

파이썬 기초 04

by goldmont@naver.com



스타트 파이썬
<http://codingschool.biz>

Chapter 04. 반복문

- 반복문이란?
 - for문
 - while문
- break문으로 빠져나가기



반복문이란?

- 반복문 : 같은 블록의 코드를 반복해서 수행
- 파이썬의 반복문 두 가지
 1. for문
 2. while문

반복문을 사용한 경우

```
for x in range(5) :  
    print('안녕하세요!')
```

:: 실행 결과

안녕하세요!
안녕하세요!
안녕하세요!
안녕하세요!
안녕하세요!

for문으로 1~10 정수 합계

```
① sum = 0
② for i in range(1, 11) :
③     sum = sum + i
④     print('i의 값 : %2d => 합계 : %d' % (i, sum))
```

:: 실행 결과

```
i의 값 : 1 => 합계 : 1
i의 값 : 2 => 합계 : 3
i의 값 : 3 => 합계 : 6
i의 값 : 4 => 합계 : 10
i의 값 : 5 => 합계 : 15
i의 값 : 6 => 합계 : 21
i의 값 : 7 => 합계 : 28
i의 값 : 8 => 합계 : 36
i의 값 : 9 => 합계 : 45
i의 값 : 10 => 합계 : 55
```



각 반복 루프의 변수 값

반복 루프	i	sum = sum + i
1번째	1	1 ← 0 + 1
2번째	2	3 ← 1 + 2
3번째	3	6 ← 3 + 3
4번째	4	10 ← 6 + 4
5번째	5	15 ← 10 + 5
6번째	6	21 ← 15 + 6
7번째	7	28 ← 21 + 7
8번째	8	36 ← 28 + 8
9번째	9	45 ← 36 + 9
10번째	10	55 ← 45 + 10

for문의 기본 형식

```
① for 변수 in range(반복 횟수의 범위) :  
②     문장1  
     문장2  
     .....
```

for문에서 range() 함수

```
① for i in range(10) :  
②     print(i, end = ' ')  
③ print()  
  
④ for i in range(1, 11) :  
    print(i, end = ' ')  
⑤ print('\n')  
  
⑥ for i in range(1, 10, 2) :  
    print(i, end = ' ')  
    print()  
  
⑦ for i in range(20, 0, -2) :  
    print(i, end = ' ')
```

:: 실행 결과

```
0 1 2 3 4 5 6 7 8 9  
1 2 3 4 5 6 7 8 9 10
```

```
1 3 5 7 9  
20 18 16 14 12 10 8 6 4 2
```



range() 함수 세 가지 유형

1)

- ① for 변수 in range(종료값) :
- ② <문장1, 문장2,>

2)

- ① for 변수 in range(시작값, 종료값) :
- ② <문장1, 문장2,>

3)

- ① for 변수 in range(시작값, 종료값, 증가 또는 감소) :
- ② <문장1, 문장2,>

for문으로 5의 배수 합계 구하기

```
① sum = 0
② for i in range(100, 301, 5) :
③     sum = sum + i
④ print('100 ~ 300의 정수 중에서 5의 배수의 합계 : %d' % sum)
```

:: 실행 결과

100 ~ 300의 정수 중에서 5의 배수의 합계 : 8200



for문에서 if문 사용하기

```
sum = 0

① for i in range(300, 901) :
②     if i % 3 != 0 :
③         sum = sum + i

④ print('300과 900까지의 정수 중 3의 배수가 아닌 수의 합계 : %d' % (sum))
```

:: 실행 결과

300과 900까지의 정수 중 3의 배수가 아닌 수의 합계 : 240000



for문으로 문자열 처리하기

```
word = input('영어 문장을 입력하세요 : ')  
  
for x in word :  
    print(x)
```

:: 실행 결과

영어 문장을 입력하세요 : I am happy!

I

a

m

h

a

p

p

y

!



for문의 문자열 처리 형식

```
for 변수 in 문자열 :
```

```
    문장1
```

```
    문장2
```

```
    ....
```

for문으로 섭씨/화씨 환산표 만들기

```
① print('-' * 30)
② print('%7s %7s' % ('섭씨', '화씨'))
   print('-' * 30)
③ for c in range(-20, 31, 5):
④     f = c * 9.0 / 5.0 + 32.0
⑤     print('%8d %7s' % (c, f))

print('-' * 30)
```

:: 실행 결과

섭씨	화씨
-20	-4.0
-15	5.0
-10	14.0
-5	23.0
0	32.0
5	41.0
10	50.0
15	59.0
20	68.0
25	77.0
30	86.0



이스케이프 코드

코드	의미
\n	줄바꿈
\t	탭
\\	역 슬래시(\) 자체를 출력
'\''	단 따옴표(') 자체를 출력
'\"'	쌍 따옴표(") 자체를 출력

연습문제 4-1. for문으로 4의 배수 아닌 수

```
count = (1)_____

for i in range(800, 901) :
    if i (2)_____ 4 != 0 :
        print('%4d'% i, end=" ")
        count = count + 1

    if count % (3)_____ == 0 :
        print()
```

:: 실행 결과

```
801 802 803 805 806 807 809 810 811 813
...
881 882 883 885 886 887 889 890 891 893
894 895 897 898 899
```



연습문제 4-2. for문으로 길이 환산표 만들기

```
print('(1)_____ ' * 60)
print('%7s Wt %7s Wt %7s Wt %7s' % ('cm', 'mm', 'm', 'inch'))
print('(1)_____ ' * 60)

for cm in range(70, 90, 2) :
    mm = (2)_____ * 10.0
    (3)_____ = cm * 0.01
    inch = cm * 0.3937
    print('%7d Wt %7.0f Wt %7.2f Wt %7.1f' % ((4)_____,
        mm, m, (5)_____ ))
print('-' * 60)
```

:: 실행 결과

cm	mm	m	inch

70	700	0.70	27.6
72	720	0.72	28.3
...			
88	880	0.88	34.6



2단 구구단

① `a = 2`

② `for b in range(1, 10) :`

`c = a * b`

`print('%d x %d = %d' % (a, b, c))`

∴ 실행 결과

`2 x 1 = 2`

`2 x 2 = 4`

...

