



6장. 반복문

파이썬 정복



Contents

❖ 목차

- 1. 반복문
- 2. 루프의 활용



1. 반복문

❖ while 반복문

- 조건이 만족하는 동안 명령을 계속 실행
- 루프 (Loop)
 - 반복적으로 처리하는 명령

while

```
student = 1
while student <= 5:
    print(student, "번 학생의 성적을 처리한다.")
    student += 1
```

실행결과

1 번 학생의 성적을 처리한다.
2 번 학생의 성적을 처리한다.
3 번 학생의 성적을 처리한다.
4 번 학생의 성적을 처리한다.
5 번 학생의 성적을 처리한다.

- 반복문은 코드 변경이 쉬움
 - 조건식이나 출력문 등 변경 통해 수정



1. 반복문

sum100

```
num = 1
sum = 0
while num <= 100:
    sum += num
    num += 1
print ("sum =", sum)
```

실행결과

sum = 5050



1. 반복문

❖ for 반복문

- 컬렉션 요소 순서대로 반복하면서 루프 명령 실행

for

```
for student in [1, 2, 3, 4, 5]:  
    print(student, "번 학생의 성적을 처리한다.")
```

실행결과

```
1 번 학생의 성적을 처리한다.  
2 번 학생의 성적을 처리한다.  
3 번 학생의 성적을 처리한다.  
4 번 학생의 성적을 처리한다.  
5 번 학생의 성적을 처리한다.
```

- range 명령

- 일정 범위 리스트 만들고 그 요소를 반복

for sum

```
sum = 0  
for num in range(1, 101):  
    sum += num  
print ("sum =", sum)
```

실행결과

```
sum = 5050
```



1. 반복문

❖ 제어 변수의 활용

- 루프의 반복 횟수와 끝낼 시점을 결정
- % 연산자** : 배수를 판별

ruler

```
for x in range(1, 51):  
    if (x % 10 == 0):  
        print('+', end= '')  
    else:  
        print('-', end = '')
```

-----+-----+-----+-----+

- $x \% 10 == 0$
 - 10의 배수 좌표
- x가 10으로 나누어 떨어지는 자리에 + 를, 나머지 자리에는 - 를 출력



1. 반복문

- 동일 문제를 while 문으로 작성할 경우

ruler3

```
x = 1
while x <= 50:
    if (x % 10):
        print('-', end= '')
    else:
        print('+', end = '')
    x += 1
```



1. 반복문

❖ break

- 반복문을 끝내는 명령
- 특정한 조건에 따라 루프 끝내고 다음 명령으로 이동

break

```
score = [ 92, 86, 68, 120, 56]
for s in score:
    if (s < 0 or s > 100):
        break
    print(s)
print("성적 처리 끝")
```

실행결과

92

86

68

성적 처리 끝



1. 반복문

❖ continue

- 현재 루프만 건너뛰고 나머지는 계속 반복 수행하는 명령
- 현재 반복을 중지하고 루프 선두의 조건 점검한 후 반복 재개

continue

```
score = [ 92, 86, 68, -1, 56]
for s in score:
    if (s == -1):
        continue
    print(s)
print("성적 처리 끝")
```

실행결과

```
92
86
68
56
성적 처리 끝
```



2. 루프의 활용

❖ 이중 루프

- 루프 안 명령 자리에 또다른 루프가 들어가 중첩된 것

gugudan

```
for dan in range(2, 10):  
    print(dan, "단")  
    for hang in range(2, 10):  
        print(dan, "*", hang, "=", dan * hang)  
    print()
```

실행결과

```
2 단  
2 * 2 = 4  
2 * 3 = 6  
2 * 4 = 8  
2 * 5 = 10  
2 * 6 = 12  
2 * 7 = 14  
2 * 8 = 16  
2 * 9 = 18
```

```
3 단  
3 * 2 = 6  
3 * 3 = 9  
....
```



2. 루프의 활용

```
dan 루프 { for dan in range(2, 10):  
           print(dan, "단")  
hang 루프 { for hang in range(2,10):  
             print(dan, "*", hang, "=", dan * hang)  
           print()
```

- 이중 루프를 작성하는 경우에도 들여쓰기에 주의하여야
- while 문으로 동일 예제 작성

gugudanwhile

```
dan = 2  
while dan <= 9:  
    hang = 2  
    print(dan, "단")  
    while hang <= 9:  
        print(dan, "*", hang, "=", dan * hang)  
        hang += 1  
    dan += 1  
    print()
```



2. 루프의 활용

❖ 범위의 원칙

- 컴퓨터에서 범위 설정할 때에는 끝 요소 제외하고 직전까지만 포함
- `range(1, 10)` = 1에서 9까지의 범위
 - 범위 구간 나누어 반복 처리할 때의 편의 위함

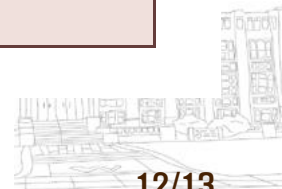
일상생활에서의 범위	컴퓨터에서의 범위
1~10	0~10
11~20	10~20
21~30	20~30
31~40	30~40
41~50	40~50

■ 오프셋

- 기준에서의 상대적인 거리
- 0부터 시작

오프셋

0	1	2	3
base			





Thank You !

파이썬 정복

