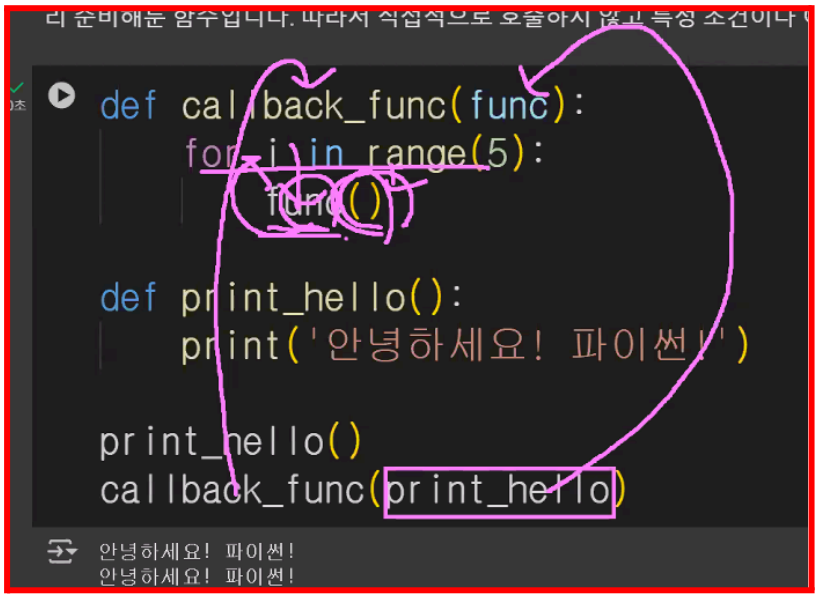
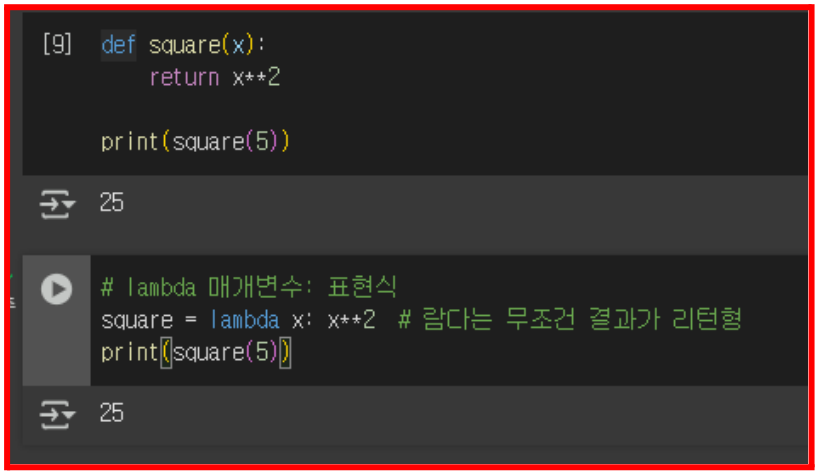
Day 8 - Call back, Lambda function

콜백함수 (callback function)

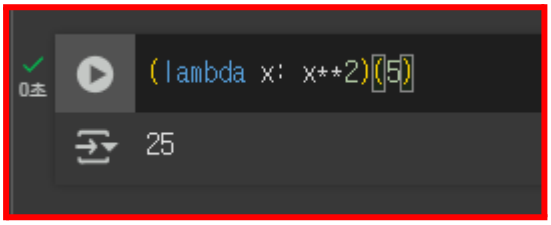


- callback_func에서 print_hello의 주소를 가져와서 그 주소의 함수를 5번 실행시킨다.
- 함수의 매개변수로 함수를 넣을 수 있다.

