

3. 다음 코드의 출력 결과는 무엇인가요? b

```
def my_gen():  
    yield 1  
    yield 2  
    yield 3  
gen = my_gen()  
print(next(gen))  
print(next(gen))
```

- a) 1
- b) 1, 2
- c) 1, 3
- d) 에러 발생

4. 다음 중 제너레이터 표현식의 올바른 예는 무엇인가요? b

- a) `[x * 2 for x in range(5)]`
- b) `(x * 2 for x in range(5))`
- c) `{x * 2 for x in range(5)}`
- d) `lambda x: x * 2 for x in range(5)`

5. 다음 코드의 실행 결과는 무엇인가요? a

```
def countdown(n):  
    while n > 0:  
        yield n  
        n -= 1  
  
gen = countdown(3)  
for x in gen:  
    print(x, end=" ")
```

- a) 3 2 1
- b) 1 2 3
- c) 에러 발생
- d) 3 3 3

6. 주어진 리스트를 이터레이터로 변환하고, 각 요소를 하나씩 출력하는 프로그램을 작성하세요.

```
numbers = [1, 2, 3, 4, 5]
```

```
python > ch15 > assign15.py > MyIterator > _iter_
1 #6. 주어진 리스트를 이터레이터로 변환하고, 각 요소를 하나씩 출력하는 프로그램을 작성하세요.
2
3 numbers = [1, 2, 3, 4, 5]
4 # 방법 1 : 클래스
5 class MyIterator:
6     def __init__(self, data):
7         self.data = data
8         self.cnt = 0
9     def __iter__(self):
10        return self
11    def __next__(self):
12        if self.cnt > len(self.data)-1: # >= 로 수정
13            raise StopIteration
14        result = self.data[self.cnt]
15        self.cnt += 1
16        return result
17
18 i = MyIterator(numbers)
19 for item in i:
20     print(item)
```

문제 출력 디버그 콘솔 터미널 포트

```
PS C:\rokey> & C:/Users/TaeyeonLee/AppData/Local/Programs/Python/Python313/python.exe c:/rokey/python/ch15/assign15.py
1
2
3
4
5
PS C:\rokey>
```

```
python > ch15 > assign15.py > ...
1 #6. 주어진 리스트를 이터레이터로 변환하고, 각 요소를 하나씩 출력하는 프로그램을 작성하세요.
2
3 numbers = [1, 2, 3, 4, 5]
4 # 방법 2 : iter()
5 i = iter(numbers)
6 for item in i:
7     print(item)
```

문제 출력 디버그 콘솔 터미널 포트

```
PS C:\rokey> & C:/Users/TaeyeonLee/AppData/Local/Programs/Python/Python313/python.exe c:/rokey/python/ch15/assign15.py
1
2
3
4
5
PS C:\rokey> []
```

7. 주어진 리스트에서 next() 함수를 사용하여 각 요소를 하나씩 출력하세요. StopIteration 예외를 처리하여 출력이 끝날 때까지 반복되도록 하세요.

```
fruits = ["apple", "banana", "cherry"]
```

```
28 # 7. 주어진 리스트에서 next() 함수를 사용하여 각 요소를 하나씩 출력하세요.
29 # StopIteration 예외를 처리하여 출력이 끝날 때까지 반복되도록 하세요.
30 fruits = ["apple", "banana", "cherry"]
31 class Fruit:
32     def __init__(self, data):
33         self.data = data
34         self.cnt = 0
35     def __iter__(self):
36         return self
37     def __next__(self):
38         if self.cnt > len(self.data) - 1:
39             raise StopIteration
40         result = self.data[self.cnt]
41         self.cnt += 1
42         return result
```

```
43
44 f = Fruit(fruits)
45
46 try:
47     while True:
48         print(next(f))
49 except StopIteration:
50     print("모든요소 출력.")
```

문제 출력 디버그 콘솔 터미널 포트

```
PS C:\rokey> c:; cd 'c:\rokey'; & 'c:\Users\TaeyeonLee\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\TaeyeonLee\.vscode\extensions\ms-p
n' '63154' '--' 'c:\rokey\python\ch15\assign15.py'
apple
banana
cherry
모든요소 출력.
PS C:\rokey> []
```

