

05_for_loop_exercise_solution.py

```

1
2 #####
3 #
4 #   1. 1~10의 숫자로 이루어진 리스트에서 for 반복문을 이용해서 리스트의 원소들을 출력하시오.
5 #
6 #####
7
8 numUnderTen = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
9
10 for item in numUnderTen:
11     print(item)
12
13
14 #####
15 #
16 #   2. 1~10의 숫자로 이루어진 리스트를 range() 함수를 이용해서 생성하고,
17 #   생성한 리스트의 원소들을 for 반복문을 이용해서 출력하시오.
18 #
19 #####
20
21 numUnderTen2 = list(range(1, 11))
22 # print('numUnderTen2:', numUnderTen2)    # 결과 확인을 위한 print() 구문
23
24 for item in numUnderTen2:
25     print(item)
26
27 # for 구문을 이용해서 원소에 접근할 경우 list 형태로 감싸주지 않아도 o.k.
28 # for item in range(1, 11):
29 #     print(item)
30
31
32 #####
33 #
34 #   3. 1~10의 숫자로 구성된 리스트를 생성하고(문제 1, 문제 2 방법 중 어떤 방법도 좋음),
35 #   for 반복문과 if-else 조건문을 이용하여 리스트에서 짝수 만 출력하시오.
36 #
37 #####
38
39 numUnderTen3 = list(range(1, 11))
40 # for 구문의 반복 대상으로 사용할 경우 numUnderTen3 = range(1, 11)도 가능합니다.
41
42 for item in numUnderTen3:
43     if item%2 == 1: # 원소(변수 item에 저장되어 있는 값)가 홀수(2로 나누어 나머지가 1)인 경우
44         continue
45     else: # 반복 대상에서 가져온 원소가 짝수인 경우
46         print(item)
47
48
49 #####
50 #
51 #   4. 1~9의 숫자로 구성된 리스트를 생성하고(문제 1, 문제 2 방법 중 어떤 방법도 좋음),

```

```

52 #   생성한 리스트와 for 반복문을 이용하여 구구단을 만들고 화면에 출력하시오
53 #
54 #####
55
56 oneToNine = list(range(1, 10))
57 # for 구문의 반복 대상으로 사용할 경우 oneToNine = range(1, 10)도 가능합니다.
58
59 for num1 in oneToNine: # 단
60     for num2 in oneToNine: # 열
61         print(num1*num2, end=' ') # end=' '는 출력 후 줄바꿈 대신 공백문자 ' '를 출력
62     print('') # 아무 문자도 출력하지 않지만 print()는 동작을 마치고 기본 end인 줄바꿈을 출
    력
63
64
65 #####
66 #
67 #   5. 1~9의 숫자로 구성된 리스트를 생성하고(문제 1, 문제 2 방법 중 어떤 방법도 좋음),
68 #   enumerate() 함수와 for 반복문을 이용하여 리스트의 인덱스와 값(원소)의 쌍을 화면에 출력하시
    오.
69 #
70 #####
71
72 oneToNine = list(range(1, 10))
73 # enumerate() 함수는 인자로 주어진 리스트, 튜플에서 인덱스와 값을 쌍으로 한번에 꺼내줍니다.
74
75 for idx, item in enumerate(oneToNine):
76     print(idx, item)
77
78
79 #####
80 #
81 #   6. enumerate() 함수와 for 반복문을 이용하여 위 문제 5에서 생성한 리스트에서
82 #   인덱스 0, 2, 4, ..., 즉 홀수 번째 위치의 값(원소)들 만 화면에 출력하시오.
83 #
84 #####
85
86 #   oneToNine    =   [1, 2, 3, 4, 5, 6, 7, 8, 9]
87 #   index        =   0, 1, 2, 3, 4, 5, 6, 7, 8
88 #   따라서 oneToNine의 홀수번째 값들의 위치는 짝수 인덱스 0, 2, 4, 6, 8이 됩니다.
89
90 for idx, item in enumerate(oneToNine):
91     if idx%2 == 1: # 인덱스(idx)%2 == 1, 즉 인덱스가 홀수 --> 짝수 번째 위치 원소들
92         continue
93     else: # 인덱스(idx)%2 == 0, 즉 짝수 인덱스 --> 홀수 번째 위치 원소들
94         print(item)
95
96
97 #####
98 #
99 #   7. 1~9의 숫자로 구성된 리스트를 생성하고(문제 1, 문제 2 방법 중 어떤 방법도 좋음),
100 #   enumerate() 함수와 for 반복문을 이용하여 구구단의 홀수단 만 화면에 출력하는 코드를 작성하시
    오.
101 #
102 #####

```

```

103
104 oneToNine = list(range(1, 10, 1))
105 # oneToNine = [1, 2, 3, 4, 5, 6, 7, 8, 9]
106 # index = 0, 1, 2, 3, 4, 5, 6, 7, 8
107
108 for idx, num1 in enumerate(oneToNine):
109     if idx%2 == 0: # 짝수 인덱스 = 홀수 단(인덱스는 0부터 시작하기 때문입니다.)
110         for num2 in oneToNine: # 열 데이터
111             print(num1*num2, end=' ') # 단(num1)과 열 값(num2)을 곱한 후 출력해 줍니다.
end=' '는 값 출력 후 줄바꿈 대신 공백문자 ' '를 출력
112             print('') # 아무 문자도 출력하지 않지만 print()는 동작을 마치고 기본 end인 줄바꿈을
출력
113         else: # 홀수 인덱스 = 짝수 단
114             continue
115
116
117 #####
118 # 문제 해결을 위해 코드를 작성하는 일반적인 접근 방법(예, 구구단)
119 #####
120
121 # ㉑ 구구단이 만들어지는 과정을 정리해본다.
122
123 # ㉒ [1, 2, 3, ..., 9]의 숫자들과 반복문을 이용해서 단을 생성한다.
124
125 # ㉓ 각 단에서 역시 [1, 2, 3, ..., 9]의 숫자들을 반복문으로 차례로 읽어 열을 생성한다.
126
127 # ㉔ 단에 해당하는 값과 열에 해당하는 값을 곱해서(단x열) 구구단의 값을 생성한다.
128
129 # ㉕ 생성한 구구단 값을 화면에 표시한다.
130
131 # ㉖ 정리한 각 단계를 파이썬 문법으로 어떻게 옮길지 생각하고 문법에 맞도록 작성해 본다.
132
133 # ㉗ 작성한 코드가 정상적으로 동작하는지 테스트하고 잘못된 부분은 찾아서 고친다(print() 함수 이
용).
134

```