

2차원 리스트 이해

7주차_03_01

한 동 대 학 교

김경미 교수

학습목표

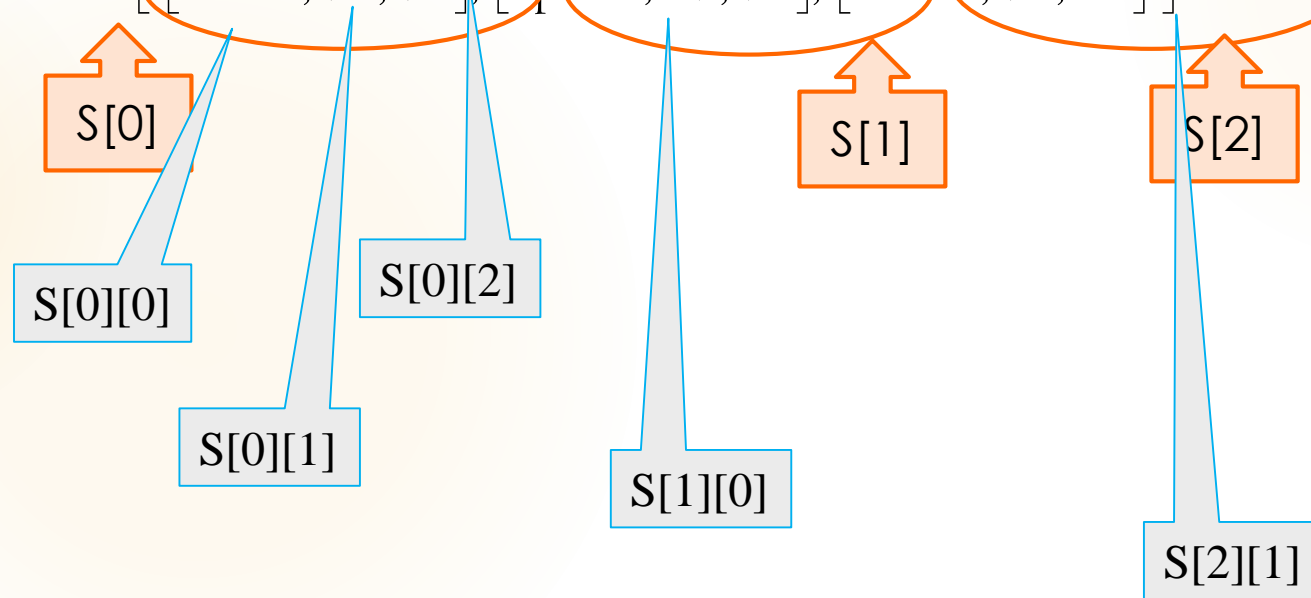
2

- ▶ 2차원 리스트 이해하기
- ▶ 2차원 리스트 활용하기

2차원 lists

3

- ▶ 2차원 목록(list) 생성
- ▶ 리스트 안에 리스트가 또 만들어진다
- ▶ `s = [["kim", 90, 75], ["park", 89, 95], ["choi", 76, 85]]`



2차원 lists

- ▶ 2차원 list의 각 항목에 접근할 시,
 - ▶ 리스트[index1][index2]
- ▶ index1
 - ▶ 전체의 2차원 리스트 안에 작은 리스트에 접근
- ▶ index2
 - ▶ 작은 리스트의 각 항목에 접근

2차원 리스트 활용

```
fq= [[0,0,0,0,0,0],[1,1,1,1,1,1],[2,2,2,2,2,2],[3,3,3,3,3,3],[4,4,4,4,4,4]]
```

```
for i in range(5):  
    print(fq[i])
```

```
for i in range(5):  
    for j in range(6):  
        print(fq[i][j])
```

```
[0, 0, 0, 0, 0, 0]  
[1, 1, 1, 1, 1, 1]  
[2, 2, 2, 2, 2, 2]  
[3, 3, 3, 3, 3, 3]  
[4, 4, 4, 4, 4, 4]  
0  
0  
0  
0  
0  
0  
0  
0  
1  
1  
1  
1  
1  
1  
1  
1  
2  
2  
2  
2  
2  
2  
2  
2  
3  
3  
3  
3  
3  
3  
3  
3  
4  
4  
4  
4  
4  
4  
4  
4
```

