6. 리스트(List) 와 Tuple

6.1 List 구조

- 일상생활에서 흔히 볼 수 있는 자료구조 (Data Structure)

  예) 할일 목록 (to-do list), 식료품 구입 리스트, 초대장 리스트, 소원목록 (wish list), 국가 목록 등.

- List 는 top-10 list 처럼 순서가 있을 수도 있고, 친구 목록처럼 순서가 없을 수도 있음.
- List 의 항목들을 요소 (element) 라고 한다.
- Python 문법에서 list 는 대괄호 (bracket) 로 표시하고 element 는 콤마(,) 로 구분하여 대괄호안에 위치함.
- List 의 element 는 지금까지 배운 Python 의 모든 자료형과 함수가 사용될 수 있음.

In [1]:

```
x = ["쌀", "보리", "옥수수", "파"]
```

In [2]:

```
print(x)
```

['쌀', '보리', '옥수수', '파']

In [3]:

```
y = [1, 'two', 6/2]
```

In [4]:

```
print(y)
```

[1, 'two', 3.0]

In [5]:

```
def f(n):
    return [n, n * 2, n * 3, n * 4]
```

In [6]:

```
print(f(3))
```

[3, 6, 9, 12]

In [7]:

```
y = f(4)
```

In [8]:

```
1 type(y)
```

Out[8]:

list

In [9]:

```
1 f('hi')
```

Out[9]:

['hi', 'hihi', 'hihihi', 'hihihihi']

In [10]:

```
1 a = 5
2 b = 3.5
3 x = [a, a+b, a-b]
```

In [11]:

```
1 print(x)
```

[5, 8.5, 1.5]

list 의 주요 method

- len() : list 의 길이
- sort() : list sort
- reverse() : list 를 역순으로 sort
- append() : list 끝에 new item 추가
- extend() : 다른 list 추가
- pop() : list 의 마지막 item 제거 및 반환
- pop(i) : i 번째 item 제거 및 반환
- del alist[i] : i 번째 item delete
- count() : list element 의 갯수
- index() : list element 의 index

여러개의 list 를 결합

In [12]:

```
1 c = ['z', 'y', 'w'] + ['a', 'c', 'b']
2 c
```

Out[12]:

['z', 'y', 'w', 'a', 'c', 'b']

In [13]:

```
1 c.pop(2)
```

Out[13]:

'w'

In [14]:

```
1 del c[2]
```

In [15]:

```
1 c
```

Out[15]:

['z', 'y', 'c', 'b']

list 의 원소(element) 갯수

In [16]:

```
1 len(c)
```

Out[16]:

4

list 원소(element) 추가 / 확장

In [17]:

```
1 d = []
```

In [18]:

```
1 print(d)
```

[]

In [19]:

```
1 len(d)
```

Out[19]:

0

In [20]:

```
1 d.append('first')
```

In [21]:

```
1 d
```

Out[21]:

['first']

In [22]:

```
1 d.extend(c)
```

In [23]:

```
1 d
```

Out[23]:

['first', 'z', 'y', 'c', 'b']

In [24]:

```
1 e = d + c
2 print(e)
```

['first', 'z', 'y', 'c', 'b', 'z', 'y', 'c', 'b']

list 내의 list

In [25]:

```
1 d.append(['second, third'])
2 d
```

Out[25]:

['first', 'z', 'y', 'c', 'b', ['second, third']]

In [26]:

```
1 animals = [['dog', 'Ruby', 7], ['cat', 'Nero', 3], ['dog', 'John', 5], ['bird', 'Seagal', 2]]
2 print(animals)
```

[['dog', 'Ruby', 7], ['cat', 'Nero', 3], ['dog', 'John', 5], ['bird', 'Seagal', 2]]

list element 의 변수 assign

In [27]:

```
1 [x, y, z] = 1, 2, 3
2 print(x, y, z)
```

1 2 3

In [28]:

```
r, s, t = [1, 2, 3]
print(r, s, t)
```

1 2 3

In [29]:

```
u = [1, 2, 3]
j, k, l = u
print(j, k, l)
```

1 2 3

In [30]:

```
x = 1
y = 2
xlist = [x, y, x + y]
x = 0
y = 0
```

In [31]:

```
xlist
```

Out[31]:

