

파이선으로 배우는 데이터구조

Data Structures Learning with Python

김영훈

한양대학교 ERICA 인공지능학과



List

리스트

리스트 다루기

- ▶ 목록은 다른 데이터 유형으로 얼마든지 구성할 수 있기 때문에 파이썬에서 가장 많이 사용되는 기본 제공 데이터 구조일 것입니다.

| Method | Description |
|--|--|
| <code>list(s)</code> | 시퀀스 <code>s</code> 의 목록을 반환합니다. |
| <code>s.append(x)</code> | <code>s</code> 의 끝에 요소 <code>x</code> 를 추가합니다. |
| <code>s.extend(x)</code> | 리스트 <code>x</code> 를 <code>s</code> 에 추가합니다. |
| <code>s.count(x)</code> | <code>s</code> 에서 <code>x</code> 의 발생 횟수를 계산합니다. |
| <code>s.index(x, [start], [stop])</code> | <code>s[i]==x</code> 에서 가장 작은 인덱스인 <code>i</code> 를 반환합니다. 검색을 위한 선택적 시작 및 중지 인덱스를 포함할 수 있습니다. |
| <code>s.insert(i,e)</code> | 인덱스 <code>i</code> 에 <code>x</code> 를 삽입합니다. |
| <code>s.pop(i)</code> | 요소 <code>i</code> 를 반환하고 목록에서 제거합니다. |
| <code>s.remove(x)</code> | <code>s</code> 에서 <code>x</code> 를 제거합니다. |
| <code>s.reverse()</code> | <code>s</code> 의 순서를 반전시킵니다. |
| <code>s.sort(key ,[reverse])</code> | 선택적 키와 반전을 사용하여 <code>s</code> 를 정렬합니다. |

Mutable!

list()

▶ 리스트를 생성

```
# 1. Create a list of books using list(s)
books = list(["1984", "Brave New World", "Fahrenheit 451"])

# 2. Add a new book to the end of the list using s.append(x)
books.append("To Kill a Mockingbird")

# 3. Add multiple books to the list using s.extend(x)
more_books = ["Moby Dick", "Pride and Prejudice"]
books.extend(more_books)

# 4. Count how many times "1984" appears in the list using s.count(x)
count_1984 = books.count("1984")
print(f"'1984' appears {count_1984} times in the list.")

# 5. Find the index of "Moby Dick" using s.index(x, [start], [stop])
index_moby_dick = books.index("Moby Dick")
print(f"'Moby Dick' is at index {index_moby_dick}.")

# 6. Insert "The Great Gatsby" at the second position using s.insert(i, e)
books.insert(1, "The Great Gatsby")

# 7. Remove the last book from the list using s.pop(i)
last_book = books.pop(-1)
print(f"Removed the last book: {last_book}")
```

append()

- ▶ 리스트 마지막에 항목 하나를 추가

1. Create a list of books using list(s)

```
books = list(["1984", "Brave New World", "Fahrenheit 451"])
```

2. Add a new book to the end of the list using s.append(x)

```
books.append("To Kill a Mockingbird")
```

3. Add multiple books to the list using s.extend(x)

```
more_books = ["Moby Dick", "Pride and Prejudice"]  
books.extend(more_books)
```

4. Count how many times "1984" appears in the list using s.count(x)

```
count_1984 = books.count("1984")  
print(f"'1984' appears {count_1984} times in the list.")
```

5. Find the index of "Moby Dick" using s.index(x, [start], [stop])

```
index_moby_dick = books.index("Moby Dick")  
print(f"'Moby Dick' is at index {index_moby_dick}.")
```

6. Insert "The Great Gatsby" at the second position using s.insert(i, e)

```
books.insert(1, "The Great Gatsby")
```

7. Remove the last book from the list using s.pop(i)

```
last_book = books.pop(-1)  
print(f"Removed the last book: {last_book}")
```

8. Remove "Fahrenheit 451" from the list using s.remove(x)

```
books.remove("Fahrenheit 451")
```

extend()

