# 리스트의 이해

7주차_02

한동대학교
김경미 교수

# List(목록 또는 배열)

- **리스트는 값들의 나열(sequence)이다**
  - 리스트 안 구성요소를
    - 원소(elements) 혹은 항목(items)이라고 부른다
    - 다양한 종류의 데이터타입으로 구성 가능하다
  - 리스트의 내용은 변경 가능하다

```
>>> cheeses = ['Cheddar', 'Edam', 'Gouda']
>>> numbers = [17, 123]
>>> print(numbers)
[17, 123]
>>> cheeses[0]
Cheddar
>>> numbers
[17, 123]
```

# 리스트에서 in operator

```
>>> cheeses = ['Cheddar', 'Edam', 'Gouda']
>>> 'Edam' in cheeses
True
>>> 'Brie' in cheeses
False
>>> cheeses
['Cheddar', 'Edam', 'Gouda']
>>> for  food in cheeses :
            print(food)
Cheddar
Edam
Gouda
```

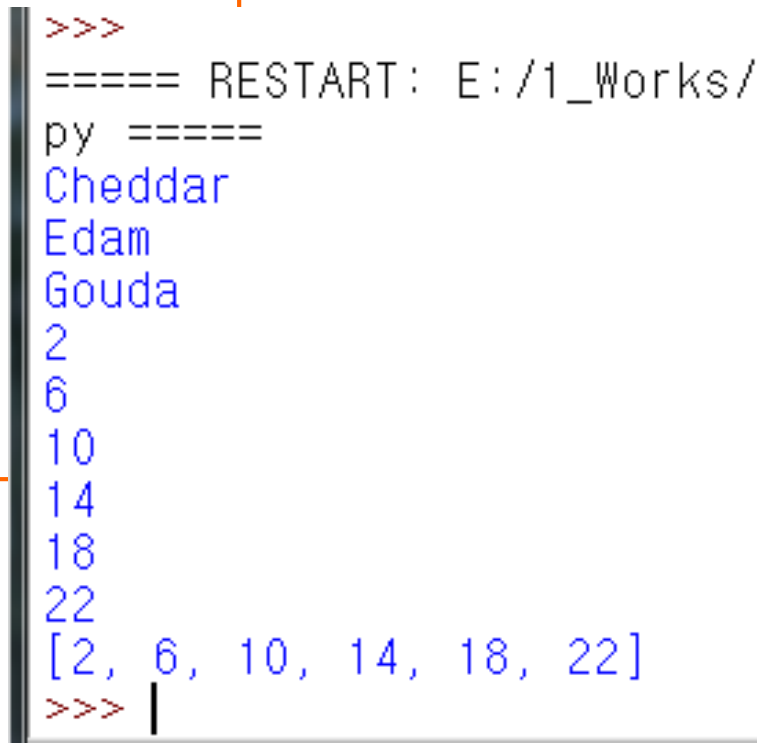
# 리스트 다루기

```
cheeses = ['Cheddar', 'Edam', 'Gouda']
numbers = [1, 3, 5, 7, 9, 11]

for cheese in cheeses :
    print(cheese)

for i in range(len(numbers)) :
    numbers[i] = numbers[i] * 2
    print(numbers[i])

print(numbers)
```



# 리스트, 연산자

- **Operator**
  - +: 두개의 리스트를 합하여 새로운 리스트를 생성
  - *: (리스트) * (정수) 형태로 정수 수만큼 리스트의 내용이 배가된다

```
# The + operator concatenates lists
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> c = a + b
>>> c
[1, 2, 3, 4, 5, 6]

# Similarly, the * operator repeats a list
a given number of times

>>> ['a'] * 4
['a', 'a', 'a', 'a']
>>> a = [1, 2, 3]
>>> a * 3
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

# 리스트 slice

- **Slice**
  - 문자열 사용과 동일
  - index를 사용하여, 리스트 내의 아이템들을 일부 사용한다
  - List_fruit[ :3]
  - 문자열에서 사용하는 것과 동일

```
#list slice
>>> t = ['a', 'b', 'c', 'd', 'e', 'f']
>>> t[1:3]
['b', 'c']
>>> t[:4]
['a', 'b', 'c', 'd']
>>> t[3:]
['d', 'e', 'f']
```

# 연습문제 1

- 새로 생성하는 List의 item 수를 입력 받는다
- list를 생성한다
- item의 수만큼 반복해서 값을 입력받는다
- '+' 연산자를 사용하여 list에 값을 추가한다
- 전체 리스트를 출력한다

# 연습문제 1, 코드와 결과

