

선생님의 거짓말!!!

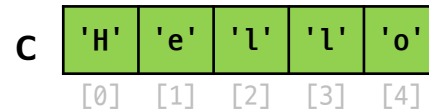
a = 5



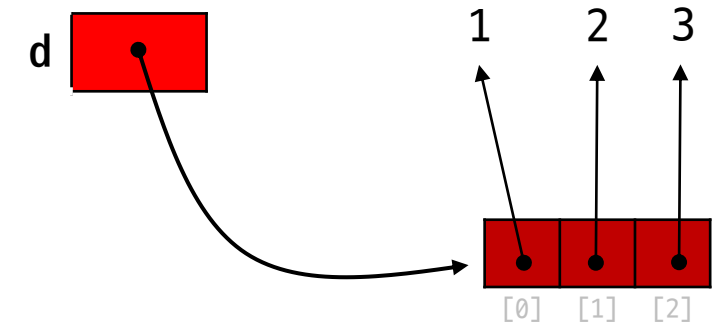
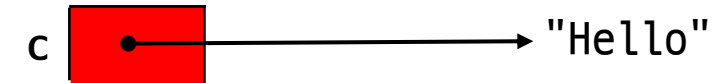
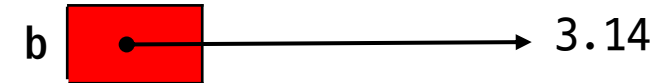
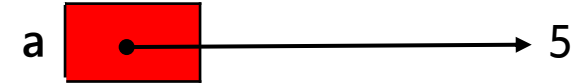
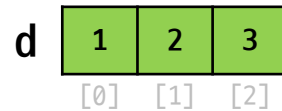
b = 3.14



c = "Hello"



d = [1, 2, 3]



2차원 리스트

1차원 리스트

L1 = [1, 2, 3]

1	2	3
[0]	[1]	[2]

2차원 리스트 – 리스트의 리스트

L2 = [[1, 2, 3], [4, 5, 6]]

	0열	1열	2열
0행	1	2	3
1행	4	5	6

파이썬의 리스트는 여러 자료형을 담을 수 있다.

```
L = [1, 3.14, [2, 3], "Hello"]  
      [0]   [1]   [2]   [3]
```

```
print( L, type(L) )
```

```
print( L[0], type(L[0]) )
```

```
print( L[1], type(L[1]) )
```

```
print( L[2], type(L[2]) )
```

```
print( L[2][0], type(L[2][0]) )
```

```
print( L[3], type(L[3]) )
```

```
print( L[3][0], type(L[3][0]) )
```

1	3.14	2	3	'H'	'e'	'l'	'l'	'o'
		[0]	[1]	[0]	[1]	[2]	[3]	[4]
[0]	[1]	[2]	[3]					

```
[1, 3.14, [2, 3], 'Hello'] <class 'list'>
```

```
1 <class 'int'>
```

```
3.14 <class 'float'>
```

```
[2, 3] <class 'list'>
```

```
2 <class 'int'>
```

```
Hello <class 'str'>
```

```
H <class 'str'>
```

1차원 리스트 vs 2차원 리스트

▶ 1차원 리스트

- ▶ `L1 = [1, 2, 3, 4, 5, 6]`
- ▶ `print(L1[3])` # 4 출력
- ▶ `L1[?] = 20`

1	2	3	4	5	6
[0]	[1]	[2]	[3]	[4]	[5]

20

▶ 2차원 리스트 - 리스트의 리스트

- ▶ `L2 = [[1, 2, 3], [4, 5, 6]]`
- ▶ `print(L2[1][2])` # 6 출력
- ▶ `print(L2[1])` # [4, 5, 6] 출력
- ▶ `print(L2[1])` # [1, 2, 3, 4, 5, 6] 출력
- ▶ `L2[?][?] = 50`

	0열	1열	2열
0행	1	2	3
1행	4	5	6

50

행 열

1차원 리스트 vs 2차원 리스트

```
L1 = [1, 2, 3, 4, 5]
```

```
L2 = [[1], [2, 3], [4, 5, 6]]
```

```
print( len(L1) )
```

```
print( len(L2) )
```

```
print( len(L2[0]) )
```

```
print( len(L2[1]) )
```

```
print( len(L2[2]) )
```

1	2	3	4	5
[0]	[1]	[2]	[3]	[4]

	0열	1열	2열
0행	1		
1행	2	3	
2행	4	5	6

1차원 리스트 순차적 접근

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

0, 1, 2, 3, 4, 5

```
for i in range(6) :  
    print(L[i])
```

1	2	3	4	5	6
[0]	[1]	[2]	[3]	[4]	[5]

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

0, 1, 2, 3, 4, 5

```
for i in range(len(L)) :  
    print(L[i])
```



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

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

```
for i in L:  
    print(i)
```

2차원 리스트 순차적 접근

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

```
    0, 1  
for i in range(2) :  
    print(L[i])
```

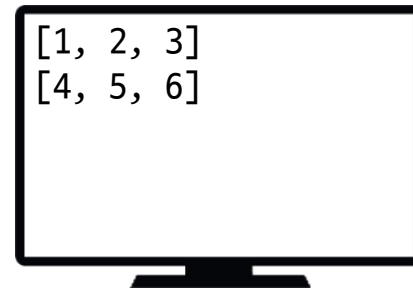
	0열	1열	2열
0행	1	2	3
1행	4	5	6

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

```
    0, 1  
for i in range(len(L)) :  
    print(L[i])
```

