

# Unit 23.

☀ Status	완료
👤 담당자	
📅 마감일	
📅 완료일	

## Unit 23. 2차원 리스트 사용하기

- 2차원 리스트는 가로 x 세로 형태로 이루어져 있고 행 과 열이 모두 0에서 시작

### 23.1 2차원 리스트를 만들고 요소에 접근하기

- 2차원 리스트는 리스트 안에 리스트를 넣어서 만들 수 있고 각 리스트는 ,(coma)로 구별된다.
- 리스트 = [[값, 값], [값, 값], [값, 값]]

```
a = [[10, 20],
      [30, 40],
      [50, 60] ]
```

#### 23.1.1 2차원 리스트의 요소에 접근

- 요소에 접근하거나 값을 할당할 때는 대괄호를 2번 사용하여 세로 인덱스 가로 인덱스를 지정
- 리스트[세로인덱스][가로인덱스]
- 리스트[세로인덱스][가로인덱스] = 값

```
>>> a = [[10, 20], [30, 40], [50, 60]]
>>> a[0][0]          # 세로 인덱스 0, 가로 인덱스 0인 요소 출력
10
>>> a[1][1]          # 세로 인덱스 1, 가로 인덱스 1인 요소 출력
40
>>> a[2][1]          # 세로 인덱스 2, 가로 인덱스 0인 요소 출력
60
>>> a[0][1] = 1000    # 세로 인덱스 0, 가로 인덱스 1인 요소에 값 할당
>>> a[0][1]
1000
```

튜플 = ((값, 값), (값, 값), (값, 값))

튜플 = ((값, 값), [값, 값], [값, 값])

리스트 = [[값, 값], (값, 값), (값, 값)]

#### 23.2.1 for 반복문을 한번만 사용하기

```
>> a = [[10, 20], [30, 40], [50, 60]]
>>> for x, y in a:      # 리스트의 가로 한 줄(안쪽 리스트)에서 요소 두 개를 꺼냄
...     print(x, y)
...
10 20
30 40
50 60
```

in 앞에 변수 두개를 지정해주면 가로 한줄에서 요소 두개를 꺼내고 반복

이렇게 사용하려면 2차원 리스트의 가로 크기가 크지 않아야 유용하게 사용 가능하다.

#### 23.2.2 for 반복문을 두번 사용하기

```
a = [[10, 20], [30, 40], [50, 60]]

for i in a:      # a에서 안쪽 리스트를 꺼냄
    for j in i:  # 안쪽 리스트에서 요소를 하나씩 꺼냄
        print(j, end=' ')
    print()
```

a에서 안쪽 리스트를 다 꺼내고 가로 한줄에서 요소를 하나씩 꺼내면 된다.

### 23.2.3 for와 range 사용하기

```
a = [[10, 20], [30, 40], [50, 60]]

for i in range(len(a)):      # 세로 크기
    for j in range(len(a[i])): # 가로 크기
        print(a[i][j], end=' ')
    print()
```

len(a)처럼 2차원 리스트의 크기를 구하면 모든 요소의 개수가 아니라 안쪽 리스트의 크기(세로 크기)가 나온다

```
print(a[i][j], end=' ')
```

요소에 접근할 때는 리스트[세로인덱스][가로인덱스] 형식으로 접근한다

### 23.2.4 while 반복문을 한번 사용하기

```
a = [[10, 20], [30, 40], [50, 60]]

i = 0
while i < len(a):      # 반복할 때 리스트의 크기 활용(세로 크기)
    x, y = a[i]         # 요소 두 개를 한꺼번에 가져오기
    print(x, y)
    i += 1              # 인덱스를 1 증가시킴
```

### 23.2.5 while 반복문을 두번 사용하기

```
a = [[10, 20], [30, 40], [50, 60]]

i = 0
while i < len(a):      # 세로 크기
    j = 0
    while j < len(a[i]): # 가로 크기
        print(a[i][j], end=' ')
        j += 1          # 가로 인덱스를 1 증가시킴
    print()
    i += 1              # 세로 인덱스를 1 증가시킴
```

안쪽 while에서 i를 증가시키면 안된다

## 23.3 for 반복문으로 1차원리스트 만들기

```
a = []      # 빈 리스트 생성

for i in range(10):
    a.append(0)    # append로 요소 추가

print(a)
```

for 반복문으로 10번 반복하면서 append로 요소를 추가하면 된다.

### 23.3.2 for 반복문으로 2차원 리스트 만들기

```
a = []      # 빈 리스트 생성

for i in range(3):
    line = []      # 안쪽 리스트로 사용할 빈 리스트 생성
    for j in range(2):
        line.append(0)    # 안쪽 리스트에 0 추가
```

```
a.append(line)      # 전체 리스트에 안쪽 리스트를 추가

print(a)
```

안쪽 리스트에 append로 0을 추가하고 append를 이용해 전체 리스트에 안쪽 리스트를 추가한다.

### 23.3.3 리스트 표현식으로 2차원 리스트 만들기

```
>>> a = [[0 for j in range(2)] for i in range(3)]
>>> a
[[0, 0], [0, 0], [0, 0]]
```

0을 두번 반복하고 나온 [0, 0]을 다시 3번 반복

### 23.3.4 가로 크기가 불규칙한 튜플형 리스트 만들기

가로 크기를 알고 있다고 가정

```
a = [3, 1, 3, 2, 5]    # 가로 크기를 저장한 리스트
b = []                 # 빈 리스트 생성

for i in a:             # 가로 크기를 저장한 리스트로 반복
    line = []           # 안쪽 리스트로 사용할 빈 리스트 생성
    for j in range(i):   # 리스트 a에 저장된 가로 크기만큼 반복
        line.append(0)
    b.append(line)       # 리스트 b에 안쪽 리스트를 추가

print(b)
```

## 24.4 2차원 리스트의 할당과 복사 알아보기

```
>>> a = [[10, 20], [30, 40]]
>>> b = a
>>> b[0][0] = 500
>>> a
[[500, 20], [30, 40]]
>>> b
[[500, 20], [30, 40]]
```

2차원 변수를 만든 뒤에 다른 변수에 할당하고 요소를 변경해보면 변수 모두 반영된다

```
>>> a = [[10, 20], [30, 40]]
>>> b = a.copy()
>>> b[0][0] = 500
>>> a
[[500, 20], [30, 40]]
>>> b
[[500, 20], [30, 40]]
```

copy 메서드를 사용해도 가능하다