8. 0부터 9까지의 숫자를 이터레이터로 순회하며, 각 숫자의 제곱을 출력하는 프로그램을 작성하세요.

```
53 # 8. 0부터 9까지의 숫자를 이터레이터로 순회하며,  
54 # 각 숫자의 제곱을 출력하는 프로그램을 작성하세요.  
55  
56 i=iter(range(10))  
57 try:  
58     while True:  
59         num = next(i) ** 2  
60         print(num)  
61 except StopIteration:  
62     print("출력 완료")  
63
```

문제 출력 디버그 콘솔 터미널 포트

```
PS C:\rokey> & C:/Users/TaeyeonLee/AppData/Local/Programs/Python/Python313/python.exe c:/rokey/python/ch15/assign15.p  
0  
1  
4  
9  
16  
25  
36  
49  
64  
81  
출력 완료  
PS C:\rokey> 
```

9, 0부터 10까지의 숫자 중 짝수만 출력하는 프로그램을 작성하세요. 이때, 이터레이터를 사용해야 합니다.

```
65 # 9. 0부터 10까지의 숫자 중 짝수만 출력하는 프로그램을 작성하세요.  
66 # 이때, 이터레이터를 사용해야 합니다.  
67 gen = (x for x in range(11) if x % 2 == 0)  
68 ~ for even_num in gen:  
69     print(even_num)
```

문제 출력 디버그 콘솔 터미널 포트

```
PS C:\rokey> & C:/Users/TaeyeonLee/AppData/Local/Programs/Python/Python313/python.exe c:/rokey/python/ch15/assign15.  
0  
2  
4  
6  
8  
10  
PS C:\rokey> 
```

10. MyRange라는 이름의 클래스를 만들어, range()와 같은 기능을 하는 이터레이터를 작성하세요. start, stop, step을 인자로 받아서 동작해야 하며, __iter__()와 __next__() 메서드를 구현해야 합니다.

```
77 class MyRange:
78     def __init__(self, start, stop, step=1):
79         self.start = start
80         self.stop = stop
81         self.step = step
82         self.cnt = 0
83
84     def __iter__(self):
85         return self
86     def __next__(self):
87         # 내가 생각한 range 공식
88         result = self.start + self.cnt*self.step
89         ###
90         # step이 양수일 때와 음수일 때 조건 분리
91         if self.step > 0:
92             if result >= self.stop: # 카운팅이 stop에 걸릴때
93                 raise StopIteration
94         else: # step < 0
95             if result <= self.stop:
96                 raise StopIteration
97         ###
98         self.cnt += 1
99         return result
100
101 print("====Case1====")
102 real = range(0,10,3)
103 my = MyRange(0,10,3)
104 for i in real:
105     print(f"range의 {i}번째 반복")
106 for i in my:
107     print(f"MyRange의 {i}번째 반복")
108
109 print("====Case2====")
110 real2 = range(10, 0, -3) # 10에서 시작해서 0보다 클 때까지 -3씩
111 my2 = MyRange(10, 0, -3)
112 for i in real2:
113     print(f"range의 {i}번째 반복")
114 for i in my2:
115     print(f"MyRange의 {i}번째 반복")
```


문제 출력 디버그 콘솔 터미널 포트

```
PS C:\rokey>
PS C:\rokey> c++; cd 'c:\rokey'; & 'c:\Users\TaeyeonLee\AppData\Local\Programs\Python\Python39\python.exe' 'c:\rokey\main.py'
=====Case1=====
range의 0번째 반복
range의 3번째 반복
range의 6번째 반복
range의 9번째 반복
MyRange의 0번째 반복
MyRange의 3번째 반복
MyRange의 6번째 반복
MyRange의 9번째 반복
=====Case2=====
range의 10번째 반복
range의 7번째 반복
range의 4번째 반복
range의 1번째 반복
MyRange의 10번째 반복
MyRange의 7번째 반복
MyRange의 4번째 반복
MyRange의 1번째 반복
PS C:\rokey> █
```