201700949 설재혁

In [12]:

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

# 1번 - 리스트 컴프리헨션
odd_numbers = [x for x in range(11) if x%2 != 0] # 0~10 사이의 숫자 중 조건문에 일치하는 숫자
odd_squares = [y * y for y in odd_numbers] # 저장된 리스트 요소들을 제곱하여 새로운 리스트에 저장

odd_numbers
odd_squares
```

Out[12]:

'2로 나눴을 때 0이 아니면 홀수라는 뜻 range함수를 통해 0~10까지 반복문을 돌려 조건에 일치하는 숫자를 리스트에 저장, 저장된 리스트 안의 요소들을 제곱하여 새로운 리스트에 저장하는 코드 - 리스트 컴프리헨션을 사용해서 작성 '

Out[12]:

```
[1, 3, 5, 7, 9]
```

Out[12]:

```
[1, 9, 25, 49, 81]
```

In [2]:

```
# 2번
from collections import Counter

"""Counter 모듈은 key와 값의 빈도를 계산할 때 용이하다."""
text = ["never", "never", "never", "give", "up"]
result = Counter(text) # 리스트 안의 요소들을 검사하여 key와 빈도수를 출력해준다.

result
```

Out[2]:

```
Counter({'never': 3, 'give': 1, 'up': 1})
```

In [3]:

```
# 3번
class Point:
    # __init__ 메서드를 통해 입력 받은 x,y 값 초기화
    def __init__(self, x, y):
        self.x = x
        self.y = y
    # x값 설정하는 메서드
    def setX(self, x):
        self.x = x
    # y값 설정하는 메서드
    def setY(self, y):
        self.y = y
    # x, y 값을 return 하는 메서드
    def get(self):
        return self.x,self.y
    # x, y 값을 입력받은 파라미터만큼 이동시키는 메서드
    def move(self, dx, dy):
        self.x += dx
        self.y += dy

point = Point(3,3)
point.get()
point.setX(5)
point.get()
point.setY(5)
point.get()
point.move(-3,5)
point.get()
```

Out[3]:

```
(3, 3)
```

Out[3]:

```
(5, 3)
```

Out[3]:

```
(5, 5)
```

Out[3]:

```
(2, 10)
```

In [21]:

```
# 4번
list1 = ['x', 'y', 'z']
list2 = [1,2,3]
list3 = list() # 빈 리스트 생성
list1 = tuple(list1) # expected : ('x','y','z')
list2 = tuple(list2) # expected : (1, 2, 3)

list3.append(list1) # 빈 리스트에 추가
list3.append(list2) # 빈 리스트에 추가
list3

temp_list1,temp_list2 = [i for i in list3] # 리스트 안의 두 개의 요소(튜플)를 임시 리스트 1,2
list4 = list(temp_list1) # expected : ['x','y','z']
list5 = list(temp_list2) # expected : [1, 2, 3]
list4
list5
```

Out[21]:

```
[('x', 'y', 'z'), (1, 2, 3)]
```

Out[21]:

```
['x', 'y', 'z']
```

Out[21]:

```
[1, 2, 3]
```

In [29]:

```
# 5번
import random

# seed가 같다면 동일한 난수를 출력하기 때문에 결과가 같다.

random.seed(10) # 시드 설정
print(random.random()) # 출력
random.seed(10) # 위와 같은 시드 설정
print(random.random()) # 시드가 같으므로 동일한 결과 출력
random.seed(16) # 새로운 시드 설정
print(random.random()) # 출력
random.seed(16)
random.randrange(16)
random.seed(16)
random.randrange(16)
random.randrange(1,6)
up_to_ten = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
random.shuffle(up_to_ten) # random.shuffle은 리스트의 항목을 임의 순서로 재정렬
print(up_to_ten)
```

```
0.5714025946899135
0.5714025946899135
0.36152277491407514
```

Out[29]:

```
11
```

Out[29]:

```
11
```

Out[29]:

```
4
```

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

In [ ]: