PYTHON SESSION

By Edu
@ 이영준 고병욱

Date _ 2018.03.24

CONTENTS

01 Python 기초

Pythonic한 문법

Python의 장점

단순 명료하고 가독성이 좋음

→ Pythonic code

01 Python 기초

Pythonic한 문법

if 문 예시(일반 코드)

```
signal = True
if signal == True:
        print(“Hi”)
else:
        print(“Bye”)
```

예시 바꾸기
하나 넣을 수 있는거

01 Python 기초

Pythonic한 문법

if 문 예시(축약 코드)

```
signal = True
print("Hi") if signal else print("Bye")
```

01 Python 기초

Pythonic한 문법

리스트 할당 예시(일반 코드)

```
temp = []
for i in range(10):
        _temp.append(i)
print(_temp)
```

출력값: [0,1,2,3,4,5,6,7,8,9]

01 Python 기초

Pythonic한 문법

리스트 할당 예시(축약 코드)

```
temp = [i for i in range(10)]
print(temp)
```

출력값: [0,1,2,3,4,5,6,7,8,9]

01 Python 기초

Pythonic한 문법

리스트 할당 예시(조건문 활용)

```
temp = [i for i in range(10) if i%2 == 0]
print(temp)
```

출력값: [0,2,4,6,8]

01 Python 기초

연습문제

아래 코드를 축약하여 작성해보세요

```
temp = []
for i in range(10):
        for j in range(10):
                if i%3 == 0 and j%3 == 0:
                        temp.append((i,j))
print(temp)
```

01 Python 기초

함수

함수란?

입력값을 받고 함수 내에서 정의된 규칙 수행해서 얻은 결과값을 출력하는 것

함수를 사용하는 이유

똑같은 코드를 반복적으로 사용할 때 이를 효율적으로 관리하기 위함

01 Python 기초

함수

함수 정의하기

```
def 함수명(입력 인자):
    (실행)
    return 결과값
```

* 인자: 함수로 전달한 값

코드 예시

```
def cal(a,b):
    return a * b
```

01 Python 기초

함수

함수 호출하기

함수로 인자를 전달

해당 인자가 매개변수에 복사됨

코드 예시

```
result = cal(2,3)
print(result)
```

출력값: 6

01 Python 기초

함수

입력값, 출력값 유무에 따른 함수 구분

1. 입력값, 출력값이 존재

```
def cal(a,b):
        return a * b
```

2. 입력값만 존재

```
def cal(a,b):
        print(a * b)
```

2번 함수 결과값

cal(2,3) → 6

print(cal2,3) → 6, None

01 Python 기초

함수

입력값, 출력값 유무에 따른 함수 구분

3. 출력값만 존재

```
def cal():
        return 777
```

4. 입력값 출력값 모두 존재하지 않음

```
def cal():
        print(777)
```

01 Python 기초

함수

입력값의 개수를 모른다면?

```
def 함수명(*args):
        (실행1)
        return
```

*args: arguments의 약자이며 보통 *args라고 사용

* 함수 내에서 arg는 튜플 자료형

01 Python 기초

함수

입력값의 개수를 모른다면?

코드 예시

```
a, b, c = "종윤", "상연", "건우"
def call_name(*args):
        for person in args:
                print(person)

call_name(a,b,c)
```

출력값: 종윤
상연
건우

01 Python 기초

함수

키워드 파라미터

```
def 함수명(**kwargs):
        (실행1)
        (실행2)
```

각 인자와 대응되는 값이 딕셔너리 자료형으로 kwargs에 할당

*kwargs: keyword arguments의 약자 **kwargs 라고 사용

01 Python 기초

함수

키워드 파라미터

코드 예시

```
def call_name(**kwargs):
        return kwargs

print(call_name₩
(종윤="남", 상연="남", 건우="남", 수진="여"))
```

출력값: {'종윤':'남', '상연':'남', '건우':'남', '수진':'여'}

01 Python 기초

함수

여러 값 Return

여러 값이 반환되는 것이 아니라 튜플으로 반환됨

```
def mul_return():
        return 1,2,3

_type = type(mul_return())
print(_type)
```

출력값: <class 'tuple'>

01 Python 기초

함수

여러 값 Return

```
result = mul_return()
print(result)

a,b,c = mul_return()
print(a,b,c)
```

출력값: (1,2,3)
1 2 3

01 Python 기초

함수

Return의 다른 용도

반복문의 break 처럼 특정 조건이 만족되었을 때 함수를 빠져나가기 위해 사용

01 Python 기초

함수

입력 인수 초깃값 설정

```
def 함수명(인자1=..., 인자2=...):
        (실행1)
        return
```

*초깃값을 설정한 인수는 가장 뒤에 있어야 함

01 Python 기초

함수

입력 인수 초깃값 설정

```
def _sum(a, b, 출력여부=True):
        if 출력여부 == True:
                print(a+b)
                return a + b
        else:
                return a + b
```

_sum(5,3) → 모니터에 8 출력
_sum(5,3, False) → 모니터에 출력x

01 Python 기초

함수

함수 내부 변수

함수 내부의 변수는 독립적으로 존재

함수 외부의 변수와 이름이 같더라도 함수 내부의 결과는 외부 변수에 영향을 미치지 않음

01 Python 기초

함수

함수 내부 변수

```
a = 0
def test(a):
        for i in range(5):
                a += 1
        print(a)
```

test(a) → 5를 출력
print(a) → 0을 출력

01 Python 기초

함수

함수 내부 변수

함수로 외부 변수 바꾸기 → return 활용

```
a = 0
def test(a):
        for i in range(5):
                a += 1
        print(a)

a = test(a)
print(a)
```

