

# 1월13일실습\_1

## [ 실습 1 ]

1. 파일명 : compreLab1.py

2. 구현해야 하는 함수 사양

함수명 : createList

매개변수 : 0개 이상의 위치 아규먼트를 받는 매개변수 1개 -->가변인수

기본값이 있는 매개변수 1개(매개변수명은 **type**이며 기본값은 1이다.)

리턴값 : 1개

기능 :

**0개 이상의 위치 아규먼트를 가지고 아래에 제시된 타입에 따른 리스트를 생성하여 리턴한다.**

위치 아규먼트가 전달되지 않은 경우에는 **1부터 30의 값**을 가지고 **type**에 따른 리스트를 생성하여 리턴한다.

type 이 2이면 데이터 값들에서 **짝수**에 해당하는 데이터들만을 가지고 리스트 생성

type 이 3이면 데이터 값들에서 **홀수**에 해당하는 데이터들만을 가지고 리스트 생성

type 이 4이면 데이터 값들에서 **10보다 큰 데이터들**만을 가지고 리스트 생성

type 이 1이면 **데이터 값들을 모두 가지고** 리스트 생성

리스트 생성은 **리스트 컴프리헨션(지능형 리스트) 구문**을 사용한다.

3. 다양한 구성으로 아규먼트를 전달하면서 createList() 함수를 호출하고 리턴 결과를 화면에 출력한다.

```
def createList(*p, type=1):
    if p :
        if type == 2:
            result = [i for i in p if i % 2 ==0]
        elif type == 3:
            result = [i for i in p if not i % 2 ==0]
        elif type == 4:
            result = [i for i in p if i > 10]
        elif type == 1:
            result = [i for i in p]
    else:
        result = [i for i in range(1, 31)]

    return result

print(createList(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,type=1))
print(createList(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,type=2))
print(createList(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,type=3))
print(createList(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,type=4))
print(createList( ))
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]

[2, 4, 6, 8, 10, 12, 14, 16]

[1, 3, 5, 7, 9, 11, 13, 15]

[11, 12, 13, 14, 15, 16]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]

## [ 실습 2 ]

1. 파일명 : compreLab2.py
2. 구현해야 하는 함수 사양  
함수명 : mycompredict  
매개변수 : **가변 키워드형**(0 개 이상의 키=값 형식의 아규먼트를 받아서 처리한다.)  
리턴값 : 1개  
기능 : funcLab11.py 에서 구현한 mydict() 라는 함수의 기능과 동일하게 구현하는데, 이번에는 **딕셔너리 컴프리헨션(지능형 딕셔너리) 구문**을 사용해서 생성한다.
3. 다양한 구성으로 가변 키워드 아규먼트를 전달하면서 mycompredict() 함수를 호출하고, 리턴 결과를 화면에 출력한다.

```
def mydict (**kwargs):  
    dic = {'my'+ kw : args for kw, args in kwargs.items()}  
    return dic  
  
print(mydict(Korean=50, Math=100, Science=30))  
print(mydict())
```

```
{'myKorean': 50, 'myMath': 100, 'myScience': 30}  
{}
```

## [ 실습 3 ]

1. 파일명 : strLab1.py
2. 구현해야 하는 함수 사양  
함수명 : myemail\_info  
매개변수 : **이메일 주소 문자열 1개**  
리턴값 : **2개의 원소를 갖는 튜플**  
기능 : 전달된 이메일 주소 문자열에서 @ 를 기준으로 쪼개서 튜플을 만들어 리턴한다.  
단, 전달된 문자열에 @가 없으면 **None**을 리턴한다.
3. 아규먼트를 전달하면서 myemail\_info() 함수를 여러 번 호출하고, 리턴 결과를 화면에 출력한다.

```
def myemail_info(email):  
    if '@' in email:  
        ad = tuple(email.split('@'))  
        return ad  
    else :  
        return None  
  
print(myemail_info('1234@naver.com'))  
print(myemail_info('abcd456@naver.com'))  
print(myemail_info('1234naver.com'))
```

```
('1234', 'naver.com')  
('abcd456', 'naver.com')  
None
```

## [ 실습 4 ] 문자열 관련 실습

소스명 : strLab2.py

- (1) s1 = "pythonjavascript" 에서의 길이를 출력한다.
- (2) s2 = "010-7777-9999" 에서 01077779999 을 출력한다.
- (3) s3 = "PYTHON" 에서 **NOHTYP** 를 출력한다.
- (4) s4 = "Python" 에서 **python** 을 출력한다.
- (5) s5 = "<https://www.python.org/>" 에서 [www.python.org](https://www.python.org/) 만을 뽑아서 출력한다.
- (6) 주민등록 번호에서 **7자리 숫자를 사용해서 남성, 여성을 판별**한다. (1,3 : 남성, 2,4 : 여성)  
s6 = '891022-2473837'
- (7) s7 = '100' 에서 s7 값을 정수형 숫자로 그리고 실수형 숫자로 변환하여 출력한다.
- (8) s8 = ' The Zen of Python' 에서 'n' 문자가 몇 개인지 출력한다.
- (9) s8 에서 'Z' 의 위치를 출력한다.
- (10) s8 에서 모두 대문자로 변환한 후 공백단위로 분리해서 리스트로 만들어 출력한다.