람다함수 (Lambda function)

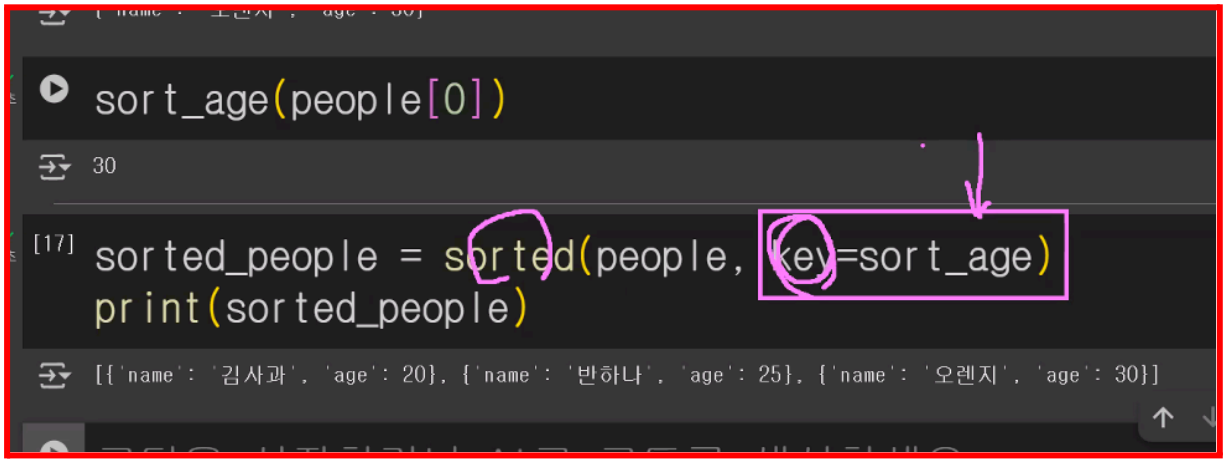


- 결과가 무조건 리턴형이다.
- 변수에 저장이 가능하다.



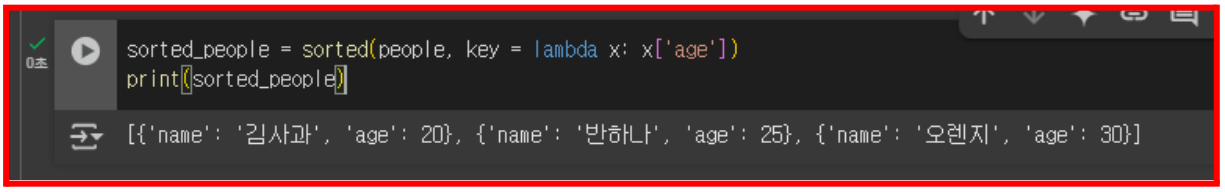
- 이런식으로 일회용으로 계산하는게 가능하다 (메모리를 차지 안한다)

sorted() 사용 풀이



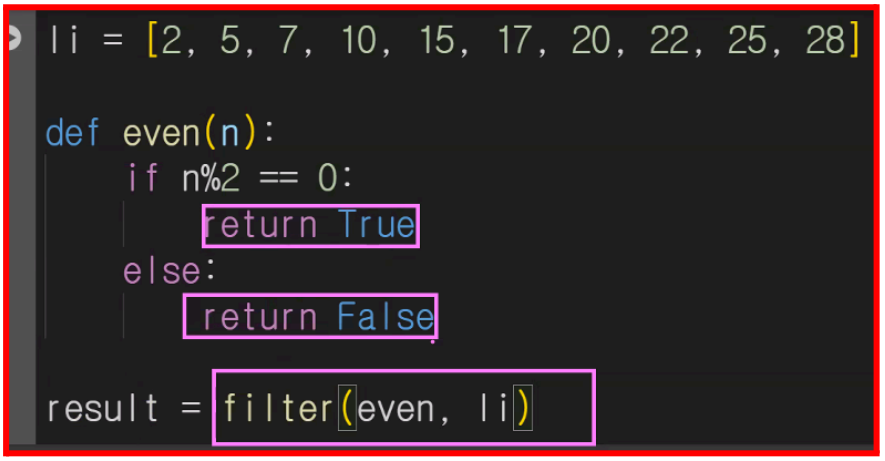
- Key 안에 콜백함수를 넣을 수 있다.
- people 리스트 안에 있는 내용마다 다 실행시켜준다.
- output: age가 낮은순대로 정렬된다.
- 하지만 한번 정렬하는건데 메모리에 계속 살려놔야 한다.

Lambda식 사용 풀이:



- 콜백함수와 다르게 메모리에 남지 않는다.
- 같은 결과가 나온다.