→ 출력값: 5

* glob 사용

01 Python 기초

내장함수

enumerate()

반복 가능한 자료형을 인자로 받음

인자(iterable)의 인덱스와 원소의 쌍을 리턴

```
for i, value in enumerate([종윤, 상연, 건우]):
    print(i, value)
```

출력값: 0 종윤
1 상연
2 건우

01 Python 기초

내장함수

lambda()

lambda(인자, 함수내용)

간결하게 함수를 정의할 때 사용

```
mul = lambda x,y: x*y
print(mul(5,5))
```

출력값: 25

01 Python 기초

내장함수

lambda()

def를 사용할 수 없는 곳에 사용 (ex. pandas)

```
df['이름_점수'] = df[['이름','성적']].apply₩
                  (lambda x: x[0]+x[1], axis=1)
```

01 Python 기초

내장함수

zip()

같은 개수를 지닌 자료형을 인자로 받아 차례대로 묶어줌

```
print(list(zip([1,2,3], [4,5,6])))
```

출력값: [(1, 4), (2, 5), (3, 6)]

01 Python 기초

파일 입력

open(“파일명”, “파일열기모드 ")

파일을 열 때 사용되는 내장함수

파일 경로 및 이름, 파일 열기 모드를 인자로 받음

close()로 파일을 항상 닫아주어야 함

01 Python 기초

파일 입력

파일열기모드의 종류

"r": 파일을 읽을 때 사용

"w": 파일에 내용을 쓸 때 사용

"a": 파일에 내용을 추가할 때 사용

코드 예시

```
f = open("C:\Users\LYJ\example.txt, "r")
```

01 Python 기초

파일 입력

write() 함수

출력값을 파일에 써주는 함수

print()는 모니터에, write()는 파일에 출력

코드 예시

```
f = open("C:\Users\LYJ\example.txt, "w")
```

01 Python 기초

파일 출력

readline() 함수

open으로 파일을 열고 나서 사용

파일의 첫 번째 줄을 읽어 출력

반복문을 사용해 모든 줄을 읽어올 수 있음

문자열 자료형을 리턴함

01 Python 기초

파일 출력

readline() 함수 코드 예시

```
f = open("C:WWUsersWWLYJWWEample.txt", 'r')
line = f.readline()
print(line)
f.close()
```

type(line) : <class 'str'>

print(line) : ,carat, depth, table, price, x, y, z

01 Python 기초

파일 출력

readlines() 함수

open으로 파일을 열고 나서 사용

파일의 모든 라인을 읽어서 출력함

리스트 자료형을 리턴함

01 Python 기초

파일 출력

readlines() 함수 코드 예시

```
f = open("C:₩₩Users₩₩LYJ₩₩Eample.txt", 'r')
lines = f.readlines()
for line in lines:
        print(line)
f.close()
```

type(lines) : <class 'list'>

print(lines) : [',carat,depth,table,price,x,y,z₩n',...]

01 Python 기초

파일 출력

with문

with문을 사용하면 close()를 할 필요가 없음

들여쓰기로 파일이 open되어 있는 구간을 구분함

코드 예시

```
with open((“C:₩₩Users₩₩Ex.txt”, ‘r’) as f:
        lines = f.readlines()
```

01 Python 기초

예외 처리

대표적인 오류들

IndexError: 참조할 수 없는 값을 인덱싱 했을 때

TypeError: 잘못된 자료형을 사용했을 때

SyntaxError: 구문이 잘못되었을 때

그 외 다양한 에러 존재

→ 구글링

01 Python 기초

예외 처리

try, except 구문

try 블록을 실행 중 에러가 발생하면 except 블록의 코드를 실행함

에러를 직접 핸들링하고 싶을 때 사용

ex) 실시간으로 데이터를 처리할 때

01 Python 기초

예외 처리

try, except 구문(기본형)

```
try:
        (실행1)
        (실행2)

except:
        (실행1')
        (실행2')
```

* 들여쓰기에 신경써야
* 모든 오류를 커버

01 Python 기초

예외 처리

try, except 구문(오류 특정)

```
try:
        (실행1)
        (실행2)

except 오류 이름:
        (실행1')
        (실행2')
```

* 해당 오류일때만 except실행

01 Python 기초

예외 처리

try, except 구문(오류 특정/메시지 변수)

```
try:
        (실행1)
        (실행2)

except 오류 이름 as 메시지 변수:
        (실행1')
        (실행2')
```

* 해당 오류일때만 except실행

01 Python 기초

예외 처리

try, except 구문(오류 특정/메시지 변수)

```
try:
        print(“Hi”)

except IndexError as i:
        print(i)
```

출력값: list assignment index out of range

01 Python 기초

예외 처리

try, except, else 구문

```
try:
        print(_list[0:4])

except IndexError as i:
        print(i)

else:
        print("Hello")
```

* 예외가 발생하지 않으면 else 블록 실행

01 Python 기초

예외 처리

try, finally 구문

```
try:
        print("Hi")

finally:
        f.close()
```

* finally 블록은 오류에 관계 없이 항상 실행됨

01 Python 기초

연습문제

연습문제

주어진 파일의 열을 뒤집은 파일을 바탕화면에 생성해주세요

1. 경로나 파일 명이 틀렸다면 “파일 경로 및 이름을 확인해주세요”를 출력해주세요
2. 함수를 작성해서 문제를 풀어주세요
3. 함수의 인자를 파일명으로 해주세요
4. 출력 파일의 확장자 명은 csv로 지정해주세요
5. 파일 생성이 끝나면 “생성 완료 " 를 출력해주세요