[1, 2, 3]

list methods

In [32]:

```
print(c)
```

['z', 'y', 'c', 'b']

In [33]:

```
c.sort()
```

In [34]:

```
print(c)
```

['b', 'c', 'y', 'z']

In [35]:

```
c.sort(reverse=True)
c
```

Out[35]:

['z', 'y', 'c', 'b']

In [36]:

```
sum(xlist)
```

Out[36]:

6

In [37]:

```
max(xlist)
```

Out[37]:

3

In [38]:

```
min(xlist)
```

Out[38]:

1

In [39]:

```
tuple(xlist)
```

Out[39]:

(1, 2, 3)

In [40]:

```
del(xlist[0])
xlist
```

Out[40]:

[2, 3]

In [41]:

```
alist = [1,2,3,3,4,4,4,5]
alist.count(3)
```

Out[41]:

2

In [42]:

```
1 alist.index(3)
```

Out[42]:

2

6-2. List 의 indexing

list 를 구성하는 element 들은 위치에 따라 index 로 접근할 수 있다.

index 는 대괄호(bracket)안에 정수(integer) 로 표시

Python 의 index 는 0 부터 시작하는 양수 혹은 -1 부터 시작하는 음수로 표시할 수 있다.

```
  0  1   2   3   4   5
+---+---+---+---+---+---+
| H | e | l | l | o | ! |
+---+---+---+---+---+---+
 -6  -5  -4  -3  -2  -1
```

In [43]:

```
1 hello = ['H', 'e', 'l', 'l', 'o', '!']
2 print(len(hello))
```

6

In [44]:

```
1 print(hello)
```

['H', 'e', 'l', 'l', 'o', '!']

In [45]:

```
1 hello[0]
```

Out[45]:

'H'

In [46]:

```
1 hello[5]
```

Out[46]:

'!'

In [47]:

```
hello[len(hello) - 1]
```

Out[47]:

'!'

In [48]:

```
x = [1,2,3] + ['a','b','c']
print(x)
```

[1, 2, 3, 'a', 'b', 'c']

In [49]:

```
x[0]
```

Out[49]:

1

In [50]:

```
x[len(x)-1]
```

Out[50]:

'c'

In [51]:

```
x[-1]
```

Out[51]:

'c'

In [52]:

```
x[3] = 4
```

In [53]:

```
print(x)
```

[1, 2, 3, 4, 'b', 'c']

In [54]:

```
x[4] = 5
x
```

Out[54]:

[1, 2, 3, 4, 5, 'c']

In [55]:

```
x[0] = 'first'
x[-1] = 'last'
x
```

Out[55]:

['first', 2, 3, 4, 5, 'last']

In [56]:

```
x[-2]
```

Out[56]:

5

In [57]:

```
x[len(x) - 1] = 'end'
x
```

Out[57]:

['first', 2, 3, 4, 5, 'end']

In [58]:

```
x[-6]
```

Out[58]:

'first'

In [59]:

```
animals = [['dog', 'Ruby', 7], ['cat', 'Nero', 3], ['dog', 'John', 5], ['bird', 'Seagal', 2]]
```

In [60]:

```
animals[2]
```

Out[60]:

['dog', 'John', 5]

In [61]:

```
animals[2][2]
```

Out[61]:

5

In [62]:

```
kind, name, age = animals[-1]
```

In [63]:

```
1 print(kind, name, age)
```

bird Seagal 2

6-3. 인덱스 (Index) 를 이용한 list 의 element 자르기 (slicing)

list 의 일부분을 범위 지정할 수 있다.

list[start : end] => start ~ end-1
list[start:] => start ~ 끝까지
list[:end] => 처음 ~ end-1
list[:] => list 전체

증가분 (increment) 를 지정하면 더욱 다양한 slicing 이 가능하다.

list[start : end : increment]

increment 가 음수 (-) 이면 뒤에서부터 거꾸로 처리한다.