- ▶ 리스트 마지막에 다른 리스트를 연결하여 덧붙임

```
# 1. Create a list of books using list(s)
books = list(["1984", "Brave New World", "Fahrenheit 451"])
```

```
# 2. Add a new book to the end of the list using s.append(x)
books.append("To Kill a Mockingbird")
```

```
# 3. Add multiple books to the list using s.extend(x)
more_books = ["Moby Dick", "Pride and Prejudice"]
books.extend(more_books)
```

```
# 4. Count how many times "1984" appears in the list using s.count(x)
count_1984 = books.count("1984")
print(f"'1984' appears {count_1984} times in the list.")
```

```
# 5. Find the index of "Moby Dick" using s.index(x, [start], [stop])
index_moby_dick = books.index("Moby Dick")
print(f"'Moby Dick' is at index {index_moby_dick}.")
```

```
# 6. Insert "The Great Gatsby" at the second position using s.insert(i, e)
books.insert(1, "The Great Gatsby")
```

```
# 7. Remove the last book from the list using s.pop(i)
last_book = books.pop(-1)
print(f"Removed the last book: {last_book}")
```

```
# 8. Remove "Fahrenheit 451" from the list using s.remove(x)
books.remove("Fahrenheit 451")
```

```
# 9. Reverse the order of the books using s.reverse()
books.reverse()
```

```
# 10. Sort the books alphabetically using s.sort(key, [reverse])
```

count()

- ▶ 주어진 항목이 리스트에서 몇 번 나오는지 카운트

```
books = list([1984, "Brave New World", "Fahrenheit 451"])  
books.append("To Kill a Mockingbird")
```

```
# 3. Add multiple books to the list using s.extend(x)  
more_books = ["Moby Dick", "Pride and Prejudice"]  
books.extend(more_books)
```

```
# 4. Count how many times "1984" appears in the list using s.count(x)  
count_1984 = books.count("1984")  
print(f"'1984' appears {count_1984} times in the list.")
```

```
# 5. Find the index of "Moby Dick" using s.index(x, [start], [stop])  
index_moby_dick = books.index("Moby Dick")  
print(f"'Moby Dick' is at index {index_moby_dick}.")
```

```
# 6. Insert "The Great Gatsby" at the second position using s.insert(i, e)  
books.insert(1, "The Great Gatsby")
```

```
# 7. Remove the last book from the list using s.pop(i)  
last_book = books.pop(-1)  
print(f"Removed the last book: {last_book}")
```

```
# 8. Remove "Fahrenheit 451" from the list using s.remove(x)  
books.remove("Fahrenheit 451")
```

```
# 9. Reverse the order of the books using s.reverse()  
books.reverse()
```

```
# 10. Sort the books alphabetically using s.sort(key, [reverse])  
books.sort()  
print("Books in alphabetical order:", books)
```

```
# Bonus: Sort the books in reverse alphabetical order
```


index()

- ▶ 주어진 항목이 리스트에서 몇 번째 위치에 (0에서 시작하는 인덱스 번호) 나오는지 찾아서 출력

```
books.extend(more_books)
```

```
# 4. Count how many times "1984" appears in the list using s.count(x)
count_1984 = books.count("1984")
print(f"'1984' appears {count_1984} times in the list.")
```

```
# 5. Find the index of "Moby Dick" using s.index(x, [start], [stop])
index_moby_dick = books.index("Moby Dick")
print(f"'Moby Dick' is at index {index_moby_dick}.")
```

```
# 6. Insert "The Great Gatsby" at the second position using s.insert(i, e)
books.insert(1, "The Great Gatsby")
```

```
# 7. Remove the last book from the list using s.pop(i)
last_book = books.pop(-1)
print(f"Removed the last book: {last_book}")
```

```
# 8. Remove "Fahrenheit 451" from the list using s.remove(x)
books.remove("Fahrenheit 451")
```

```
# 9. Reverse the order of the books using s.reverse()
books.reverse()
```

```
# 10. Sort the books alphabetically using s.sort(key, [reverse])
books.sort()
print("Books in alphabetical order:", books)
```

```
# Bonus: Sort the books in reverse alphabetical order
books.sort(reverse=True)
print("Books in reverse alphabetical order:", books)
```


insert()

