

Fastcampus Data Science Extension SCHOOL

Python

Index

- Review
- File I/O
- Error Handle
- Packages
- work with vs code

Do it yourself

```
recycle_bin = [  
1, 2, "Fastcampus", ['dog', 'cat', 'pig'], 5, 4, 5.6, False,  
"패스트캠퍼스", 100, 3.14, 2.71828, {'name': 'Kim'}, True,  
]
```

1. 위 리스트의 요소 중 정수와 실수인 요소만 리스트로 구성하기
2. 위 리스트의 요소 중 정수만 각각 제공하여 리스트로 구성하기
3. 위 리스트의 요소 중 정수만 각각 제공한 수들의 합계 출력하기

Hint: `isinstance(1, int)`

Answer

1. 위 리스트의 요소 중 정수와 실수인 요소만 리스트로 구성하기

```
list(filter(lambda a: not isinstance(a,bool) and isinstance(a,(int,float)), recycle_bin ))
```

2. 위 리스트의 요소 중 정수만 각각 제공하여 리스트로 구성하기

```
list(map(lambda a:a**2, filter(lambda a: not isinstance(a,bool) and isinstance(a,int), recycle_bin )))
```

3. 위 리스트의 요소 중 정수만 각각 제공한 수들의 합계 출력하기

```
reduce(lambda a,b:a+b, map(lambda a:a**2, filter(lambda a: not isinstance(a,bool) and isinstance(a,int), recycle_bin )))
```

Do it yourself

Order ID	Quantity	Unit Price
181121001	2	2400
181121002	12	9800
181121003	5	124800
181121004	10	76000
181121005	24	2810

- 앞서 배운 개념들을 활용해 아래 문제를 해결하세요.
 1. 튜플과 리스트를 활용해 위 테이블을 변환하세요
 2. 각 주문 별 총 주문가격을 산출하세요
 3. 11월 21일에 발생한 총 매출, 평균 구매금액을 산출하세요

Answer

1. 튜플과 리스트를 활용해 위 테이블을 변환하세요

```
orders = [(181121001, 2, 2400), (181121002, 12, 9800),  
(181121003, 5, 124800), (181121004, 10, 76000), (181121005, 24,  
2810)]
```

2. 각 주문 별 총 주문가격을 산출하세요

```
list(map(lambda a:a[1]*a[2], orders))
```

3. 11월 21일에 발생한 총 매출, 평균 구매금액을 산출하세요

```
reduce(lambda a,b:a+b, map(lambda a:a[1]*a[2], orders))
```

```
reduce(lambda a,b:a+b, map(lambda a:a[1]*a[2], orders)) /  
len(orders)
```

File I/O

File I/O

```
f = open(filename, mode)
f.close()
```

- mode
 - r - 읽기모드
 - w - 쓰기모드
 - a - 추가모드(파일의 마지막에 새로운 내용을 추가)

Create New File

```
f = open("Newfile.txt", 'w')  
f.close()
```

Write text

```
f = open("Newfile.txt", 'a')
for i in range(1,11):
    text = "line %d. \n" % i
    f.write(text)
f.close()
```

Read text

```
f = open("Newfile.txt", 'r')
text = f.readline()
print(text)
f.close()
```

Read All text

```
f = open("Newfile.txt", 'r')
while True:
    text = f.readline()
    if not text: break
    print(text)
f.close()
```

Read All text using readlines

```
f = open("Newfile.txt", 'r')
texts = f.readlines()
for text in texts:
    print(texts)
f.close()
```

Add text

```
f = open("Newfile.txt", 'a')
for i in range(11, 20):
    text = "New line %d \n" % i
    f.write(text)
f.close()
```

Get rid of f.close()

```
with open("foo.txt", 'w') as f:  
    f.write("foo is text dummy")
```

Do these with CSV format

file I/O with json

```
import json
```


dump(dictionary to json)

```
with open(' ', 'w') as f:  
    json.dump({1:1, 2:2}, f)
```

load(json to dictionary)

```
with open(' ', 'r') as f:  
    json.load(f)
```

do it yourself!

`best_boxing_movies.csv` 파일의 모든 텍스트를 읽어 리스트로 구성한 뒤,
`best_boxing_moviex.json` 과 같은 형태가 되도록 json 포맷의 파일을 생성하세요.

Error Handle - Try Except

Error Handle

by using `try, except`

필요한 만큼만 적절히 사용하셔야 합니다 by PEP 8

Error Handle - Syntax

```
try:
    실행문
except:
    실행문
```

Error Handle - ValueError

```
try:
    some_input = int(input("type some number: "))
except ValueError:
    print("I said type some NUMBER!!!!")
```

Error Handle - ValueError

```
try:
    some_input = int(input("type some number: "))
except ValueError as e:
    print("I said type some NUMBER!!!!")
    print(e)
```


Error Handle - FileNotFoundError

```
try:
    f = open('error_example.txt', 'r')
except FileNotFoundError as e:
    print(e)
else:
    text = f.read()
    f.close()
```

Error Handle - Multiple Error

```
try:
    ...
except error type 1:
    ...
except error type 2:
    ...
```

Error Handle - Pass Error

```
try:
    f = open('error_example.txt', 'r')
except FileNotFoundError as e:
    pass
else:
    text = f.read()
    f.close()
```

Finally!

```
try:
    f = open('error_example.txt', 'r')
except FileNotFoundError as e:
    print("Oops!")
    pass
else:
    text = f.read()
    f.close()
finally:
    print("어쨌거나 끝났습니다")
```

Do it yourself!

임의의 숫자(1~1000 사이의 정수) 두 개 로 이루어진 100개의 tuple 을 csv 파일로 저장한 뒤,

이를 불러와 곱셈 연산을 수행 하여 새로운 파일에 두 수와 곱셈 결과를 다시 csv 파일로 작성하는 파일을 작성하세요.

(단, 파일을 불러올 때 try except를 적용하여, FileNotFoundError가 발생했을 시 에러메시지만 출력한 뒤, pass하세요.)

Python Packages

PIP(Package Installer for Python)

Python의 외부 패키지 설치를 위한 관리자

install and uninstall

```
$ pip install <package>  
$ pip uninstall <package>
```

list outdated and upgrade

```
$ pip list --outdated  
$ pip install --upgrade <package>
```


output packages and import output

```
$ pip freeze  
$ pip freeze > requirements.txt
```

install all requirements

```
$ pip install -r requirements.txt
```

search package

```
$ pip search <package>
```

Do it yourself!

1. pandas 패키지를 설치하세요
2. requirements.txt에 numpy를 입력하여 저장하세요
3. requirements.txt를 활용해 패키지를 인스톨하세요
4. 현재 인스톨된 패키지를 requirements.txt에 입력하세요
5. requirements.txt에 beautifulsoup4를 추가한 뒤, 다시 인스톨하세요
6. 현재 인스톨된 패키지를 다시 requirements.txt에 입력하세요

Let's install VS Code

<https://code.visualstudio.com/>

Set configuration

Install Extensions

work with vs code