

조건문과 반복문

불리언 로직 (Boolean logic)

- 불리언 로직
 - 참(true)과 거짓(false)을 판별하는 로직
 - 파이썬에서 참과 거짓
 - 거짓: False, 0, None, [], (), {}
 - 참: 거짓인 경우를 제외한 모든 경우 → -1은 True or False?
 - 불리언 연산자
 - and, or, not, in, all, any

```
A = 1
B = 2
A and B
A or B
not A

1 in [1,2,3]
"b" in "abc"

all([1,1,1])
any([0,1,0])
```

불리언 로직 (Boolean logic)

Data Type	True & False
"Python"	True
""	False
[1, 2, 3]	True
[]	False
()	False
{}	False
1	True
0	False
None	False

String, List, Tuple, Dictionary 등의 값이 비어 있지 않으면 참, 비어 있으면 ("", [], (), {}) 거짓.

불리언 로직 (Boolean logic)

- 비교 연산자
 - <, >, <=, >=, ==, !=, is
 - 예제: 모두 참!
 - [1, 2, 3] < [1, 2, 4]
 - 'C' < 'Pascal' < 'Perl' < 'Python'
 - (1, 2, 3, 4) < (1, 2, 4)
 - (1, 2) < (1, 2, -1)
 - (1, 2, 3) == (1.0, 2.0, 3.0)
 - (1, 2, ('aa', 'ab')) < (1, 2, ('abc', 'a'), 4)
 - '=='과 'is'의 차이점
 - ==: 값을 비교
 - is: 주소를 비교 (변수는 주소와 값으로 구성)

값이 작을 때는 같은 주소 공유!

```
a=1
b=1
print(a==b)
print(a is b)
print(id(a))
print(id(b))

a=10000000000000000000
b=10000000000000000000
print(a==b)
print(a is b)
print(id(a))
print(id(b))

True
True
140723292874528
140723292874528
True
False
2092886183856
2092886413648
```

조건문 (if statement)

- 조건문

- 조건의 참/거짓에 따라 분기하는 구문
- 형식

```
if condition :  
    do_something
```

```
if condition : statement
```

```
if condition :  
    do_something  
else:  
    do_something_else
```

```
if condition :  
    do_something  
elif condition2 :  
    do_something_else
```

```
if condition :  
    do_something  
elif condition2 :  
    do_something_else1  
else:  
    do_something_else2
```

조건문 (if statement)

```
x = int(input("x="))
```

```
if 1 < x < 5:  
    print("lower range")
```

```
if 1 < x < 5:  
    print("lower range")
```

```
else:  
    print("out of range")
```

```
if 1 < x < 10: print("in range")
```

```
n1 = float(input("num1="))  
n2 = float(input("num2="))  
op = input("operator=")  
if op == '+':  
    res = n1+n2  
    print("두 수의 합은", res, "입니다.")  
elif op == '-':  
    res = n1-n2  
    print("두 수의 차는", res, "입니다.")  
elif op == '*':  
    res = n1*n2  
    print("두 수의 곱은", res, "입니다.")  
elif op == '/':  
    res = n1/n2  
    print("두 수의 나눗셈", res, "입니다.")  
else:  
    print("잘못된 연산자를 입력하셨습니다.")  
  
num1=10  
num2=20  
operator=/  
두 수의 나눗셈은 0.5 입니다.
```

조건문 (if statement)

- 조건 표현문(conditional expression)

- 조건에 따라 값을 할당하는 구문
- 형식: 변수 = 값1 if 조건 else 값2

```
frac = 1/x if x>0 else 0
```

```
if 1 < x < 5:  
    print("lower range")  
else:  
    print("out of range")
```



```
x = int(input("x="))  
message = "lower range" if 1 < x < 5 else "upper range"  
print(message)  
  
x=4  
lower range
```

```
if 1 < x < 5:  
    print("lower range")  
elif 5 < x < 10:  
    print("upper range")  
else:  
    print("out of range")
```



```
x = int(input("x="))  
message = "lower range" if 1 < x < 5 else "upper range" if 5 < x < 10 else "out of range"  
print(message)  
  
x=6  
upper range
```

반복문 (for statement)

- for문

- 주어진 리스트에서 값을 하나씩 가져와 참인 동안 특정 구간을 반복하는 구문
- 형식

```
for variable in sequence :  
    statement1  
    statement2  
    ...
```

```
for i in range(10):  
    print(i)
```

```
for i in range(10,0,-1):  
    print(i)
```

```
for ch in "mystring":  
    print(ch)
```

```
for e in ["red", "green", "blue"]:  
    print(e)
```

반복문 (for statement)

