

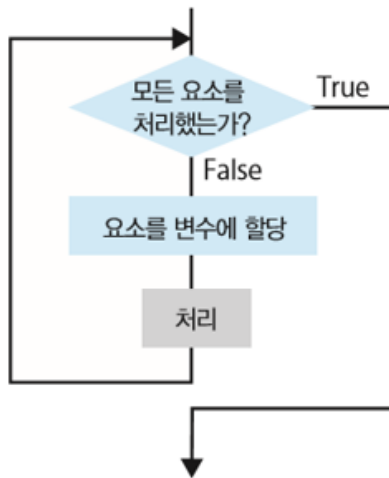


## CHAPTER 4. 리스트 다루기



# 리스트와 반복 문

- for~in 문은 문자열이나 리스트, 튜플 등 일정한 순서대로 나열된 시퀀스형 객체를 요소의 처음부터 끝까지 하나씩 꺼내서 순서대로 처리할 때 사용



```
[ for 변수 in 대상 리스트 등 :  
    처리 ]
```

# 숫자 리스트 만들기

- range()는 for 문과 자주 사용하는 내장 함수
- range() 함수는 등차수열 리스트를 생성하는 함수로, for 문과 조합해서 사용하면 다음과 같이 동작

```
range(x)  
range(x, y)  
range(x, y, z)
```

- 숫자리스트에 적용할 수 있는 기본적인 통계 함수 : min()/max()/sum()
- 리스트 내포(comprehension)  
[ 식 for 반복변수 in 리스트 if 조건식 ]

# 리스트 슬라이스

## ■ 리스트 자르기

파이썬 시퀀스형(문자열, 리스트, 튜플등)은 인덱스를 지정할 때 정수  $x$ 와  $y$ 를 콜론(:)으로 이어서  $[x:y]$ 처럼 작성하면 인덱스  $x$ 에서  $y-1$  범위에 있는 요소들을 한 번에 지정할 수 있음(이때 인덱스  $y$ 가 가리키는 요소는 포함되지 않음!!)

```
>>> a = [1, 2, 3, 4]
>>> a[0:2]
[1, 2]
```

```
>>> a[:2]
[1, 2]
```

```
>>> a = [1, 2, 3, 4]
>>> a[0:2] = [0, 0]
>>> a
[0, 0, 3, 4]
```

# 리스트 복사

- 새로운 리스트객체명 = 기존리스트객체명      같은 리스트를 가리킴
- VS
- 새로운 리스트객체명 = 기존리스트객체명[:]      리스트 복사

# 튜플

- 튜플은 리스트와 비슷한 자료형이지만, 불변이므로 일부 요소만 변경하거나 제거할 수 없다는 점이 다름
- 튜플에도 다양한 객체를 요소로 저장할 수 있음

## 변수에 튜플을 할당하고 인덱스 지정하기

- 튜플을 정의하려면 **각 요소를 콤마로 이어서 작성하고** 전체를 소괄호(**( )**)로 둘러쌘  
(요소1, 요소2, 요소3, 요소4, 요소5, ...)
- 괄호 생략 가능, 그러나 생략하면 나중에 프로그램이 읽기 힘들어지므로 생략하지 않는 것이 좋음
- 시퀀스형이므로 문자열이나 리스트처럼 각 요소를 인덱스로 지정할 수 있음

```
>>> a = ('P', 'y', 't', 'h', 'o', 'n')
>>> a[0]
'P'
>>> a[-1]
'n'
```

# 튜플

- 하나의 튜플에 서로 다른 자료형이 존재 가능

```
>>> a = ('Python', 1234, 1.234)
>>> a[0]
'Python'
>>> a[1]
1234
>>> a[2]
1.234
```

- 튜플에 튜플을 저장할 수도 가능

```
>>> a = (1, 2, 3, (4, 5, 6))
>>> a[3][0]
4
```

# 튜플

- 여러 개의 변수에 튜플을 할당하면 각 변수에 튜플의 요소가 할당됨

```
>>> a = (1, 2, 3)
>>> x, y, z = a
>>> x
1
>>> y
2
>>> z
3
```

## 여러 요소를 인덱스로 지정하기

튜플도 슬라이스로 여러 요소를 지정할 수 있음

```
>>> a = (1, 2, 3, 4)
>>> a[0:2]
(1, 2)
```



# 튜플

- 연산자로 튜플 연결/반복하기
- 문자열이나 리스트처럼 + 연산자와 \* 연산자로 튜플을 연결, 반복 가능

```
>>> a = ('(^o^)', '^(^)')  
>>> b = (':o:)', '(T^T)')  
>>> a + b  
('^o^)', '^(^)', ':o:)', '(T^T)')
```

```
>>> a = ('(^o^)', '^(^)')  
>>> a * 3  
('^o^)', '^(^)', '^o^)', '^(^)', '^o^)', '^(^)')
```