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

```
    [[1, 2, 3], [4, 5, 6]]  
for i in L :  
    print(i)
```

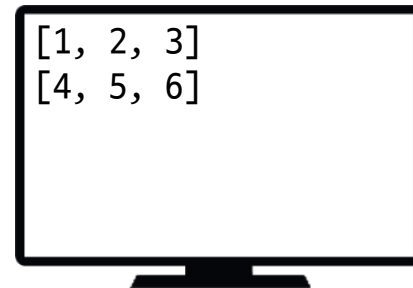


2차원 리스트 순차적 접근

```
L2 = [[1, 2, 3], [4, 5, 6]]  
for i in range(2) :  
    for j in range(3) :  
        print(L2[i][j], end=' ')  
  
    print()
```

	0열	1열	2열
0행	1	2	3
1행	4	5	6

```
L = [[1, 2, 3], [4, 5, 6]]  
for i in range(len(L)) :  
    for j in range(len(L[i])) :  
        print(L[i][j], end=' ')  
  
    print()
```



2차원 리스트 순차적 접근

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

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

```
for i in L :
```

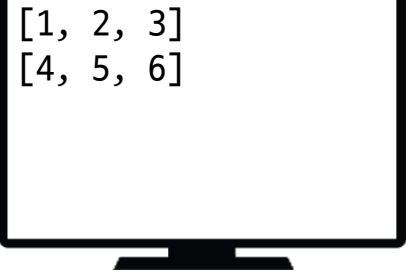
```
    [1, 2, 3], 다음번에는 [4, 5, 6]
```

```
        for j in i :
```

```
            print(j, end=' ')
```

```
print()
```

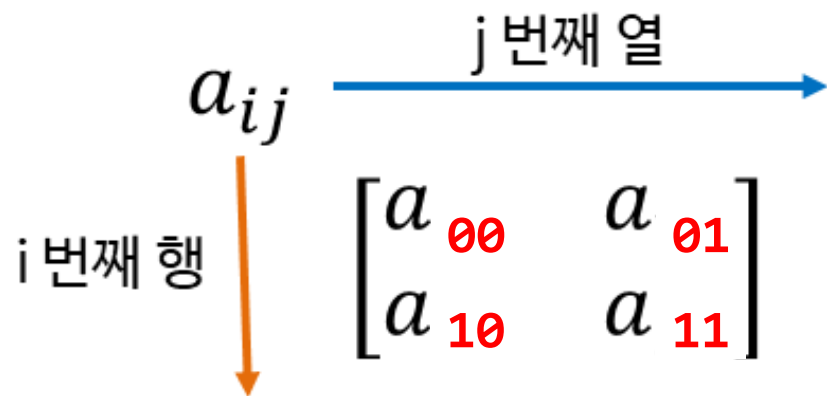
	0열	1열	2열
0행	1	2	3
1행	4	5	6



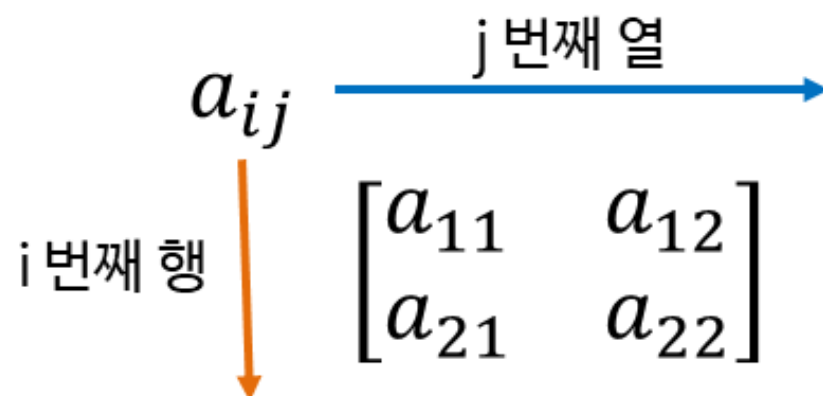
```
[1, 2, 3]  
[4, 5, 6]
```

리스트와 수학의 행렬

파이썬 리스트의 행과 열번호



수학 행렬에서 행과 열번호



리스트 컴프리헨션으로 1차원 리스트 만들기

```
L = []
```

```
for i in range(5):  
    L.append(i)
```

```
print(L)
```

0	1	2	3	4
[0]	[1]	[2]	[3]	[4]

```
L = [ i for i in range(5) ]
```

```
print(L)
```

```
L = list(range(5))
```

```
print(L)
```

```
L = []
```

```
for i in range(5):  
    L.append(0)
```

```
print(L)
```

0	0	0	0	0
[0]	[1]	[2]	[3]	[4]

```
L = [ 0 for i in range(5) ]
```

```
print(L)
```

```
L = [0] * 5
```

```
print(L)
```

리스트 컴프리헨션 → 2차원 리스트 만들기

```
L = []
```

```
for i in range(2) :  
    L.append([])  
    for j in range(5) :  
        L[i].append(j)
```

```
print(L)
```

[0]
[1]

0	1	2	3	4
0	1	2	3	4

[0] [1] [2] [3] [4]

```
L = [ [ j for j in range(5) ] for i in range(2) ]
```

```
print(L)
```

```
L = []
```

```
for i in range(2) :  
    L.append([])  
    for j in range(5) :  
        L[i].append(0)
```

```
print(L)
```

[0]
[1]

0	0	0	0	0
0	0	0	0	0

[0] [1] [2] [3] [4]

```
L = [ [ 0 for j in range(5) ] for i in range(2) ]
```

```
print(L)
```