```
num = int(input("List element 개수 입력: "))
NewList = []
tempList = [0]

for i in range(num):
    print(i, "번째")
    t = input( " 추가할 element 입력: ")
    tempList = [t]
    NewList = NewList + tempList

print(NewList)
```

```
>>>
===== RESTART: E:/1_Works/
py =====
List element 개수 입력: 4
0 번째
추가할 element 입력: 12
1 번째
추가할 element 입력: 34
2 번째
추가할 element 입력: 2
3 번째
추가할 element 입력: 9
['12', '34', '2', '9']
>>>
```

# 리스트, Methods

| | |
|---|---|
| .append | 리스트 내에 새로운 아이템 한 개를 추가하여, 마지막에 위치한다 |
| .insert | 리스트 내에 index 번호와 아이템 내용을 추가한다 |
| .extend | 다른 이름의 리스트, 아이템 모두를 추가한다 |
| .sort | 리스트의 아이템들을 순서대로 정열 , 순서는 ascii code 순이다 |
| .pop | 리스트 내에 존재하는 아이템의 index 번호를 입력 받아 삭제한다 |
| .remove | 리스트 내에 존재하는 아이템을 삭제, 동일한 아이템 있으면 처음 아이템만 삭제 |

# .append

```
# method append

>>> t1 = ['x', 'y', 'z']
>>> t1.append('a')
>>> t1
['x', 'y', 'z', 'a']
>>> t1.append('e')
>>> t1
['x', 'y', 'z', 'a', 'e']
```

# .insert

```
# method insert

>>> t1 = ['x', 'y', 'z']
>>> t1
['x', 'y', 'z']
>>> t1.insert(1, 'a')
>>> t1
['x', 'a', 'y', 'z']
>>> t1.insert(1, 'e')
>>> t1
['x', 'e', 'a', 'y', 'z']
```

# .extend

```
# method extend

>>> t1 = [‘x', ‘y', ‘z']
>>> t2 = ['d', 'e']
>>> t1.extend(t2)
>>> t1
[‘x', ‘y', ‘z', 'd', 'e']
>>> t2.extend(t1)
>>> t2
[‘d', ‘e', ‘x', ‘y', ‘z', 'd', 'e']
```

# .sort

```
#method sort

>>> t = ['d', 'c', 'e', 'b', 'a']
>>> t.sort()
>>> t
['a', 'b', 'c', 'd', 'e']
```

# .pop

```
#method pop(index)

>>> t = ['a', 'b', 'c', 'd', 'e']
>>> x = t.pop(0) + t.pop(1)
>>> t
['c', 'd', 'e']
>>> t.append('a')
>>> t
['c', 'd', 'e', 'a']
```

# .remove

```
# method remove(value)

>>> t = ['a', 'b', 'c']
>>> t.remove('b')
>>> t
['a', 'c']
```

# Method 활용하기 1

```
f1=['apple', 'blueberry', 'melon', 'tomato']
f2=['strawberry', 'lemon', 'banana']
f3=f1+f2
print('f1+f2= ', f3)

f3.append('blackberry')
f3.sort()
print("after sorting = ", f3)
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/turtle_test.py =====
f1+f2=  ['apple', 'blueberry', 'melon', 'tomato', 'strawberry', 'lemon', 'banana']
after sorting =  ['apple', 'banana', 'blackberry', 'blueberry', 'lemon', 'melon',
'strawberry', 'tomato']
>>>
```

# Method 활용하기 2

```
f1=['apple', 'blueberry', 'melon', 'tomato']
f2=['strawberry', 'lemon', 'banana']
f3=f1+f2
print(f3)

#remove element with its first char 'b'
index=len(f3)
i=0

while i < index:
    if f3[i][0] == "b" :
        f3.remove(f3[i])
        i=i-1
        index=index-1
    i=i+1

