

# 월말평가 - Python

## 평가 안내

- 제공되는 `.py` 파일에 답안을 작성한 후, 폴더 이름을 본인의 `(지역)_(반)_(이름)` 으로 변경하고 압축하여 제출합니다.
  - 예를 들어, 서울 지역 1반 학생의 이름이 '홍길동'인 경우 폴더명을 `서울_1반_홍길동` 으로 변경하고 압축합니다.
  - 압축을 풀었을 때, 폴더 구조는 아래와 같아야 합니다.
    - 서울\_1반\_홍길동/
      - 01.py
      - 02.py
      - 03.py
      - 04.py
      - 05.py

## 문제 1. 나누어 떨어지는 숫자들 (15점) - 01.py

- 리스트 `numbers`에 담겨있는 숫자들 중, 주어진 `divisor`로 나누어 떨어지는 숫자들을 오름차순으로 정렬한 리스트를 반환하는 `can_divide` 함수를 작성하시오.
- `divisor`로 나누어 떨어지는 숫자가 하나도 없다면 리스트에 `-1` 을 담아 반환하시오.

```
def can_divide(numbers, divisor):  
    """  
    아래에 코드를 작성하시오.  
    numbers는 0이 아닌 양의 정수가 담긴 리스트입니다.  
    divisor는 0이 아닌 양의 정수입니다.  
    numbers에 담겨있는 숫자들 중, divisor로 나누어 떨어지는 숫자들을 오름차순으로 정렬한 리스트를  
    반환합니다.  
    """
```

```
# 예시 입력 및 출력  
print(can_divide([20, 3, 5, 7], 5))  
print(can_divide([4, 3, 2, 1], 1))  
print(can_divide([7, 11, 13], 3))
```

```
[5, 20]  
[1, 2, 3, 4]  
[-1]
```

## 문제 2. 알파벳 카운트 (15점) - 02.py

- 문자열에서 나온 알파벳과 알파벳 갯수를 딕셔너리로 반환하는 `alphabet_count` 함수를 작성하시오.

```
def alphabet_count(word):  
    """  
    아래에 코드를 작성하시오.  
    word는 소문자로만 구성되어 있습니다.  
    알파벳을 key로, 갯수를 value로 가지는 딕셔너리를 반환합니다.  
    """
```

```
# 예시 입력 및 출력  
print(alphabet_count('hello'))  
print(alphabet_count('internationalization'))  
print(alphabet_count('haha'))
```

```
{'h': 1, 'e': 1, 'l': 2, 'o': 1}  
{ 'i': 4, 'n': 4, 't': 3, 'e': 1, 'r': 1, 'a': 3, 'o': 2, 'l': 1, 'z': 1}  
{ 'h': 2, 'a': 2}
```

## 문제 3. 문자열 요약하기 (20점) - 03.py

- 반복되는 알파벳으로 이루어진 문자열을 해당 알파벳과 그 갯수로 이루어진 문자열로 요약하는 함수 `summary` 함수를 작성하시오.

```
def summary(word):  
    """  
    아래에 코드를 작성하시오.  
    word는 알파벳으로 구성되어 있습니다.  
    요약된 문자열을 반환합니다.  
    """
```

```
# 예시 입력 및 출력  
print(summary('aabbbaacc'))  
print(summary('ffgggeeeef'))  
print(summary('abcdefg'))
```

```
a2b2a2c2  
f2g3e4f1  
a1b1c1d1e1f1g1
```

# 문제 4. 나만의 단어장 (20점) - 04.py

- Python 클래스를 활용하여 나만의 단어장을 만드시오.
- `Word` 클래스에 대한 명세는 다음과 같습니다.

인스턴스 변수	타입
wordbook	dict

메소드명	인자	반환값(return)	비고
(생성자)	없음	없음	인스턴스가 생성 될 때, 인스턴스 변수 <code>wordbook</code> 을 빈 딕셔너리로 초기화 합니다.
add	영문, 한글	없음	영문, 한글을 딕셔너리 wordbook에 각각 key와 value로 추가합니다.
delete	영문	True/False	딕셔너리 wordbook의 key에 해당 영문이 존재하는 경우, 딕셔너리 wordbook에서 삭제 후 True를 반환하며, 없는 경우에는 False를 반환합니다.
print	없음	없음	딕셔너리 wordbook을 아래 출력 예시에 맞춰 출력합니다.

