제4강 Lists

학습 목차

- 리스트 정의
- 리스트 멤버 액세스
- 슬라이스
- 다중 배열
- 멤버 삭제와 삽입
- 리스트 iteration
- 멤버 판별
- 기타 기능
- 튜플

리스트 (List)

- 순서가 있는 값들의 집합
- 파이썬의 내장 자료구조

```
>>> [1, 2, 3]
[1, 2, 3]
>>> ['cat', 'bat', 'rat', 'elephant']
['cat', 'bat', 'rat', 'elephant']
>>> ['hello', 3.1415, True, None, 42]
['hello', 3.1415, True, None, 42]
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam
['cat', 'bat', 'rat', 'elephant']
```

리스트 멤버 액세스

```
>>> black_pink = ['jisu', 'jeni', 'rose', 'risa']
>>> black_pink[0]
'jisu'
>>> black_pink[3]
'risa'
>>> black_pink[4]
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    black_pink[4]
IndexError: list index out of range
>>> black_pink[-1]
'risa'
>>> black_pink[-2]
'rose'
```

슬라이스 – 규칙에 따른 여러 개의 멤버들을 추출

[start : stop : step]

```
>>> num = '0123456789ABCDEF’
>>> num[0:5]
'01234’
>>> num[1:9:2]
'1357’
>>> num[:]
'0123456789ABCDEF’
>>> num[::2]
'02468ACE’
>>> num[-4:]
'CDEF’
>>> num[::-1]
'FEDCBA9876543210'
```

리스트 슬라이스

```
>>> spam = ['cat', 'bat', 'rat', 'elephant']
>>> spam[0:4]
['cat', 'bat', 'rat', 'elephant']
>>> spam[1:3]
['bat', 'rat']
>>> spam[0:-1]
['cat', 'bat', 'rat']
```

다중 배열

```
>>> spam = [['cat', 'bat'], [10, 20, 30, 40, 50]]
>>> spam[0]
['cat', 'bat']
>>> spam[0][1]
'bat'
>>> spam[1][4]
50
```

리스트 처리

```
>>> black_pink = ['jisu', 'jeni', 'rose', 'risa']
>>> len(black_pink)
4
>>> black_pink[0] = 'daehyun'
>>> black_pink
['daehyun', 'jeni', 'rose', 'risa']
>>> black_pink.append('JYP')
>>> black_pink
['daehyun', 'jeni', 'rose', 'risa', 'JYP']
>>> black_pink = black_pink + ['YG']
>>> black_pink
['daehyun', 'jeni', 'rose', 'risa', 'JYP', 'YG']
```

```
>>> twice = ['momo', 'sana', 'zwi', 'nayun', 'dahyun']
>>> unite = black_pink + twice
>>> unite
['daehyun', 'jeni', 'rose', 'risa', 'JYP', 'YG', 'momo', 'sana', 'zwi', 'nayun', 'dahyun']
>>> unite.remove('daehyun')
>>> unite
['jeni', 'rose', 'risa', 'JYP', 'YG', 'momo', 'sana', 'zwi', 'nayun', 'dahyun']
>>> del unite[0]
>>> unite
['rose', 'risa', 'JYP', 'YG', 'momo', 'sana', 'zwi', 'nayun', 'dahyun']
>>> unite.insert(2, 'IU')
>>> unite
['rose', 'risa', 'IU', 'JYP', 'YG', 'momo', 'sana', 'zwi', 'nayun', 'dahyun']
```

리스트 멤버 iteration

```
black_pink = ['jisu', 'jeni', 'rose', 'risa’]

for member in black_pink:
    print(member)

for i in range(len(black_pink)):
    print(i, 'th member is', black_pink[i])

for i, member in enumerate(black_pink):
    print('%d th member is %s' % (i, black_pink[i]))
```

리스트 멤버 여부 판별

```
>>> 'howdy' in ['hello', 'hi', 'howdy', 'heyas']
True
>>> spam = ['hello', 'hi', 'howdy', 'heyas']
>>> 'cat' in spam
False
>>> 'howdy' not in spam
False
>>> 'cat' not in spam
True
```

