

# 과목평가 대비 문제

## 1,2번. List Slice

- 다음의 코드를 print 한 결과를 작성하시오

```
aList = [1,2,3,4,5,6,7,8,9,10]

# 참고 : a[10]은 IndexError

(1)aList[2:10]

(1-2)aList[2:12]

(2)aList[:5]

(3)aList[5:]

(4)aList[0:8:2]

(5)aList[:8:2]

(6)aList[::2]

(7)aList[5::2]

(8)aList[:]

(9)aList[2:2]

(10)aList[:-2]

(11)aList[-4:]

(12)aList[::-1]

(13)aList[::-2]

(14)aList[::-3]
```

## 3번. 내장함수 - 줄맞춤

- 다음의 코드를 print 한 결과를 작성하시오

```

str = 'abc'

(1)str.ljust(8, '+')

(2)str.rjust(8, '+')

(3)str.rjust(8, 'lm') # hint: 에러가 날까 안날까?

(4)str.center(9, '+')

# -----
str2 = '-123'

(5)str2.rjust(8, '0')

(5-2)str2.zfill(8)) # 보너스 문제
# 답: -0000123
# 오.른.쪽 정렬을 하면서 0을 채워주는 메서드로 zfill() 이 있다
# rjust() 와 zfill() 메서드 차이점은 + - 기호를 포함해서 인식하는지 차이가 있다
# 금액 표시같은 경우에는 zfill()를 사용하지만 우편번호나 전화번호등,
# 문자열 그대로 사용하고 싶은 경우에는 rjust()를 사용하는 것이 좋다

```

## 4번. 논리 연산자

- 다음의 코드를 print 한 결과를 작성하시오

```

(1)a = 5 and 4

(2)b = 5 or 3

(3)c = 0 and 5

(4)d = 5 or 0

(5)e = 0 or 1

(6)f = '' or 2

```

## 5번. Global , Local 변수

- 다음 중, 에러가 발생하는 코드를 모두 고르시오

```

# (1)
a = 10
def func1():

```

```

global a
a = 3

print(a)
func1()
print(a)

-----

# (2)
a = 10
def func1():
    print(a)
    global a
    a = 3

print(a)
func1()
print(a)

-----

# (3)

a = 10
def func1():
    global a
    a = 3

print(a)
func1(3)
print(a)

-----

# (4)

def ham():
    a = 'spam'
    return a

ham()
print(a)

```

## 6번. Positional, Keyword Packing

- 다음의 코드를 print 한 결과를 작성하시오

```

(1)
def add(*args):
    print(args, type(args))

print(add(1, 2, 3))
print(add(1))

```

```
(2)
def family(**kwargs):
    print(kwargs, type(kwargs))

family(father = '고길동', son='둘리')
```

## 7번. Join메서드에 대한 이해

- 다음의 코드를 print 한 결과를 작성하시오

```
numbers = [1,2,3]
words = ['안녕', 'hello']

(1)'!'.join('ssafy')

(2)' '.join(['3', '5'])

(3)numbers.join(' ')

(4)' '.join(numbers)

(5)words[0].join(words[1])

(6)"".join(words)
```

## 8번. 얇은 복사, 깊은 복사

- 다음의 코드를 print 한 결과를 작성하시오

```
(1)할당

original_list = [1, 2, 3]
copy_list = original_list
copy_list[0] = 'hello'
print(original_list == copy_list)

-----

(2)얇은 복사1

a = [1, 2, ['a', 'b']]
b = a[:]
print(a, b)
b[2][0] = 0
print(a, b)
```

```

-----

(3)얕은 복사2
original_list = [1, 2, [0, 1]]
copy_list = original_list[:]

copy_list[2] = 'h'
print(copy_list, original_list)

-----

(4)얕은 복사3
original_list = [1, 2, {'a': 'apple'}]
copy_list = original_list[:]

copy_list[2]['a'] = 'h'
print(copy_list, original_list)

-----

(5)깊은 복사
import copy

a = [1, 2, ['a', 'b']]
b = copy.deepcopy(a)
print(a, b)
b[2][0] = 0
print(a, b)

```

## 9번. 재귀기초(복습)

- 다음의 코드를 print 한 결과를 작성하시오

```

(1)
def bbq(level):

    print(level)
    if level == 2:
        return

    bbq(level+1)
    print(level)

bbq(0)

-----

(2)
def bbq(level):

    if level == 5:
        print(level)
        return

```

```

    print(level)
    bbq(level+1)
    print(level)

bbq(1)

-----
(3)
def bbq(level):

    if level == 5:
        print(level)
        print(level)
        return

    print(level)
    bbq(level+1)
    print(level)

bbq(1)

```

## 10번. 상속관련 메서드 - Super()

- name은 가져오고 싶고, age는 가져오기 싫을 때, Super()메서드를 사용할 수 있는가?
- 아래 코드를 살펴보고, 에러가 뜨는지 안 뜨는지 답하시오.

```

class Person:
    def __init__(self, name, age, number, email):
        self.name = name
        self.age = age
        self.number = number
        self.email = email

class Student(Person):
    def __init__(self, name, age, number, email, student_id):
        super().__init__(name, number, email)
        self.age = age
        self.student_id = student_id

```