In [64]:

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

In [65]:

```
1 y = x[0:2]
2 print(y)
```

[1, 2]

In [66]:

```
1 x[2:4]
```

Out[66]:

[3, 4]

In [67]:

```
1 x[2:]
```

Out[67]:

[3, 4, 5, 6]

In [68]:

```
1 x[:6]
```

Out[68]:

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

In [69]:

```
1 x[:]
```

Out[69]:

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

In [70]:

```
1 x[-3:]
```

Out[70]:

[4, 5, 6]

In [71]:

```
1 x[2:-1]
```

Out[71]:

[3, 4, 5]

In [72]:

```
1 x[-6:3]
```

Out[72]:

[1, 2, 3]

In [73]:

```
1 s = [1,2,3,4,5,6,7,8,9,10]
2 python = ['p','y','t','h','o','n']
```

In [74]:

```
1 s[:: 2]
```

Out[74]:

[1, 3, 5, 7, 9]

In [75]:

```
1 python[::2]
```

Out[75]:

['p', 't', 'o']

In [76]:

```
1 s[5::2]
```

Out[76]:

[6, 8, 10]

In [77]:

```
1 python[5::2]
```

Out[77]:

['n']

In [78]:

```
1 s[5:-1:2]
```

Out[78]:

[6, 8]

In [79]:

```
1 s[::-1]
```

Out[79]:

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

In [80]:

```
1 s
```

Out[80]:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

In [81]:

```
1 s[7:1:-1]
```

Out[81]:

[8, 7, 6, 5, 4, 3]

In [82]:

```
1 s[-1:-8:-1]
```

Out[82]:

[10, 9, 8, 7, 6, 5, 4]

In [83]:

```
1 t = ['a','b','c']
```

In [84]:

```
1 t[1:100]
```

Out[84]:

['b', 'c']

In [85]:

```
1 t[-2:-1]
```

Out[85]:

['b']

In [86]:

```
1 t[-100:100]
```

Out[86]:

['a', 'b', 'c']

In [87]:

```
1 t[-100]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-87-dfa1185694f1> in <module>
----> 1 t[-100]

IndexError: list index out of range
```

In [88]:

```
1 t[5]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-88-332e8dd8018d> in <module>
----> 1 t[5]

IndexError: list index out of range
```

In [89]:

```
1 animals = [['dog', 'Ruby', 7], ['cat', 'Nero', 3], ['dog', 'John', 5], ['bird', 'Seagal', 2]]
```

In [90]:

```
1 animals[1][0]
```

Out[90]:

'cat'

In [91]:

```
1 animals[-1][2]
```

Out[91]:

2

In [92]:

```
1 animals_tuple = (('dog', 'Ruby', 7), ('cat', 'Nero', 3), ('dog', 'John', 5), ('bird', 'Seagal', 2))
```

In [93]:

```
1 animals_tuple[1][0]
```

Out[93]:

'cat'

In [94]:

```
1 animals_tuple[-1][2]
```

Out[94]:

2

6-4. 튜플 (Tuple)

튜플은 element 값을 변경할 수 없는 list 이다.

- Mutable : 값을 변경 가능 (list, variable)

- Immutable : 값을 변경 불가능 (tuple)

tuple 은 괄호 ( ) 로 element 들을 감싸준다.

element 를 변경할 수 없는 것 이외의 operation 은 list 와 거의 유사함.

