

Basics: Python

Computer Programming for Engineers (DASF003-41)

Spring, 2022

Assignment: PA1

Instructor:

Youngjoong Ko (nlp.skku.edu)

PA1

- 편의점 아르바이트생이 되어 물건 판매, 재고 관리, 총 매출 확인을 해봅시다.
- 기본 작동 방식 → main함수에 구현됨
 - 파일 입력으로 현재 편의점 재고 상황 & 각 물건 별 가격을 global dictionary "stock", "price_tag"에 저장
 - while문으로 사용자 입력 계속 받음
 - (1: sell things), (2: check stock), (3: check total sales), (4: finish)
 - 숫자 4 입력 받을 경우, 변경된 재고 상황 & 총 매출 파일로 출력하고 프로그램 종료

Main Function

- 'stock', 'sold', 'price_tag' 3개의 global 딕셔너리 값을 초기화
 - read_stock() 함수를 사용하여 "stock", "sold" 딕셔너리에 초기값을 저장합니다.
 - read_price() 함수를 사용하여 "price_tag" 딕셔너리에 초기값을 저장합니다.
- while문으로 사용자 입력 계속 받음
 - global 변수 END가 True일 때 while문이 종료됩니다.

```
def main():  
    read_stock()  
    read_price()  
  
    ##### Write your code #####  
    # END 값이 True가 될 때까지 1, 2, 3, 4 input 반복해서 받기  
  
    #####
```

read_stock function

- " grocery.txt " 파일에서 현재 재고 상황 읽고, global dictionary인 'stock'에 저장
 - 'stock' 은 편의점 재고 상황을 저장 해놓는 딕셔너리입니다.
 - 'stock' 딕셔너리의 (key: value) -> <물건명> : <재고 개수>
 - <물건명>: string 자료형, <재고 개수>: int 자료형
 - 또한 'sold ' 에 물건의 매출 현황을 0으로 초기화해줍니다.
 - 'sold' 딕셔너리의 (key: value) -> <물건명>: <팔린 개수>
 - <물건명>: string 자료형, <팔린 개수>: int 자료형

read_price function

- "price.txt" 파일에서 각 물건 별 가격을 읽고, global dictionary인 'price_tag'에 저장
 - 'price_tag' 은 물건들의 가격표를 저장 해놓는 딕셔너리입니다.
 - 'price_tag' 딕셔너리의 (**key: value**) -> <물건명> : <가격>
 - <물건명>: string 자료형, <가격>: int 자료형

get_input function

- 사용자에게 1, 2, 3, 4 숫자들 중 입력 받음
- 1: **sell things** – 물건 판매 함수 **sell()** 함수 실행
- 2: **check stock** – 재고 관리 함수 **check()** 함수 실행
- 3: **check total sales** – 매출 확인 함수 **sales()** 함수 실행
- 4: **finish** – 프로그램 종료를 위한 함수 **finish()** 함수 실행
- 예외처리: 그 외 입력 받을 경우 다시 입력하도록 함
 - 자연수 1, 2, 3, 4 이외의 입력을 받을 경우 **프롬프트에 “Wrong input.\n”**을 출력합니다.

out_stock function

- "output.txt" 파일에 변경된 재고 상황 & 총 매출 출력
 - 'stock' 딕셔너리에 남아있는 모든 물건들에 대해 "<물건명>: <재고 개수> left."를 출력
 - 물건들 출력 후 개행(줄바꿈)을 하고, 총 매출 출력 "The total sales of the convenience store is <총 매출> won." 출력

##출력 예시##

```
cookie: 10 left.  
water: 10 left.  
mask: 28 left.  
coffee: 8 left.  
kimbap: 3 left.
```

```
The total sales of the convenience store is 6000 won.
```

output.txt에 있어야
하는 내용 예시

sell function

■ (a) 현재 stock 딕셔너리에 있는 물건들을 전부 프롬프트 출력

- “Sales List: <물건명1> <물건명2> <물건명3>....”
- <물건명>들은 space(“ ”)로 구분

■ (b) 입력1: 구매할 물건 ('request' 변수에 할당)

- 예외처리1: 앞에서 출력한 물건 리스트에 없는 물건을 입력할 경우
- 예외처리2: 입력된 물건의 개수 **stock 딕셔너리 재고 상** 개수가 0개일 경우
- 두 경우 모두 (1: sell things), (2: check stock), (3: check total sales), (4: finish) 초기 입력부터 다시 받도록 함

sell function

■ (c) 입력2: 구매할 개수 ('count' 변수에 할당)

- 예외처리1: 입력이 자연수가 아닌 경우
 - 참고: 파이썬 내장함수 `isdigit()` 은 string변수가 전부 숫자로 구성됐을 경우 True를 반환, 반대의 경우 False를 반환
- 예외처리2: 입력이 실제 재고를 초과할 경우
- 두 경우 모두 구매할 개수 입력부터 다시 받도록 함

isdigit() 내장함수 활용 예시

```
>>> "hi".isdigit()  
False  
>>> "1234".isdigit()  
True  
>>> "-12".isdigit()  
False  
>>> "12.3".isdigit()  
False  
>>> num = "100"  
>>> num.isdigit()  
True
```

