

3.1 자료구조와 순차 자료형

3.1.1 튜플

튜플 : 변경 불가, 고정 길이를 갖는 파이썬의 순차 자료형

```
In [1]: tup=(4,5,6) #괄호 생략 가능
```

```
In [2]: tup = 4, 5, 6
tup
```

```
Out[2]: (4, 5, 6)
```

```
In [3]: tuple([4, 0, 2])
tup = tuple('string')
tup
```

```
Out[3]: ('s', 't', 'r', 'i', 'n', 'g')
```

```
In [4]: tup[0]
```

```
Out[4]: 's'
```

```
In [5]: nested_tup = (4, 5, 6), (7, 8)
nested_tup
nested_tup[0]
nested_tup[1]
```

```
Out[5]: (7, 8)
```

```
In [22]: tup = tuple(['foo', [1, 2], True])
tup[2] = False # 튜플은 수정 불가
```

```
-----
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_31340\3314447763.py in <module>
      1 tup = tuple(['foo', [1, 2], True])
      2 tup[2] = False
```

```
Out [65]: {0: 4, 1: 3, 2: 2, 3: 1, 4: 0}
```

기본값

```
if key in some_dict:
    value=some_dict[key]
else:
    value=default_value
```

get과 pop은 반환 값을 받는다.

```
value=some_dict.get(key,default_value)
```

get은 None 반환, pop은 예외 발생

```
In [66]: words = ["apple", "bat", "bar", "atom", "book"]
by_letter = {}
```

```
for word in words:
    letter = word[0]
    if letter not in by_letter:
        by_letter[letter] = [word]
    else:
        by_letter[letter].append(word)

by_letter
```

```
Out [66]: {'a': ['apple', 'atom'], 'b': ['bat', 'bar', 'book']}
```

```
In [70]: by_letter = {}
for word in words:
    letter = word[0]
    by_letter.setdefault(letter, []).append(word) #if-else 블록과 동일한 효과
by_letter
```

```
Out [70]: {'a': ['apple', 'atom'], 'b': ['bat', 'bar', 'book']}
```

```
In [71]: from collections import defaultdict
```

```
In [23]: tup[1].append(3) #튜플 내에 저장된 객체는 그 위치에서 변경 가능
tup
```

```
Out [23]: ('foo', [1, 2, 3], True)
```

```
In [7]: (4, None, 'foo') + (5, 0) + ('bar',)
```

```
Out [7]: (4, None, 'foo', 5, 0, 'bar')
```

```
In [8]: ('foo', 'bar') * 4 #복사본 반복 가능, 참조만 복사
```

```
Out [8]: ('foo', 'bar', 'foo', 'bar', 'foo', 'bar', 'foo', 'bar')
```

```
In [9]: tup = (4, 5, 6)
a, b, c = tup
b
```

```
Out [9]: 5
```

```
In [10]: tup = 4, 5, (6, 7)
a, b, (c, d) = tup
d
```

```
Out [10]: 7
```

click to expand output; double click to hide output

```
a
b
b, a = a, b #변수의 이름을 바꿔야 할 때
a
b
```

```
Out [11]: 1
```

```
In [12]: seq = [(1, 2, 3), (4, 5, 6), (7, 8, 9)]
for a, b, c in seq:
    print(f'a={a}, b={b}, c={c}')
```

```
a=1, b=2, c=3
```