리스트를 활용한 다중 대입

```
>>> cat = ['fat', 'black', 'loud']
>>> size, color, disposition = cat
```

Augmented Assignment Operators

Augmented assignment statement	Equivalent assignment statement
`spam = spam + 1`	`spam += 1`
`spam = spam - 1`	`spam -= 1`
`spam = spam * 1`	`spam *= 1`
`spam = spam / 1`	`spam /= 1`
`spam = spam % 1`	`spam %= 1`

index

```
>>> spam = ['hello', 'hi', 'howdy', 'heyas']
>>> spam.index('hello')
0
>>> spam.index('heyas')
3
>>> spam.index('howdy howdy howdy')
Traceback (most recent call last):
  File "<pyshell#31>", line 1, in <module>
    spam.index('howdy howdy howdy')
ValueError: 'howdy howdy howdy' is not in list
```

sort

```
>>> spam = [2, 5, 3.14, 1, -7]
>>> spam.sort()
>>> spam
[-7, 1, 2, 3.14, 5]
>>> spam = ['ants', 'cats', 'dogs', 'badgers', 'elephants']
>>> spam.sort()
>>> spam
['ants', 'badgers', 'cats', 'dogs', 'elephants']
>>> spam.sort(reverse=True)
>>> spam
['elephants', 'dogs', 'cats', 'badgers', 'ants']
```

```
>>> spam = [1, 3, 2, 4, 'Alice', 'Bob']
>>> spam.sort()
Traceback (most recent call last):
  File "<pyshell#70>", line 1, in <module>
    spam.sort()
TypeError: unorderable types: str() < int()
```

copy()

```
>>> spam = ['A', 'B', 'C', 'D']
>>> cheese = spam
>>> cheese[1] = 77
>>> spam
['A', 77, 'C', 'D']
>>> cheese
['A', 77, 'C', 'D']
```

```
>>> import copy
>>> spam = ['A', 'B', 'C', 'D']
>>> cheese = copy.copy(spam)
>>> cheese[1] = 77
>>> spam
['A', 'B', 'C', 'D']
>>> cheese
['A', 77, 'C', 'D']
```

