교육 일지

**2021.9.24 Daily Assignment**

일시: 2021-09-24

장소: 자택

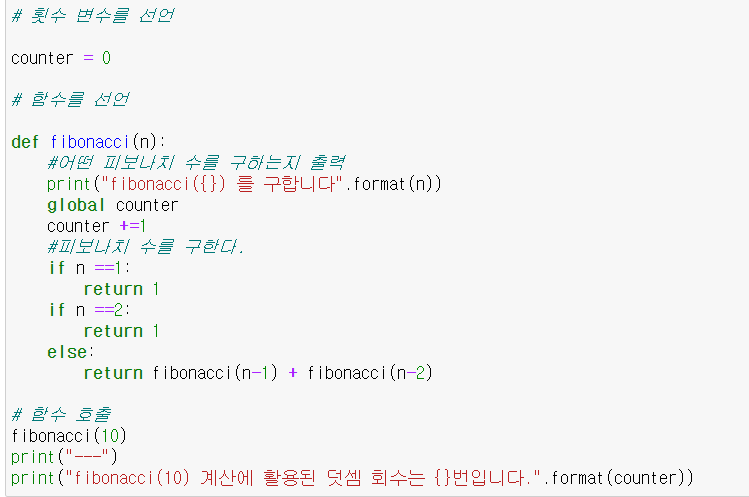
작성자: 강명훈

**오전 교육 내용**

재귀 함수

* 계속해서 같은 함수를 호출하는 것
* 계속해서 같은 것을 반복적으로 호출하여 기하급수적으로 많은 것을 반복 할 수 있음 -> 메모리 오버플로우
* 필요 시 적재적소에 활용하면 간결하게 코드 사용 가능
* 재귀 함수의 문제를 해결하기 위해 memoization 기법이 필요.
* 메모이제이션: 한번 계산한 값을 저장한 후, 계산하는 과정 대신 나중에 이를 다시 활용하는 방법

피보나치 수열 (오로지 재귀만 사용)

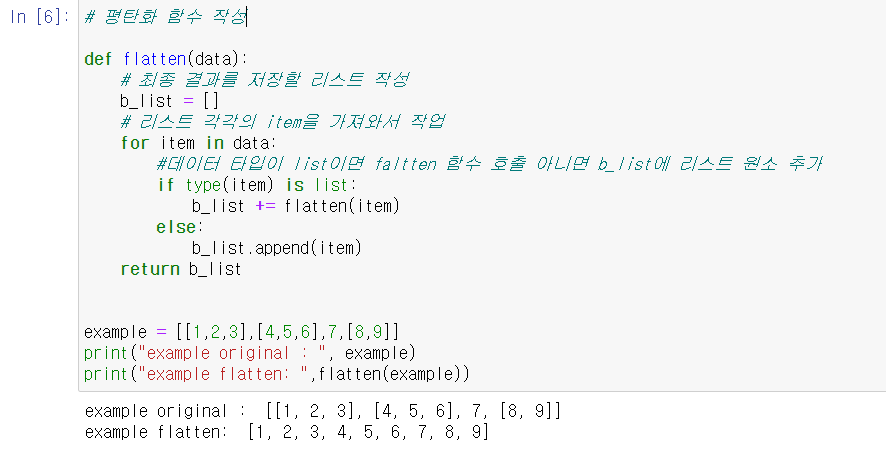


피보나치 수열 (메모이제이션 사용)



Dictionary를 선언하여 한번 계산한 피보나치 수열 내용을 저장하여 추후에 같은 값을 호출할 때 재귀를 통해서 계산하는 것이 아닌 Dictionary에서 호출을 해서 기존에 있는 값을 사용할 수 있게 하여 계산횟수와 반복작업을 줄여 효율적으로 프로그래밍 할 수 있도록 한다.

