

Chapter 02. 파이썬 문법

01. 변수명과 예약어

02. 자료형과 연산자

03. 객체

04. 제어문

05. 함수

06. 입출력

07. 모듈

08. 클래스

09. 예외처리

4. 제어문

4.1 조건문

1) if 문

```
if( 조건식 1) :  
    구문1  
    구문2  
  
elif( 조건식2) :  
    구문3  
    구문4  
  
else :  
    구문 5
```

[예제 condition.py]

```
a = 10  
if a > 5:  
    print("big")  
else:  
    print("small")
```

4. 제어문

```
n = -2
if n > 0:
    print('양수')
elif n < 0:
    print('음수')
else:
    print('0')

order = 'spam'
if order == 'spam':
    price = 1000
elif order == 'egg':
    price = 500
elif order == 'spagetti':
    price = 2000
else:
    price = 0

print(price)
```

4. 제어문

4.1 조건문

2) 조건 표현식(Conditional Expression)

C 또는 Java 3항 연산자에서 역할을 한다.

value = true-expr if condition else false-expr

[예제 condition.py]

```
print('big' if a > 5 else 'small')
```

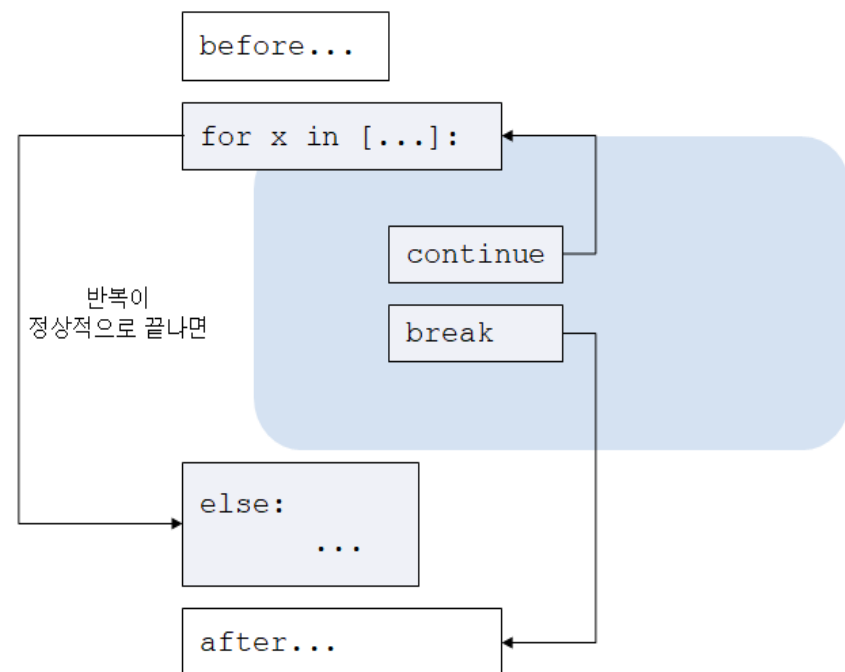
4. 제어문

4.2 반복문

1) for 문

```
for <타킷> in <객체>:  
    구문1  
    구문2  
else :  
    구문 3
```

1. <객체>는 list, str, tuple, bytes, bytearray, range 등 시퀀스 자료형
2. 반복횟수는 <객체>의 크기
3. <객체> 안의 객체 하나씩 순차적으로 구문1, 구문2 가 실행된다.
4. 반복이 정상적으로 끝나면 else 블록의 구문3이 실행된다.
5. for 문내에서 break로 빠져 나오면 else 블록은 실행되지 않는다.