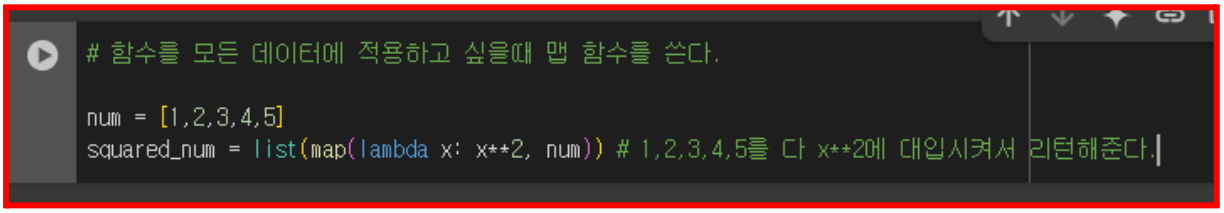
filter 함수:



- 데이터를 내가 원하는것만 딱 골라서 자료구조에 담기위해서 쓴다
- 조건을 함수로 하면 한번만 쓸 거기 때문에 비효율적이다.
- 이터레이터 (반복에 쓸 수 있는 객체)
- output: [2, 10, 20, 22, 28]

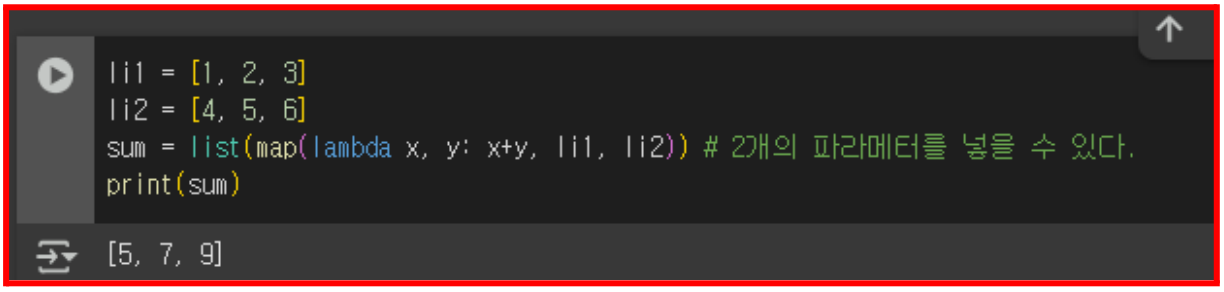
- 람다식으로 조금 더 편하게 사용가능하다.

Map함수:



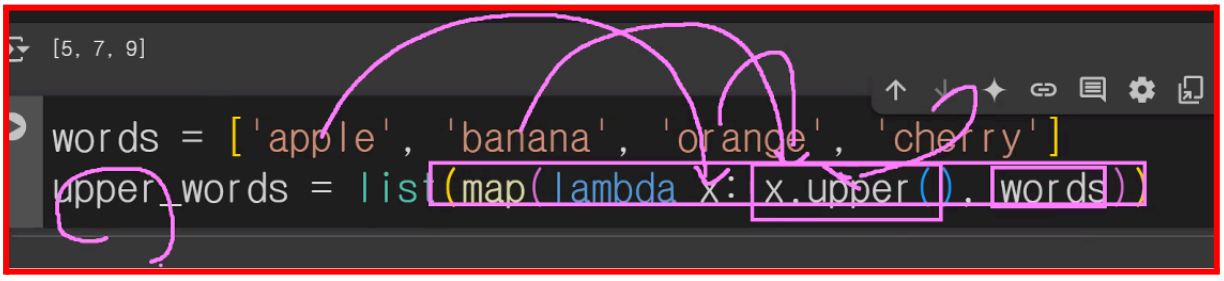
- 주어진 함수를 이터러블의 모든 항목에 적용하여 결과를 반환하는 이터레이터를 생성합니다.
- 주로 리스트나 순차적인 데이터타입에 사용

ex1)



- Map을 이용해서 2개의 파라미터를 넣을 수 있다.

ex2)



- 하나하나 꺼내서 대문자로 바꾼다음 리스트로 만든다.
- ['APPLE', 'BANANA', 'CHERRY']