2차원 리스트 예제 I

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

```
a[0].append(0)
```

```
a[0].append(30)
```

```
a[1].append(500)
```

```
a[2].append(7)
```

```
print(a)
```

```
[[10, 20, 0, 30], [30, 40, 500], [50, 60, 7]]
```

2차원 리스트 예제 2

7

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

```
for i in range(3):  
    for j in range(2):  
        if j==0:  
            a[i][j]=a[i][j]*10  
        print(a[i][j])
```

100

20

300

40

500

60

>

2차원 list, 성적처리

```
s = [ ["kim", 90, 75], ["park", 89, 95], ["choi", 76, 85] ]
print( s )

for i in range( len(s) ):
    print( s[i][0] )
    sum=0
    for j in range( 1, len(s[i]) ):
        sum = sum + s[i][j]

    print("sum = ", sum, "average = ", sum/j, "\n")
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise
Code/turtle_test.py =====
[['kim', 90, 75], ['park', 89, 95], ['choi', 76, 85]]
kim
sum = 165 average = 82.5

park
sum = 184 average = 92.0

choi
sum = 161 average = 80.5
>>> |
```


연습문제 I

- ▶ 2단에서 16단까지 구구단을 계산하여,
- ▶ 생성한 2차원 리스트에 저장한다
- ▶ 이 때, 각 단을 한 개의 row에 저장한다
- ▶ 저장한 결과를 출력한다

연습문제 1, 코드와 결과

```
mul = [ 10 * [0] for i in range(15) ]  
print(mul)
```

```
for i in range(2, 17):  
    for j in range(1, 11):  
        mul[i][j] = i * j  
    print(mul[i]) #print by row
```

```
print("Done!!")
```

```
>>>  
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise  
Code/turtle_test.py =====  
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]  
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30]  
[4, 8, 12, 16, 20, 24, 28, 32, 36, 40]  
[5, 10, 15, 20, 25, 30, 35, 40, 45, 50]  
[6, 12, 18, 24, 30, 36, 42, 48, 54, 60]  
[7, 14, 21, 28, 35, 42, 49, 56, 63, 70]  
[8, 16, 24, 32, 40, 48, 56, 64, 72, 80]  
[9, 18, 27, 36, 45, 54, 63, 72, 81, 90]  
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]  
[11, 22, 33, 44, 55, 66, 77, 88, 99, 110]  
[12, 24, 36, 48, 60, 72, 84, 96, 108, 120]  
[13, 26, 39, 52, 65, 78, 91, 104, 117, 130]  
[14, 28, 42, 56, 70, 84, 98, 112, 126, 140]  
[15, 30, 45, 60, 75, 90, 105, 120, 135, 150]  
[16, 32, 48, 64, 80, 96, 112, 128, 144, 160]  
Done!!  
>>>
```

연습문제 2

11

- ▶ 단위행렬을 만든다
- ▶ 먼저 단위행렬의 크기를 입력 받는다
- ▶ 2D list로 단위행렬을 만들고 출력한다

연습문제 2 코드

12

```
Size = int(input("행렬의 크기: "))
Temp_row = []
Unit_Matrix = []
for a in range(size):
    for b in range(size):
        temp_row.append(0)

    Unit_Matrix.append(temp_row)
    temp_row = []

for i in range(size):
    for j in range(size):
        if i == j:
            Unit_Matrix[i][j] = 1
        else:
            Unit_Matrix[i][j] = 0

print(Unit_Matrix)
```

```
>>>
```

```
행렬의 크기: 7
```

```
[1, 0, 0, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 1]
```

```
>>> |
```

강의 요약

13

- ▶ 2차원 리스트를 이해하기
 - ▶ 리스트 안에 리스트를 만들어 활용

목표 달성 질문

14

- ▶ 2차원 리스트 구성은 리스트내에 리스트를 추가하는 것인가?
- ▶ 2차원 리스트 a 의 첫번째 아이템내, 두번째 아이템의 이름을 쓰시오

감사합니다