```
s1 = "pythonjavascript"
s2 = "010-7777-9999"
s3 = "PYTHON"
s4 = "Python"
s5 = "https://www.python.org/"
s6 = '891022-2473837'
s7 = '100'
s8 = ' The Zen of Python'

print(len(s1))
a = s2.split('-')
print(''.join(a))
print(s3[::-1])
print(s4.lower())
i = s5.split('/')
print(i[2])

def gender(a):
    if s6[7] == '1' or s6[7] == '3':
        sex='남성'
    elif s6[7]== '2' or s6[7] == '4':
        sex='여성'
    return sex
print(gender(s6))

print(int(s7))
print(float(s7))
print(s8.count('n'))
print(s8.index('Z'))
print((s8.upper()).split(' '))
```

16

01077779999

NOHTYP

python

[www.python.org](https://www.python.org/)

여성

100

100.0

2

5

['', 'THE', 'ZEN', 'OF', 'PYTHON']

## [ 실습 5 ]

파일명 : compreLab3.py

아래 리스트 항목 중에서 **소문자만 추출해서 리스트를 생성하여**

listv2에 저장하고 출력한다.(리스트 컴프리헨션 사용)

listv1 = ["A", "b", "c", "D", "e", "F", "G", "h"]

```
listv1 = ["A", "b", "c", "D", "e", "F", "G", "h"]
result = [i for i in listv1 if i.islower()]
print(result)
```

['b', 'c', 'e', 'h']

## [ 실습 6 ]

파일명 : packunpacLab.py

다음 리스트를 생성하고

listv3 = ['p', 'y', 't', 'h', 'o', 'n']

(1) v1, v2, v3, v4, v5, v6 에 **언패킹해서 저장한 후에 각 변수의 값을 행 단위로 화면에 출력한다.**

(2) listv3 를 **언패킹**하여 print() 함수에 전달하여 출력한다.

(3) listv3 를 그냥 print() 함수에 전달하여 출력한다.

```
listv3 = ['p', 'y', 't', 'h', 'o', 'n']
v1, v2, v3, v4, v5, v6 = listv3
print(v1)
print(v2)
print(v3)
print(v4)
print(v5)

print(*listv3)
print(listv3)
```

p  
y  
t  
h  
o  
python  
['p', 'y', 't', 'h', 'o', 'n']

## [ 실습 7 ]

파일명 : compreLab4.py

**컴프리헨션 문법을 사용**해서 다음에 제시된 데이터들로 구성되는 자료구조를 생성한다.

(1) 난수 추출 함수를 사용하여 **0 부터 100 사이의 값 10개로 구성되는 리스트**를 하나 생성한다.

(2) 위에서 생성된 리스트를 이용하여 다음과 같이 구성되는 **딕셔너리**를 생성한다.

(추출된 점수가 **60점 이상이면 pass, 60점 미만이면 nopass** 로 처리한다.)

```
import random
randlist = [random.randint(0,100) for _ in range(10)]
print(randlist)

dic_p_np = { (randlist.index(e)+1): 'pass' if e > 60 else 'nopass' for e in
randlist}
print(dic_p_np)
```

[26, 35, 73, 54, 36, 75, 64, 57, 0, 32]

{1: 'nopass', 2: 'nopass', 3: 'pass', 4: 'nopass', 5: 'nopass', 6: 'pass', 7: 'pass', 8: 'nopass', 9: 'nopass', 10: 'nopass'}

```
# 난수 추출 함수를 사용하여 0부터 100까지의 값 10개로 구성되는 리스트의 생성
rotto = []

import random
for i in range(0,11):
    rand1 = random.randint(1,100)
    rotto.append(rand1)

# 리스트를 이용하여 딕셔너리를 생성 (60 이상이면 pass, 미만이면 nopass)
dic1 = {i: 'pass' if i>=60 else 'nopass' for i in rotto}
print(dic1)
```

{10: 'nopass', 66: 'pass', 97: 'pass', 98: 'pass', 47: 'nopass', 12: 'nopass', 42: 'nopass', 44: 'nopass', 46: 'nopass', 61: 'pass'}

```
import random
n = []
for i in range(10):
    n.append(random.randint(0, 100))
print(n)
d = {i + 1: 'pass' if n[i] >= 60 else 'nopass' for i in range(10)}
print(d)
```

[45, 48, 21, 82, 91, 24, 95, 71, 30, 26]

{1: 'nopass', 2: 'nopass', 3: 'nopass', 4: 'pass', 5: 'pass', 6: 'nopass', 7: 'pass', 8: 'pass', 9: 'nopass', 10: 'nopass'}

```
import random

i = random.randint(0,100)
list_num = [random.randint(0,100) for i in range(10)]

dic_num = {i : 'pass' if i > 60 else 'nopass' for i in list_num}

print(list_num)
print(dic_num)
```

[27, 20, 89, 40, 4, 75, 42, 72, 45, 42]

{27: 'nopass', 20: 'nopass', 89: 'pass', 40: 'nopass', 4: 'nopass', 75: 'pass', 42: 'nopass', 72: 'pass', 45: 'nopass'}