Tuple 과 List 비교

In [95]:

```
1 xlist = [1, 2, 3, 4]
2 ytuple = (1, 2, 3, 4)
```

In [96]:

```
1 xlist[1:4]
```

Out[96]:

[2, 3, 4]

In [97]:

```
1 ytuple[1:4]
```

Out[97]:

(2, 3, 4)

In [98]:

```
1 ytuple[-1]
```

Out[98]:

4

In [99]:

```
1 xlist.append(6)
2 x
```

Out[99]:

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

In [100]:

```
1 ytuple.append(6)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-100-4fa395ce64a4> in <module>
----> 1 ytuple.append(6)

AttributeError: 'tuple' object has no attribute 'append'
```

In [101]:

```
1 xlist[3] = '4th'
2 xlist
```

Out[101]:

```
[1, 2, 3, '4th', 6]
```

In [102]:

```
1 ytuple[3] = '4th'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-102-1292cdf64e6f> in <module>
----> 1 ytuple[3] = '4th'

TypeError: 'tuple' object does not support item assignment
```

tuple merge

In [103]:

```
1 z = ('a','b')
```

In [104]:

```
1 new_tuple = ytuple + z
2 new_tuple
```

Out[104]:

```
(1, 2, 3, 4, 'a', 'b')
```

tuple 생성

In [105]:

```
1 t = 1, "two", 3+4
```

In [106]:

```
t
```

Out[106]:

(1, 'two', 7)

In [107]:

```
type(t)
```

Out[107]:

tuple

In [108]:

```
u = tuple(t)
u
```

Out[108]:

(1, 'two', 7)

In [109]:

```
j, k, l = u
print(j, k, l)
```

1 two 7

In [110]:

```
my_tuple = ('cat', 'dog', 'apple')
```

In [111]:

```
cc, dd, aa = my_tuple
print(cc, dd, aa)
```

cat dog apple

tuple indexing

In [112]:

```
print(my_tuple[-1])
print(my_tuple[-2])
```

apple
dog

In [113]:

```
1 my_tuple[-1] = 'banana'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-113-e7b8cfea9eb1> in <module>
----> 1 my_tuple[-1] = 'banana'

TypeError: 'tuple' object does not support item assignment
```

tuple 정렬 (sort)

In [114]:

```
1 changed_tuple = my_tuple + ('banaba',)
```

In [115]:

```
1 changed_tuple
```

Out[115]:

```
('cat', 'dog', 'apple', 'banaba')
```

In [116]:

```
1 changed_tuple.sort()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-116-2dc348c3a5d3> in <module>
----> 1 changed_tuple.sort()

AttributeError: 'tuple' object has no attribute 'sort'
```

In [117]:

```
1 sorted(changed_tuple)
```

Out[117]:

```
['apple', 'banaba', 'cat', 'dog']
```

zip

- zip(*iterable)은 동일한 개수로 이루어진 자료형을 묶어 주는 역할을 하는 함수

In [118]:

```
1 list(zip([1, 2, 3], [4, 5, 6]))
```

Out[118]:

[(1, 4), (2, 5), (3, 6)]

In [119]:

```
1 list(zip([1, 2, 3], [4, 5, 6], [7, 8, 9]))
```

Out[119]:

[(1, 4, 7), (2, 5, 8), (3, 6, 9)]

In [120]:

```
1 list(zip("abc", "def"))
```

Out[120]:

[('a', 'd'), ('b', 'e'), ('c', 'f')]

In [121]:

```
1 fruits = ['apple', 'orange', 'pear', 'grape']
2 prices = [100, 200, 150, 50]
3
4 for f, p in zip(fruits, prices):
5     print(f, p)
```

apple 100
orange 200
pear 150
grape 50

List / Tuple 연습문제

1) 다음 프로그램의 결과값은 ?

```
xlist = []
xlist.append('Good')
xlist.append('Morning')
print(xlist)

xlist.append([3, 4])
print(xlist)
```

2) 다음 list 의 element 를 오름차순으로 정렬 (ascending sort) 한다. 또한, 내림차순 (descending order)으로 정렬한다. sort(), sort(reverse=True) 함수를 사용한다.

```
xlist = [2, 1, 3, 5, 4]
```

3) 두개의 list element 들을 짝을 지워 출력

stocks = ['삼성전자', '대한항공', 'google', 'apple']
close = [40000, 2000, 50000, 100000]