또 다른 예제



함수 고급

튜플

* 함수와 함께 많이 사용되는 리스트와 비슷한 자료형으로, 한번 결정된 요소를 바꿀 수 없다

람다

* 매개변수로 함수를 전당하기 위해 함수 구문을 작성하는 것이 번거롭고, 코드 공간 낭비라는 생각이 들 때 함수를 간단하고 쉽게 선언하는 방법

Filer() 함수와 map() 함수

* 함수를 매개변수로 전당하는 대표적인 표준 함수
* Map -> map(함수, 리스트) -> 결과 리스트
* Filter – filter(함수, 리스트) -> 결과 리스트

사용 예제



람다 함수는 여러 개의 매개 변수도 사용이 가능하다

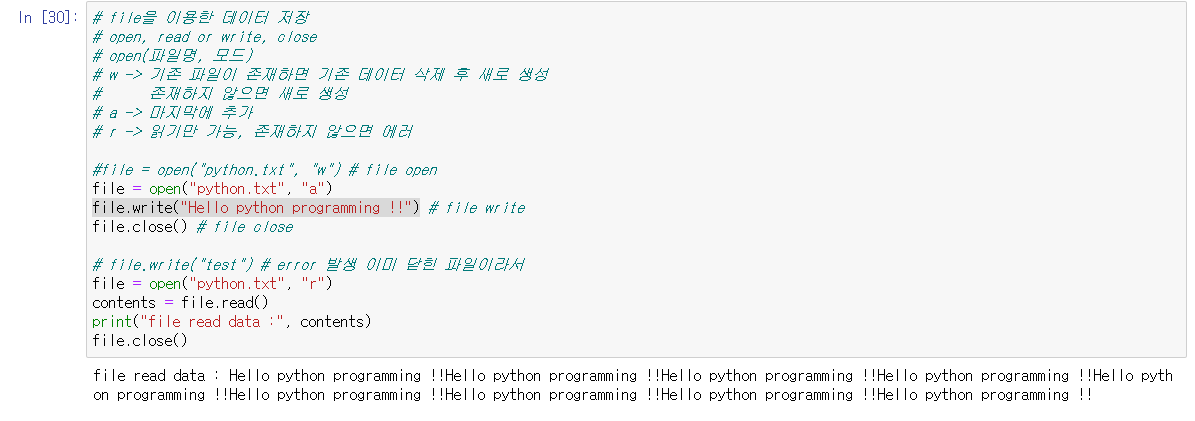
Ex) lambda x,y : x \* y

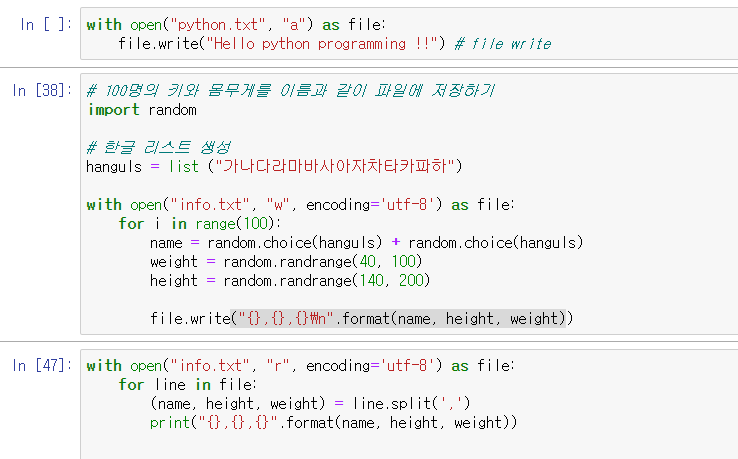
파일 처리

* file을 이용한 데이터 저장
* open, read or write, close

open(파일명, 모드)

