

파이썬의 제어문

이번 장에서는 파이썬에서 분기, 반복과 같은 제어 방법을 알아봅니다.



1. 조건문과 코드블록
2. 분기문
3. 반복문





1. 조건문과 코드블록

□ Bool 자료형

- True, False의 두가지 값을 갖는 자료형
- 어떤 데이터가 참, 거짓으로 평가되는 지 알고싶다면 bool() 함수를 사용
예) >>> bool(0)
False

□ 조건문

- 평가 결과가 참(True) 또는 거짓(False)으로 평가됨
- 거짓으로 평가되는 예
 - False, None, 숫자 0, 비어있는 순서열((), [] 등), 비어있는 딕셔너리({ })

□ 코드블록 (code block)

- 여러 코드가 이루는 블록으로 파이썬에서는 들여쓰기로 구역을 구분하며
- 코드블록 시작전의 바로 윗줄의 끝에 콜론(:)이 위치해야 함
- 참고: C, Java 언어는 { } 로 구분



2. 분기문

□ if ~ else 문

if 조건문:

명령1

명령2

...

else:

명령3

명령4

...

if 뒤에 조건이 위치하고, 그 뒤에 콜론 :

: 뒤에는 들여쓰기로 이루어진 코드블록이 위치함.
이 코드블록은 if문의 조건이 True일 경우 실행됨

if문의 조건 평가의 결과가 False일 때 else의 코드 블록이 수행됨
else 뒤에도 코드블록이 오므로 콜론 : 이 위치함



2. 분기문

□ if ~ elif ~ else 문

```
if 조건문1:
```

```
    명령1
```

```
    명령2
```

```
    ...
```

```
elif 조건문2:
```

```
    명령3
```

```
    명령4
```

```
    ...
```

```
else 조건문3:
```

```
    명령5
```

```
    명령6
```

```
    ...
```



□ if 문 예

```
print('수를 입력하세요 : ')\n\na = int(input())\n\nif a == 0 :\n    print('0은 나눗셈에 이용할 수 없습니다.')\nelse :\n    print('3 /', a, '=', 3/a)
```





□ if 문 예

```
money = 10000
```

```
if money > 5000 :
```

```
    print( '택시 타고 가자.' )
```

```
elif money > 1000 :
```

```
    print( '버스 타고 가자.' )
```

```
else :
```

```
    print( '걸어가자.' )
```



□ if 문 예

```
print('수를 입력하세요 : ')\n\na = int(input())\n\nif a > 10:\n    if a % 2 == 0:\n        print('10보다 큰 짝수')\n    else:\n        print('10보다 큰 홀수')\nelse:\n    if a % 2 == 0:\n        print('10 이하의 짝수')\n    else:\n        print('10 이하의 홀수')
```




3. 반복문

□ while 문

while 조건문:

코드블록





3. 반복문

□ while 문 예

```
print('몇 번 반복할까요? : ')\nlimit = int(input())\n\ncount = 0\nwhile count < limit:\n    count = count + 1\n    print('{0}회 반복.'.format(count))\n\nprint('종료되었습니다.')
```





3. 반복문

□ for 문

- 조건문을 평가하는 것이 아니라
- 순서열의 항목을 반복변수에 차례대로 복사하고, 순서열의 마지막까지 가게 되면 종료

```
for 반복변수 in 순서열:  
    코드블록
```





3. 반복문

□ for 문 예

```
for i in (1, 2, 3) :  
    print(i)
```

변수 i에는 매 반복마다 튜플 (1, 2, 3)의 요소들이 차례대로 복사됨. 결국 for문은 3번 반복을 수행.

```
for s in ['뇌를', '자극하는', '파이썬']:  
    print(s)
```



3. 반복문

□ for 문 예

```
>>> for i in range(5):  
    print(i)
```

0

1

2

3

4



3. 반복문

□ for 문 예

```
>>> for i in range(0, 5, 1):  
    print(i)
```

0

1

2

3

4

```
>>> for i in range(0, 10, 2):  
    print(i)
```

0

2

4

6

8



3. 반복문

□ 다음을 실행해보시오

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

```
>>> for i in range(10, -5, -2):  
    print(i)
```



3. 반복문

□ for 문 예

```
>>> dic = {'명지대': 'www.mju.ac.kr',  
          '컴공과': 'cs.mju.ac.kr',  
          '방산안보연구소': 'idis.mju.ac.kr'}  
>>> for k, v in dic.items():  
    print("{0} : {1}".format(k, v))
```

명지대 : www.mju.ac.kr

컴공과 : cs.mju.ac.kr

방산안보연구소 : idis.mju.ac.kr



3. 반복문

□ continue 문

- 반복문이 실행하는 코드블록의 나머지 부분을 실행하지 않고, 다음 반복을 수행함

```
for i in range(10):  
    if i % 2 == 1:  
        continue  
  
    print(i)
```



3. 반복문

□ break 문

- 반복문을 중단하고 빠져나감





3. 반복문

□ 다음을 실행하고, 내용을 설명하시오.

```
a = [1, 2, 3, 1, 2]
new_a = [ ]

for i in a :
    if i not in new_a :
        new_a.append(i)

print(new_a)
```

