

제어문 (2)

 PDF	비어 있음
 비교	반복문
 숫자	9
 실습문제	비어 있음
 실습문제답안	비어 있음

02. 반복문

02-01. 반복문 개요

02-01-01. 반복문이란

- 💡 코드의 반복을 최소화하여 읽기 쉽고 유지보수가 용이한 코드를 작성하기 위하여 일정 범위만큼 또는 조건에 부합하는 경우, 어떤 코드를 반복 수행하게 하는 구문이다.

02-02. 반복문 for

02-02-01. for-in

- for문의 표현식에서 객체는 일반적으로 문자열, 리스트, 튜플, 딕셔너리를 의미한다.

`for` 변수 `in` 객체: (실행 구문)

- 객체의 첫 번째 인덱스에 해당하는 값부터 마지막 값까지 차례로 변수에 담겨 실행 구문에서 사용할 수 있다.

- 예시

```
# 리스트 ohgiraffers = ['pig', 'bear', 'gorilla', 'squirrel'] for teacher in ohgiraffers: print(teacher, len(teacher))
```

▼ 실행 결과

```
pig 3
bear 4
gorilla 7
squirrel 8
```

```
# 문자열 for word in 'ohgiraffers': print(word)
```

▼ 실행 결과

```
o
h
g
i
r
a
f
f
e
r
s
```

```
# 딕셔너리 # 딕셔너리는 for문을 통해 반복하면 기본적으로 key 값을 가지고 옴 for x in {'bear': '곰', 'tiger': '호랑이', 'monkey': '원숭이'}: print(x)
```

▼ 실행 결과

```
bear
tiger
monkey
```

```
# 중첩 리스트 for [x, y] in [[1, 2], [3, 4], [5, 6]]: print(x, y)
```

▼ 실행 결과

```
1 2
3 4
5 6
```

- for 키워드는 다음과 같이 간략하게 활용하여 사용할 수도 있다.

```
a = 'abcde' b = [val + 'k' for val in a] print(b)
```

▼ 실행 결과

```
['ak', 'bk', 'ck', 'dk', 'ek']
```

02-02-02. range()

- range는 일련의 숫자를 반복해야 하는 경우 사용하는 함수로, 숫자 배열을 생성하여 for-in에 사용하는 것과 같다.

- 넘겨주는 인자의 갯수에 따라 다른 의미를 가진다.
 - 인자 1개 = 반복 횟수
 - 파라미터로 숫자 하나만 전달하면 변수가 0부터 1씩 증가하며, 파라미터로 준 길이만큼 반복한다.
 - 예) for i in range(10): i가 0부터 9까지 1씩 증가하며 총 10회 반복한다.

```
for 변수 in range(반복횟수): (실행 구문)
```

- 인자 2개 = 생성할 숫자 배열의 시작값과 끝값 지정
 - 시작값부터 (끝값-1) 까지 1씩 증가하며 반복한다.
 - 예) for i in range(5, 9): i가 5부터 8까지 1씩 증가하며 총 4(= 9 - 5)회 반복한다.

```
for 변수 in range(시작값, 끝값): (실행 구문)
```

- 인자 3개 = 생성할 숫자 배열의 시작값과 끝값, 증감 크기 지정
 - 시작값부터 (끝값-1)까지 증감크기 만큼 증가 또는 감소하며 반복한다.
 - 예1) for i in range(5, 9, 2): i가 5부터 8까지 2씩 증가하며 반복한다. (총 2회 반복)
 - 예2) for i in range(5, 1, -2): i가 5부터 1까지 2씩 감소하며 반복한다. (총 2회 반복)

```
for 변수 in range(시작값, 끝값, 증감크기): (실행 구문)
```

[참고] range()의 특징

1. 인덱스의 길이만큼 반복하려면 range() 와 len() 을 같이 사용할 수 있다.

```
introduce = ['i', 'am', 'a', 'pig'] for idx in range(len(introduce)): print(idx, introduce[idx])
```

▼ 실행 결과

```
0 i
1 am
2 a
3 pig
```

2. range() 를 단독 사용하면 그냥 텍스트 자체로 출력된다.

```
range(10) # range(0, 10)
```

3. 연속적인 항목을 얻는 함수나 구성의 대상으로 활용하기에 적합하다.

```
sum(range(4)) # 4 (0 + 1 + 2 + 3)
```

02-03. 반복문 while

02-03-01. while

- 조건이 참(True)인 동안, 실행 구문을 계속해서 반복한다.

while 조건식 : 실행 구문