■ (d) 물건 팔기

- stock 딕셔너리에서는 팔린 물건(key) 개수만큼 **value 빼기**
- sold 딕셔너리에서는 팔린 물건(key) 개수만큼 **value 더하기**
- 손님에게 받을 가격 계산하기. 가격은 'money' 변수에 할당

check function

- (a) 현재 stock 딕셔너리에 있는 물건들 중에 재고가 0개인 물건은 **stock** 딕셔너리에서 **아예 삭제**
 - (key: value) 모두 지우기
 - 재고가 0개인 물건이 **여러 개** 있다면 **모두** 지워줘야 합니다.
- (b) stock 딕셔너리에 남아있는 물건들을 **프롬프트 출력**
 - “<물건명>: <물건 개수>” 형태로 출력
 - (예시)

```
mask: 28
cookie: 13
water: 10
coffee: 8
kimbap: 3
```

sales function

- 매출 확인 함수
- 총 매출 프롬프트 출력

finish function

- out_stock() 함수 실행
- 프로그램 종료

Coding guide

- 모든 코드는 python3 버전으로 작성해주세요.
- pa1.py 코드파일은 주피터 노트북 터미널에서 “**python3 pa1.py**” command로 실행 가능합니다.
- pa1.py에 미리 작성된 코드는 ***절대*** 수정하지 말아주세요.
- **#### Write your code ####** 라고 주석 처리된 부분부터 코드를 작성해주세요.
 - 해당 주석은 지우셔도 무방합니다.
- PA1과 관련된 질문은 구글 스프레드시트(PA1)에 올려주세요.

https://docs.google.com/spreadsheets/d/1Y7WfGSEop1nI_tIGjKrLeYDo8jnv3sEhuA0OsbVs5Y8/edit#gid=331055745

Submission guide

- 예시로 주어진 grocery.txt와 price.txt 파일은 제출하지 않으셔도 됩니다.
- pa1.py 파일을 “student name_id.zip” 로 압축하여 아이캠퍼스로 제출해주세요.
- **제출 기한은 04/22 23:59:59 까지 입니다.**

Example

■ Command 1

☐ 빨간 박스는 사용자 입력

```
Enter number (1: sell things), (2: check stock), (3: check total sales), (4: finish)
1
Sales List: cookie water mask coffee kimbap

What would you like to buy? cookie
How many do you want to buy? 3
The price is 6000 Won.
Thank you for your purchase.
```

```
Enter number (1: sell things), (2: check stock), (3: check total sales), (4: finish)
1
Sales List: cookie water mask coffee kimbap

What would you like to buy? kimbap
How many do you want to buy? 4
We only have 3 kimbap.
How many do you want to buy? one
Tell us a number.
How many do you want to buy? -1
Tell us a number.
How many do you want to buy? 3
The price is 6000 Won.
Thank you for your purchase.
```

<sort dic>

cookie	13-3
water	10
mask	28
coffee	8
kimbap	3

cookie	10
water	10
mask	28
coffee	8
kimbap	3-3

Example

■ Command 1

☐ 빨간 박스는 사용자 입력

```
Enter number (1: sell things), (2: check stock), (3: check total sales), (4: finish)
1
Sales List: cookie water mask coffee kimbap

What would you like to buy? kimbap
The item is out of stock.
```

cookie	10
water	10
mask	28
coffee	8
kimbap	0

```
Enter number (1: sell things), (2: check stock), (3: check total sales), (4: finish)
1
Sales List: cookie water mask coffee kimbap

What would you like to buy? snack
You can only enter items in the list.
```

Example

■ Command 2

☐ 빨간 박스는 사용자 입력

```
Enter number (1: sell things), (2: check stock), (3: check total sales), (4: finish)
2
mask: 28
cookie: 10
water: 10
coffee: 8
```

cookie	10
water	10
mask	28
coffee	8
kimbap	0

■ Command 3

```
Enter number (1: sell things), (2: check stock), (3: check total sales), (4: finish)
3
The total sales of the convenience store is 12000 Won.
```


Example

■ Command 4 빨간 박스는 사용자 입력

```
Enter number (1: sell things), (2: check stock), (3: check total sales), (4: finish)
4
I'm done with my work!
```

■ 종료 후 output.txt 파일 생성

```
≡ output.txt ×
≡ output.txt
1  cookie: 10 left.
2  water: 10 left.
3  mask: 28 left.
4  coffee: 8 left.
5
6  The total sales of the convenience store is 12000 Won.
```

■ Wrong Command

```
Enter number (1: sell things), (2: check stock), (3: check total sales), (4: finish)
5
Wrong input.
```