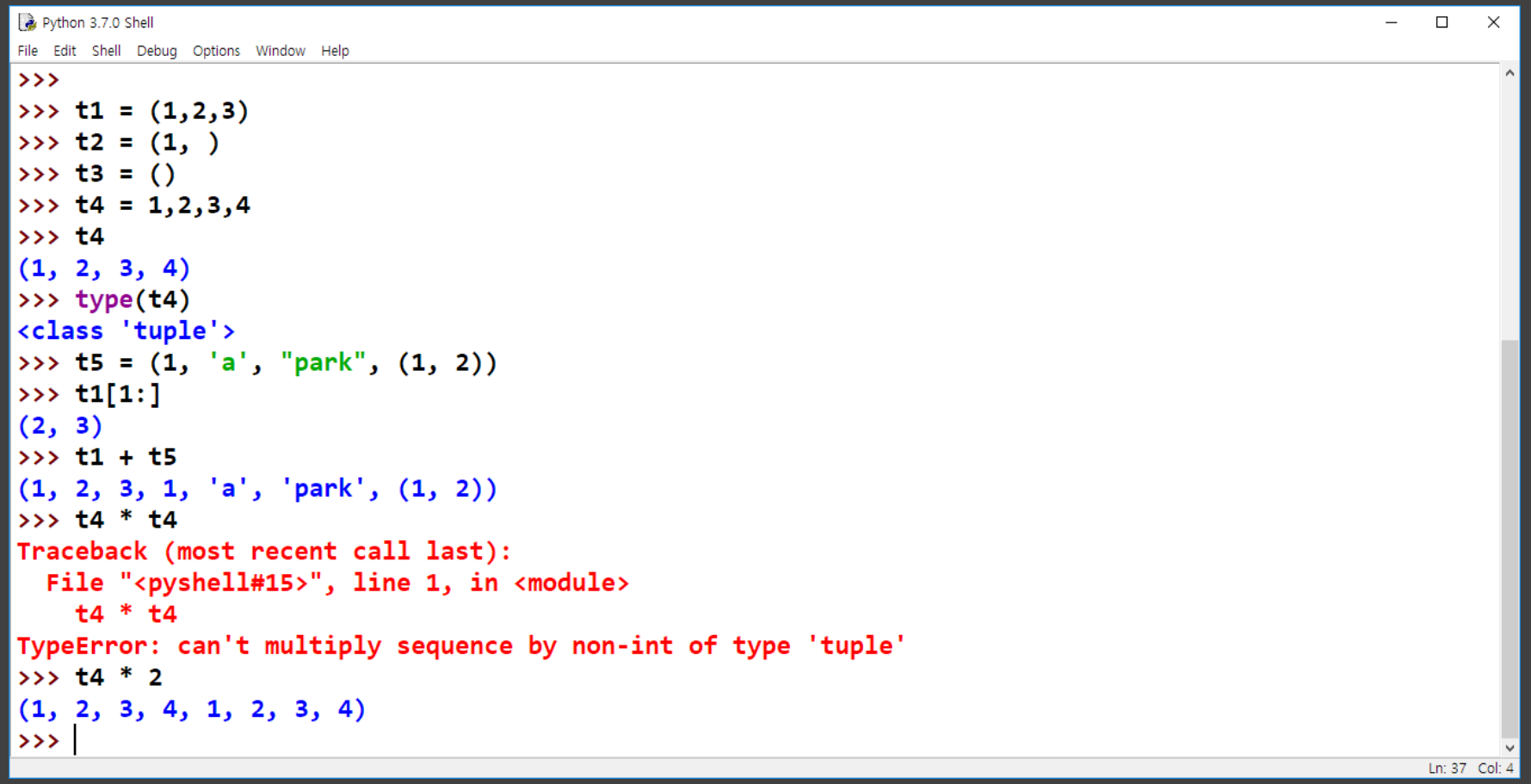
List Comprehension

- 리스트 데이타값들을 규칙에 의해서 자동 filling하는 것.

```
>>> values = [i for i in range(10)]
>>> values
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> odd_values = [n for n in range(20) if n % 2 == 1]
>>> odd_values
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
>>> [(x, y) for x in [1, 2, 3] for y in ['a', 'b', 'c']]
[(1, 'a'), (1, 'b'), (1, 'c'), (2, 'a'), (2, 'b'), (2, 'c'), (3, 'a'), (3, 'b'), (3, 'c')]
>>> vec = [[1,2,3], [4,5,6], [7,8,9]]
>>> [num for elem in vec for num in elem]
[1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> import random
>>> values = [random.randint(0, 5) for i in range(20)]
>>> values
[4, 2, 5, 4, 5, 5, 1, 5, 5, 0, 1, 4, 2, 0, 3, 4, 3, 1, 1, 1]
>>> [v if v != 0 else 'ZERO' for v in values]
[4, 2, 5, 4, 5, 5, 1, 5, 5, 'ZERO', 1, 4, 2, 'ZERO', 3, 4, 3, 1, 1, 1]
```

Tuple

- 여러 개의 값을 동시에 관리. 리스트와 유사.
- 하지만, 기본적으로 값을 바꿀 수는 없음. ==> 프로그램 중 변경이 되지 않는 값들의 모음이 필요할 때 사용하면 됨.



함수 다중 인자 전달 및 액세스

```
>>> def sum(*values):
...     result = 0
...     for v in values:
...         result += v
...     return result
...
>>> sum(1)
1
>>> sum(1,2,3)
6
>>> sum(1,2,3,4,5,6,7,8,9,10)
55
```

list ←→ tuple

```
>>> tuple(['cat', 'dog', 5])
('cat', 'dog', 5)
>>> list(('cat', 'dog', 5))
['cat', 'dog', 5]
>>> list('hello')
['h', 'e', 'l', 'l', 'o']
```

정리

- 리스트 생성
- 리스트 액세스, 변경, 및 삭제
- 리스트 활용
- 튜플

과제

과제명	**제 4 강 Lists 과제**
제출방식	온라인
공개일	2021.03.15 오전 11:40
마감일	2021.03.15 오후 12:20