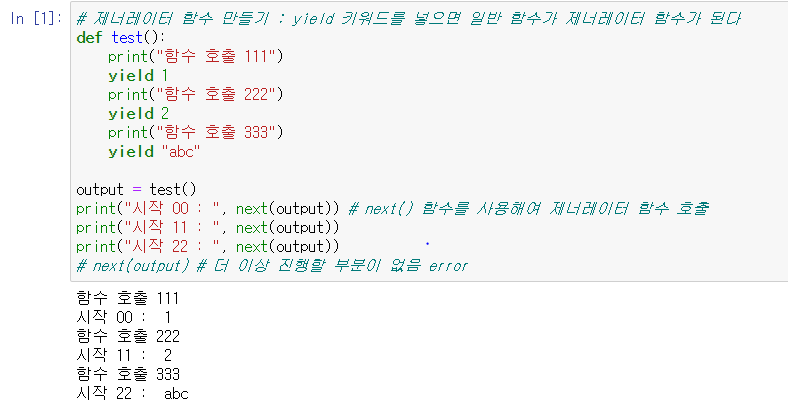
* w -> 기존 파일이 존재하면 기존 데이터 삭제 후 새로 생성, 존재하지 않으면 새로 생성
* a -> 마지막에 추가
* r -> 읽기만 가능, 존재하지 않으면 에러





제너레이터 함수

* 이터레이트를 직접 만들 때 사용하는 코드
* 일반 함수와 달리 함수를 호출해도 내부의 코드가 실행되지 않는다.
* 제네레이터 객체는 next() 함수를 사용해 함수 내부의 코드를 실행, next 함수의 리턴 값으로 yield 뒤의 값이 출력

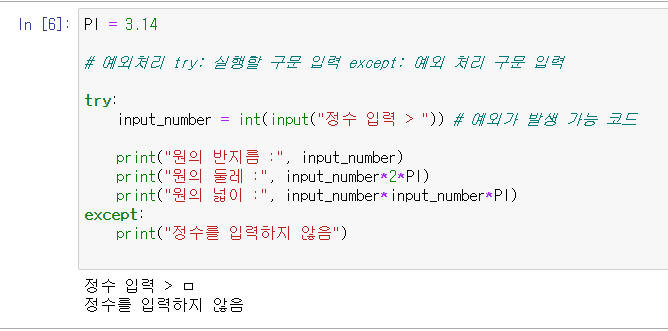


정리 예제 문제



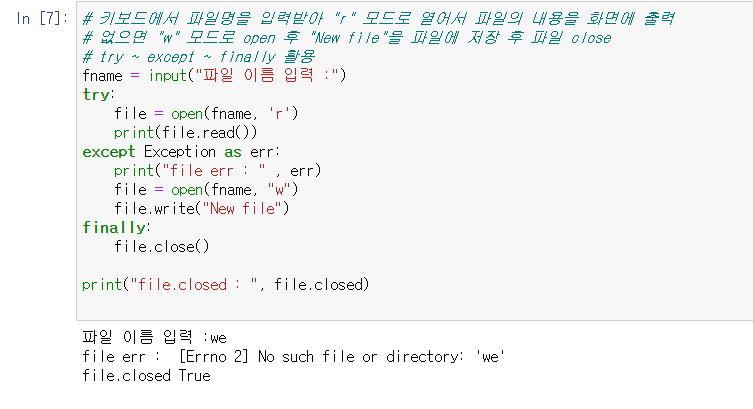
구문 오류와 예외

Try – Except 구문



**오후 교육 내용**

예외처리 + 파일 입출력 응용



다수의 예외처리를 활용 방법

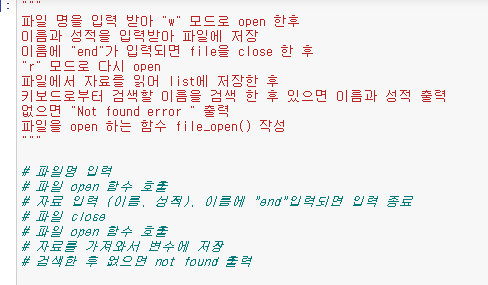


여러 가지 예외가 나올 수 있는 영역을 각각의 영역에 맞는exception으로 처리하고 그 외적인 내용은 단순 Exception으로 처리 할 수 있도록 한다.

Raise

* 아직 구현이 되지 않은 부분에서 문제가 발생 할 수 있음으로 예외를 발생 시켜서 강제종료 시키는 방법이다.

배운 내용 종합 응용 문제



구현

