

# Python Homework 5

## 231 ~ 240

---

### 231

아래 코드를 실행한 결과를 예상하라.

```
def n_plus_1 (n) :  
    result = n + 1  
  
n_plus_1(3)  
print (result)
```

### 232

문자열 하나를 입력받아 인터넷 주소를 반환하는 make\_url 함수를 정의하라.

```
make_url("naver")  
  
www.naver.com
```

### 233

문자열을 입력받아 각 문자들로 구성된 리스트로 반환하는 make\_list 함수를 정의하라.

```
make_list("abcd")  
  
['a', 'b', 'c', 'd']
```

## 234

숫자로 구성된 하나의 리스트를 입력받아, 짝수들을 추출하여 리스트로 반환하는 pickup\_even 함수를 구현하라.

```
pickup_even([3, 4, 5, 6, 7, 8])  
  
[4, 6, 8]
```

## 235

콤마가 포함된 문자열 숫자를 입력받아 정수로 변환하는 convert\_int 함수를 정의하라.

```
convert_int("1,234,567")  
  
1234567
```

## 236

아래 코드의 실행 결과를 예측하라.

```
def 함수(num) :  
    return num + 4  
  
a = 함수(10)  
  
b = 함수(a)  
  
c = 함수(b)  
  
print(c)
```

## 237

아래 코드의 실행 결과를 예측하라.

```
def 함수(num) :  
    return num + 4  
  
c = 함수(함수(함수(10)))  
print(c)
```

## 238

아래 코드의 실행 결과를 예측하라.

```
def 함수 1(num) :  
    return num + 4  
  
def 함수 2(num) :  
    return num * 10  
  
a = 함수 1(10)  
c = 함수 2(a)  
print(c)
```

239

아래 코드의 실행 결과를 예측하라.

```
def 함수 1(num) :  
    return num + 4  
  
def 함수 2(num) :  
    num = num + 2  
    return 함수 1(num)  
  
c = 함수 2(10)  
print(c)
```

240

아래 코드의 실행 결과를 예측하라.

```
def 함수 0(num) :  
    return num * 2  
  
def 함수 1(num) :  
    return 함수 0(num + 2)  
  
def 함수 2(num) :
```

```
num = num + 10

return 함수 1(num)

c = 함수 2(2)

print(c)
```

## 291 ~ 300

---

### 291 파일 쓰기

바탕화면에 '매수종목 1.txt' 파일을 생성한 후 다음과 같이 종목코드를 파일에 써보세요.

```
005930
005380
035420
```

### 292 파일 쓰기

바탕화면에 '매수종목 2.txt' 파일을 생성한 후 다음과 같이 종목코드와 종목명을 파일에 써보세요.

```
005930 삼성전자
005380 현대차
035420 NAVER
```

## 293 CSV 파일 쓰기

바탕화면에 '매수종목.csv' 파일을 생성한 후 다음과 같이 종목코드와 종목명을 파일에 써보세요. 인코딩은 'cp949'를 사용해야합니다.

	A	B	C	
1	종목명	종목코드	PER	
2	삼성전자	005930	15.79	
3	NAVER	035420	55.82	

## 294 파일 읽기

바탕화면에 생성한 '매수종목 1.txt' 파일을 읽은 후 종목코드를 리스트에 저장해보세요.

005930

005380

035420

## 295 파일 읽기

바탕화면에 생성한 '매수종목 2.txt' 파일을 읽은 후 종목코드와 종목명을 딕셔너리로 저장해보세요. 종목명을 key 로 종목명을 value 로 저장합니다.

005930 삼성전자

005380 현대차

## 296 예외처리

문자열 PER (Price to Earning Ratio) 값을 실수로 변환할 때 에러가 발생합니다. 예외처리를 통해 에러가 발생하는 PER 은 0 으로 출력하세요.

```
per = ["10.31", "", "8.00"]
```

```
for i in per:  
    print(float(i))
```

## 297 예외처리 및 리스트에 저장

문자열로 표현된 PER 값을 실수로 변환한 후 이를 새로운 리스트에 저장해보세요.

```
per = ["10.31", "", "8.00"]
```

```
for i in per:  
    print(float(per))
```

## 298 특정 예외만 처리하기

어떤 값을 0 으로 나누면 ZeroDivisionError 에러가 발생합니다. try ~ except 로 모든 에러에 대해 예외처리하지 말고 ZeroDivisionError 에러만 예외처리해보세요.

## 299 예외의 메시지 출력하기

다음과 같은 코드 구조를 사용하면 예외 발생 시 에러 메시지를 변수로 바인딩할 수 있습니다.

```
try:
```

```
    실행코드
```

```
except 예외 as 변수:
```

```
    예외처리코드
```

리스트의 인덱싱에 대해 에러를 출력해보세요.

```
data = [1, 2, 3]
```

```
for i in range(5)
```

```
    print(data[i])
```

## 300 try, except, else, finally 구조 사용해보기

파이썬 예외처리는 다음과 같은 구조를 가질 수 있습니다.

```
try:
```

```
    실행 코드
```

```
except:
```

```
    예외가 발생했을 때 수행할 코드
```

```
else:
```

```
    예외가 발생하지 않았을 때 수행할 코드
```



### **finally:**

예외 발생 여부와 상관없이 항상 수행할 코드

아래의 코드에 대해서 예외처리를 사용하고 try, except, else, finally 에 적당한 코드를 작성해봅시다. else 와 finally 는 적당한 문구를 print 하시면 됩니다.

```
per = ["10.31", "", "8.00"]
```

```
for i in per:
```

```
    print(float(per))
```