중첩 for문

- for문 내부에 for문이 존재하는 반복 구문

형식

for variable in sequence :
 for variable in sequence :
 statement1
 statement2
 ...

```
for i in range(2,10):
    print("{0:d}단".format(i))
    for j in range(1,10):
        k = i*j
        print("{0:d}*{1:d}={2:d}".format(i,j,k))
    print("")
```

```
[2단]
2*1=2
2*2=4
2*3=6
2*4=8
2*5=10
2*6=12
2*7=14
2*8=16
2*9=18
```

```
v=[[1,2,3],[4,5,6]]
for i in v:
    print(i)
    for j in i:
        print(j)
```

```
[1, 2, 3]
1
2
3
[4, 5, 6]
4
5
6
```

```
[3단]
3*1=3
3*2=6
3*3=9
3*4=12
3*5=15
3*6=18
3*7=21
3*8=24
3*9=27
```



Edited by Harksoo Kim

반복문 (for statement)

One-line for문

- for문을 1개 라인으로 축약해서 표현하는 구문

형식

[variable for variable in sequence]

```
v = list(range(10))
print(v)

for i in v:
    print(i)

print([i for i in v])

v=[[1,2,3],[4,5,6]]

for i in v:
    for j in i:
        print(j)

print([j for i in v for j in i])
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
0
1
2
3
4
5
6
7
8
9
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
1
2
3
4
5
6
[1, 2, 3, 4, 5, 6]
```



Edited by Harksoo Kim

반복문 (for statement)

continue와 break

- continue: 반복문으로 이동하는 구문
- break: 반복문을 빠져나가는 구문

```
for i in range(10):
    if 2<i<5:
        continue
    print(i)
```

```
for ch in "this is a string":
    if ch == ' ':
        break
    print(ch)
```



Edited by Harksoo Kim

반복문 (for statement)

예제

```
i = 0
for ch in "mystring":
    print(i,ch)
    i = i + 1
```



```
for i,ch in enumerate("mystring"):
    print(i,ch)
```

```
0 m
1 y
2 s
3 t
4 r
5 i
6 n
7 g
0 m
1 y
2 s
3 t
4 r
5 i
6 n
7 g
```

```
nums = [1,2,3,4,5]
for i in nums:
    if i==3: del nums[3]
    print(i)
```

Don't modify list while iterating over it!



Edited by Harksoo Kim

반복문 (while statement)

- while문

- 주어진 조건이 참인 동안 특정 구간을 반복하는 구문
- 형식

```
while condition :  
    statement1  
    statement2  
    ...  
  
i = 0  
while i < 10 :  
    print(i)  
    i += 1  
  
i = 0  
while 1 :  
    print(i)  
    i += 1  
    if i >= 10:  
        break  
  
i = 0  
while 1 :  
    i += 1  
    if i < 5:  
        continue  
    print(i)
```

참조 (references)

- 참조 할당

- 리스트와 딕셔너리 할당은 값이 아니라 주소를 할당

```
# content copy  
v1 = 10  
v2 = v1  
v1 = 50
```

```
# address copy in list  
l1 = [10]  
l2 = l1  
l1[0] = 50
```

```
# content copy in list  
l1 = [10]  
l2 = l1[:]  
l1[0] = 50
```

```
v1=10  
v2=v1  
v1=50  
print(v1,v2)  
  
l1=[10]  
l2=l1  
l1[0]=50  
print(l1,l2)  
  
l1=[10]  
l2=l1[:]  
l1[0]=50  
print(l1,l2)  
  
50 10  
[50] [50]  
[50] [10]
```

실습

- 국어, 영어, 수학 점수를 입력 받아서 평균을 계산하고 평균에 따라 등급을 부여하는 프로그램을 작성 하시오.

- 실행 예

```
국어= 90  
영어= 85  
수학 = 72  
평균은 82.3이며, 등급은 B입니다.
```

?

실습

- 라인 수를 입력 받아서 아래와 같은 모양을 출력하는 프로그램을 작성 하시오.

- 실행 예

```
Lines=11  
10000000001  
01000000010  
00100000100  
00010001000  
00001010000  
00000100000  
00001010000  
00010001000  
00100000100  
01000000010  
10000000001
```

?

실습

- 반복문과 조건문을 사용하여 입력된 정수들 중에 가장 큰 값을 출력하는 프로그램을 작성하시오.
 - 실행 예

Numbers? 10 456 87 78
Max = 456



Edited by Harksoo Kim

질의응답

Q&A

Homepage: <http://nlp.konkuk.ac.kr>
E-mail: nlpdrkim@konkuk.ac.kr



Edited by Harksoo Kim