# 아래의 코드를 실행하였을 때 아래와 같이 출력되어야 합니다.

```
my_book = Word()  
my_book.add('grape', '포도')  
my_book.add('peach', '복숭아')  
my_book.add('watermelon', '수박')  
my_book.add('mango', '망고')  
my_book.print()  
print(my_book.delete('watermelon'))  
print(my_book.delete('mango'))  
print(my_book.delete('carrot'))  
my_book.print()
```

```
grape: 포도  
peach: 복숭아  
watermelon: 수박  
mango: 망고  
True  
True  
False  
grape: 포도  
peach: 복숭아
```

# 문제 5. 도형 만들기 (30점) - 05.py

- Python 클래스를 활용하여 꼭지점과 사각형을 표현하시오.
- `Point` 클래스에 대한 명세는 다음과 같습니다.

인스턴스 변수	타입	비고
x	int	x 좌표
y	int	y 좌표

메소드명	인자	반환값(return)	비고
(생성자)	x 좌표, y 좌표	없음	인스턴스가 생성 될 때, 인스턴스 변수 <code>x</code> 와 <code>y</code> 를 입력받은 integer로 초기화 합니다.

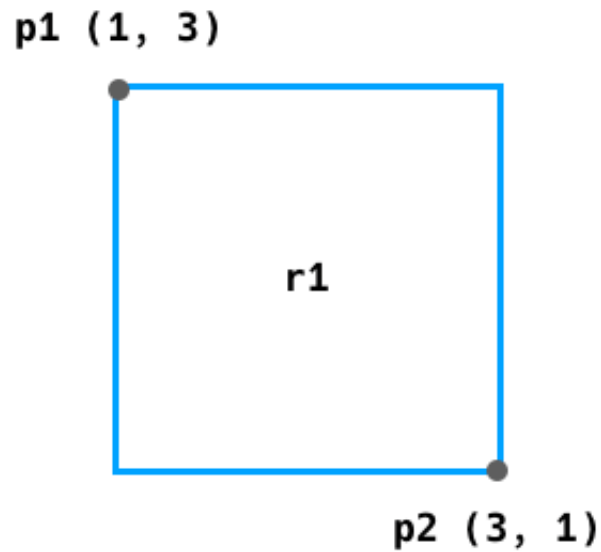
```
# 예) 좌표 (4, 3)의 점
p1 = Point(4, 3)
```

- `Rectangle` 클래스에 대한 명세는 다음과 같습니다.

인스턴스변수	타입	비고
p1	Point 클래스 인스턴스	좌측 상단 좌표
p2	Point 클래스 인스턴스	우측 하단 좌표

메소드명	인자	반환값 (return)	비고
(생성자)	Point 클래스 인스턴스, Point 클래스 인스턴스	없음	인스턴스가 생성 될 때, 인스턴스 변수 <code>p1</code> 과 <code>p2</code> 를 입력받은 Point 클래스 인스턴스들(좌측 상단 좌표, 우측 하단 좌표)로 초기화 합니다.
get_area	없음	면적	면적을 계산하여 반환합니다.
get_perimeter	없음	둘레값	둘레를 계산하여 반환합니다.
is_square	없음	True/False	정사각형인지 아닌지를 판단하여 True/False를 반환합니다.

아래의 예시의 r1은 다음과 같이 구성됩니다.



# 아래의 코드를 실행하였을 때 아래와 같이 출력되어야 합니다.

```
p1 = Point(1, 3)
p2 = Point(3, 1)
r1 = Rectangle(p1, p2)
print(r1.get_area())
print(r1.get_perimeter())
print(r1.is_square())
p3 = Point(3, 7)
p4 = Point(6, 4)
r2 = Rectangle(p3, p4)
print(r2.get_area())
print(r2.get_perimeter())
print(r2.is_square())
```

```
4
8
True
9
12
True
```