

제4장 4-3 반복문과 while반복문 마무리

▶ 4가지 키워드로 정리하는 핵심 포인트

1. ()는 정수의 범위를 나타내는 값입니다. range() 함수로 생성합니다.
2. ()은 조건식을 기반으로 특정 코드를 반복해서 실행할 때 사용하는 구문입니다.
3. ()는 반복문을 벗어날 때 사용하는 구문입니다.
4. ()는 반복문의 현재 반복을 생략할 때 사용하는 구문입니다.

정답 : 1. 범위, 2. while 반복문, 3. break 키워드, 4. continue 키워드

▶ 확인문제

1. 다음 표를 채워 보세요. 코드가 여러 개 나올 수 있는 경우 가장 간단한 형태를 넣어 주세요. 예를 들어 range(5), range(0, 5), range(0, 5, 1)은 모두 같은 값을 나타내는데, 이때는 range(5)로 넣어 주세요.

코드	나타내는 값
range(5)	[0, 1, 2, 3, 4]
range(4, 6)	()
range(7, 0, -1)	()
()	[3, 4, 5, 6, 7]
()	[3, 6, 9]

정답 : [4, 5], [7, 6, 5, 4, 3, 2, 1], range(3, 8), range(3, 9 + 1, 3)

2. 빈칸을 채워 키와 값으로 이루어진 각 리스트를 조합해 하나의 딕셔너리를 만들어 주세요.

```
key_list = ["name", "hp", "mp", "level"]
```

```
value_list = ["기사", 200, 30, 5]
```

```
character = {}
```

```
()
```

```
print(character)
```

정답

```
for i in range(0, len(key_list)):
    character[key_list[i]] = value_list[i]
```

3. 1부터 숫자를 하나씩 증가시키면서 더하는 경우를 생각해 봅시다. 몇을 더할 때 1000을 넘는지 구해 보세요. 그리고 그때의 값도 출력해 보세요. 다음은 100000이 넘는 경우를 구한 예입니다.

(예) 1, 1 + 2 = 3, 1 + 2 + 3 = 6, 1 + 2 + 3 + 4 = 10, ...

```
limit = 10000
```

```
i = 1
```

```
()
```

```
print("{}를 더할 때 {}을 넘으며 그때의 값은 {}입니다.".format(i, limit,
sum_value))
```

정답

```
sum_value = 0
```

```
while sum_value < limit:
```

```
    sum_value += i
```

```
    i += 1
```

4. 1부터 100까지의 숫자가 있다고 합시다. 이를 다음과 같이 계산한다고 했을 때, 최대가 되는 경우는 어떤 숫자를 곱했을 때인지를 찾아 주세요.

1 * 99, 2 * 98, 3 * 97, ..., 98 * 2, 99 * 1

```
max_value = 0
```

```
a = 0
```

```
b = 0
```

```
for i in ():
```

```
    j = 100 - i
```

```
    ()
```

```
print("최대가 되는 경우: {} * {} = {}".format(a, b, max_value))
```

정답

```
range(1, 100 // 2 + 1),
```

```
current = i * j
```

```
    if max_value < current:
```

```
        a = i
```

```
        b = j
```

```
        max_value = current
```