print("remove all 'b' elements = ", f3)
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Co
de/turtle_test.py =====
list =  ['apple', 'blueberry', 'melon', 'tomato', 'straw
berry', 'lemon', 'banana']
remove all 'b' elements =  ['apple', 'melon', 'tomato',
'strawberry', 'lemon']
>>>
```

# Method 활용하기 2, 설명

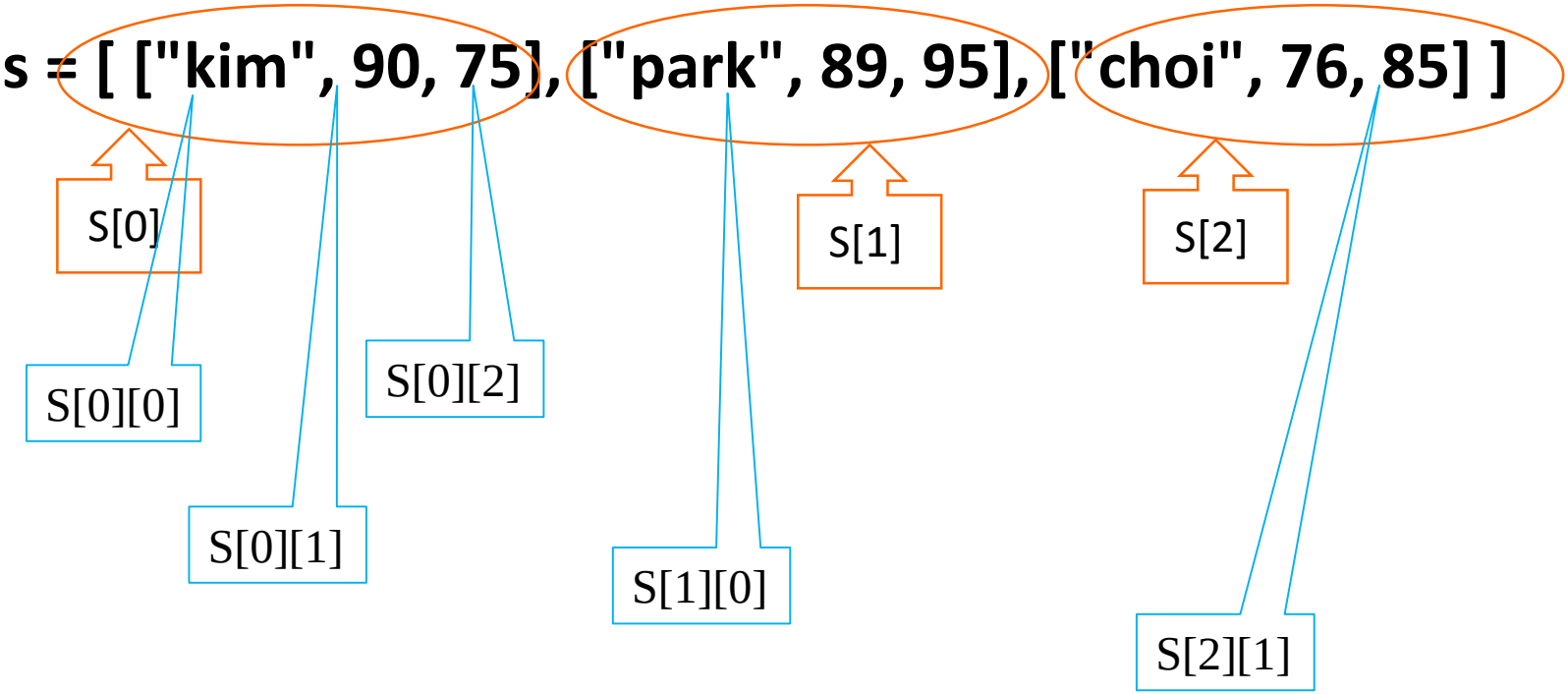
```
#이전 예제 중 일부

while i < index:
    if f3[i][0] == "b" :
        f3.remove(f3[i])
        i=i-1
        index=index-1
    i=i+1
```

- **'b'로 시작하는 문자열을 찾아서 삭제하고 나면**
  - 리스트 f3에 있는 전체 아이템의 개수는 하나 줄어든다(index=index-1)
  - 그 다음 문자열을 확인하기 위해서는 i=i-1 이 필요하다
  - 조건에 맞아서 삭제된 아이템이 가지는 첨자를 그 뒤에 있는 아이템이 가지게 된다

# 2차원 lists

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



# 2차원 리스트 활용

```
fq= [[0,0,0,0,0,0],[0,0,0,0,0,0],[0,0,0,0,0,0],[0,0,0,0,0,0],[0,0,0,0,0,0]]

for i in range(5):
    for j in range(6):
        fq[i][j] = i+j
    print(i,"th row : ", fq[i])

print("="*50)
print("all : ", fq)
print("="*50)
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exe
rcise Code/turtle_test.py =====
0 th row :  [0, 1, 2, 3, 4, 5]
1 th row :  [1, 2, 3, 4, 5, 6]
2 th row :  [2, 3, 4, 5, 6, 7]
3 th row :  [3, 4, 5, 6, 7, 8]
4 th row :  [4, 5, 6, 7, 8, 9]
==================================================
all :  [[0, 1, 2, 3, 4, 5], [1, 2, 3, 4, 5, 6],
[2, 3, 4, 5, 6, 7], [3, 4, 5, 6, 7, 8], [4, 5, 6
, 7, 8, 9]]
==================================================
>>>
```

# 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 2

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

# 연습문제 2, 코드와 결과

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

for i in range(15):
    for j in range(10):
        mul[i][j] = (i+2) * (j+1)
    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!!
>>>
```

# 숙제

- 연습문제 1, 2, method 활용하기 2번의 코드와
- 실행결과를 캡쳐한 사진을 게시판에 올려주세요!

# 요약

- 리스트를 이해한다
- 리스트에서 연산자를 활용한다
- 리스트에서 메소드를 활용한다
- 2차원 리스트를 이해한다

# 감사합니다

7주차_02 리스트의 이해