`2 x 9 = 18`



이중 for문 : 전체 구구단 표

```
print('-' * 50)

① for a in range(2, 10) :
②     for b in range(1, 10) :
        c = a * b
        print('%d x %d = %d' % (a, b, c))

print('-' * 50)
```

∴ 실행 결과

2 x 1 = 2

2 x 2 = 4

...

2 x 9 = 18

...

9 x 1 = 9

...

9 x 8 = 72

9 x 9 = 81



while문 : 1~10 합계 구하기

```
① sum = 0
② i = 1
③ while i <= 10 :
④     sum = sum + i
⑤     print('i의 값 : %2d => 합계 : %d' % (i, sum))
⑥     i = i + 1
```

∴ 실행 결과

i의 값 : 1 => 합계 : 1
i의 값 : 2 => 합계 : 3
i의 값 : 3 => 합계 : 6
i의 값 : 4 => 합계 : 10
i의 값 : 5 => 합계 : 15
i의 값 : 6 => 합계 : 21
i의 값 : 7 => 합계 : 28
i의 값 : 8 => 합계 : 36
i의 값 : 9 => 합계 : 45
i의 값 : 10 => 합계 : 55

반복 루프 내의 변수 값

반복 루프	i 의 값	조건식(i <= 10)	sum = sum + i	i = i + 1
1번째	1	1 <= 10 : 참	1 ← 0 + 1	2 ← 1 + 1
2번째	2	2 <= 10 : 참	3 ← 1 + 2	3 ← 2 + 1
3번째	3	3 <= 10 : 참	6 ← 3 + 3	4 ← 3 + 1
4번째	4	4 <= 10 : 참	10 ← 6 + 4	5 ← 4 + 1
5번째	5	5 <= 10 : 참	15 ← 10 + 5	6 ← 5 + 1
6번째	6	6 <= 10 : 참	21 ← 15 + 6	7 ← 6 + 1
7번째	7	7 <= 10 : 참	28 ← 21 + 7	8 ← 7 + 1
8번째	8	8 <= 10 : 참	36 ← 28 + 8	9 ← 8 + 1
9번째	9	9 <= 10 : 참	45 ← 36 + 9	10 ← 9 + 1
10번째	10	10 <= 10 : 참	55 ← 45 + 10	11 ← 10 + 1
11번째	11	11 <= 10 : 거짓	반복 루프를 빠져나감	

while문의 기본 형식

- ① 변수 값 초기화
- ② while 조건식 :
- ③ 문장1
 문장2

- ④ 변수 값의 증가(또는 감소)

while문으로 영어 모음 개수 구하기

```
① s = 'Python is widely used by a number of big companies'  
i = 0  
count = 0  
print('모음 : ', end = '')
```

∴ 실행 결과

모음 : o i i e u e a u e o i o a i e

모음의 개수 : 15



while문으로 영어 모음 개수 구하기

```
② while i <= len(s) - 1 :  
③     if (s[i] == 'a' or s[i] == 'A' or s[i] == 'e' or s[i] == 'E' ₩  
        or s[i] == 'i' or s[i] == 'I' or s[i] == 'o' or s[i] == 'O' ₩  
        or s[i] == 'u' or s[i] == 'U') :  
④         count = count + 1  
⑤         print(s[i], end=' ')  
  
        i = i + 1  
  
⑥ print('₩n모음의 개수 : %d' % count)
```


연습문제 4-3. while문 홀수 누적 합계

```
a = 200
count = 1
sum = 0
while (1)_____ <= 300 :
    if a (2)_____ 2 == 1 :
        sum = sum + a
        print('%6d' % (3)_____, end='')

    if count (4)_____ 10 == 0 :
        print()

    count = count + 1

a = (5)_____ + 1
```

연습문제 4-3. while문 홀수 누적 합계

:: 실행 결과

```
201  404  609  816 1025 1236 1449 1664 1881 2100
2321 2544 2769 2996 3225 3456 3689 3924 4161 4400
...
9881 10164 10449 10736 11025 11316 11609 11904 12201 12500
```



연습문제 4-4. while문 화폐 통화 환산 표

```
print('-' * 50)
print('%7s ₩t %7s ₩t %7s' % ('달러($)', '원화(원)', '유로(€)'))
print('-' * 50)

dollar = 10

while (1)_____ <= 100 :
    won = (2)_____ * 1080
    euro = dollar * 0.81

    print('%7d ₩t %8.0f ₩t %7.1f' % (dollar, (3)_____, euro))

    dollar = (4)_____ + 10

print('-' * 50)
```

연습문제 4-4. while문 화폐 통화 환산 표

:: 실행 결과

달러(\$)	원화(원)	유로(€)
10	10800	8.1
20	21600	16.2
...		
100	108000	81.0

break문 : 반복 루프 빠져 나가기

```
① for i in range(1, 1001) :  
②     print(i)  
  
③     if i == 10 :  
④         break
```

:: 실행 결과

1
2

...

10