4. 제어문

[예제 for.py]

list, range 객체를 사용하는 for문

```
a = ['cat', 'cow', 'tiger']
```

```
for animal in a:  
    print(animal)
```

```
for x in range(10):  
    print(x, end=" ")
```

복합 자료형을 사용하는 for문

```
l = [('둘리', 10), ('마이콜', 20), ('도우넳', 30)]
```

```
for data in l:  
    print('이름:%s, 나이:%d' % data)
```

```
for name, age in l:  
    print('이름:{0}, 나이:{1}'.format(name, age))
```

4. 제어문

요소의 값뿐만 아니라 인덱스가 필요한 경우에는 enumerate() 함수를 사용한다.

```
l = ['red', 'orange', 'yellow', 'green', 'blue']  
for index, color in enumerate(l):  
    print(index, color)
```

어떤 조건에서 반복을 중지하고 빠져 나가야 하는 경우에는 break 문을 사용한다.

```
for i in range(10):  
    if i > 5:  
        break  
  
    print(i, end=' ')
```

continue 문을 만나면, 이 후 구문은 실행 하지 않고 처음으로 이동한다.

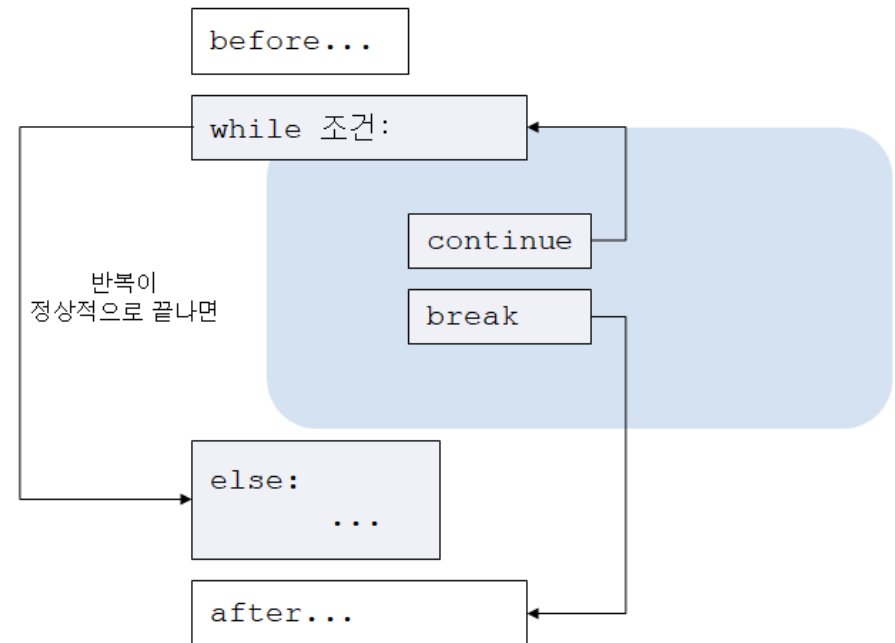
```
for i in range(10):  
    if i <= 5:  
        continue  
  
    print(i, end=' ')
```

4. 제어문

2) while 문

```
while <조건식>:  
    구문1  
    구문2  
else :  
    구문 3
```

1. <조건>이 참이면 구문1, 구문2 가 반복해서 실행된다.
2. else 블록은 while문을 빠져 나올 때 실행된다.
3. 단, while문 내 break로 빠져 나오면 실행되지 않는다.



4. 제어문

[예제 while.py]

while 반복문

```
count = 1
while count < 11:
    print(count, end=' ')
    count += 1
else:
    print("")
```

1 ~ 10 까지 합

```
hap, i = 0, 1
while i <= 10:
    hap += i
    i += 1

print( '합: {0}'.format(hap))
```

4. 제어문

break, continue, else 사용하기

```
i = 0
while i < 10:
    i += 1
    if i < 5:
        continue
    print(i, end=' ')
    if i > 10:
        break
else:
    print('else block')

print('done')
```

다음과 같은 구문으로 무한루프를 구성할 수 있으나 반드시 break할 수 있는 조건이 내부에 있어야 한다.

```
i = 0
while True:
    print(i)
    if i > 5:
        break
    i += 1
```