- ▶ 주어진 위치 (인덱스 번호)에 주어진 새로운 항목을 끼워넣기

```
print(f"1984 appears {count_1984} times in the list.")  
# 5. Find the index of "Moby Dick" using s.index(x, [start], [stop])  
index_moby_dick = books.index("Moby Dick")  
print(f"'Moby Dick' is at index {index_moby_dick}.")
```

```
# 6. Insert "The Great Gatsby" at the second position using s.insert(i, e)  
books.insert(1, "The Great Gatsby")
```

```
# 7. Remove the last book from the list using s.pop(i)  
last_book = books.pop(-1)  
print(f"Removed the last book: {last_book}")
```

```
# 8. Remove "Fahrenheit 451" from the list using s.remove(x)  
books.remove("Fahrenheit 451")
```

```
# 9. Reverse the order of the books using s.reverse()  
books.reverse()
```

```
# 10. Sort the books alphabetically using s.sort(key, [reverse])  
books.sort()  
print("Books in alphabetical order:", books)
```

```
# Bonus: Sort the books in reverse alphabetical order  
books.sort(reverse=True)  
print("Books in reverse alphabetical order:", books)
```

pop()

- ▶ 주어진 위치 (인덱스 번호)에서 항목 하나를 빼내어 리턴
- ▶ 위치를 주지 않으면 마지막 항목

```
# 6. Insert "The Great Gatsby" at the second position using s.insert(i, e)  
books.insert(1, "The Great Gatsby")
```

```
# 7. Remove the last book from the list using s.pop(i)  
last_book = books.pop(-1)  
print(f"Removed the last book: {last_book}")
```

```
# 8. Remove "Fahrenheit 451" from the list using s.remove(x)  
books.remove("Fahrenheit 451")
```

```
# 9. Reverse the order of the books using s.reverse()  
books.reverse()
```

```
# 10. Sort the books alphabetically using s.sort(key, [reverse])  
books.sort()  
print("Books in alphabetical order:", books)
```

```
# Bonus: Sort the books in reverse alphabetical order  
books.sort(reverse=True)  
print("Books in reverse alphabetical order:", books)
```

remove()

- ▶ 주어진 위치 항목을 제거

```
# 7. Remove the last book from the list using s.pop(i)
```

```
last_book = books.pop(-1)
```

```
print(f"Removed the last book: {last_book}")
```

```
# 8. Remove "Fahrenheit 451" from the list using s.remove(x)
```

```
books.remove("Fahrenheit 451")
```

```
# 9. Reverse the order of the books using s.reverse()
```

```
books.reverse()
```

```
# 10. Sort the books alphabetically using s.sort(key, [reverse])
```

```
books.sort()
```

```
print("Books in alphabetical order:", books)
```

```
# Bonus: Sort the books in reverse alphabetical order
```

```
books.sort(reverse=True)
```

```
print("Books in reverse alphabetical order:", books)
```

reverse()

- ▶ 리스트의 항목의 순서를 뒤집기

```
# 8. Remove "Fahrenheit 451" from the list using s.remove(x)  
books.remove("Fahrenheit 451")
```

```
# 9. Reverse the order of the books using s.reverse()  
books.reverse()
```

```
# 10. Sort the books alphabetically using s.sort(key, [reverse])  
books.sort()  
print("Books in alphabetical order:", books)
```

```
# Bonus: Sort the books in reverse alphabetical order  
books.sort(reverse=True)  
print("Books in reverse alphabetical order:", books)
```

sort()

- ▶ 리스트의 항목을 정렬
 - ▶ 문자열의 경우 알파벳 순서
 - ▶ 숫자의 경우 오름차순
- ▶ `reverse=True` 인자를 주어 내림차순으로 정렬할 수도 있음

```
# 9. Reverse the order of the books using s.reverse()  
books.reverse()
```

```
# 10. Sort the books alphabetically using s.sort(key, [reverse])  
books.sort()  
print("Books in alphabetical order:", books)
```

```
# Bonus: Sort the books in reverse alphabetical order  
books.sort(reverse=True)  
print("Books in reverse alphabetical order:", books)
```



코딩 시작!

연습 문제 풀이


List 1.

- ▶ 목록의 모든 항목을 합산하는 Python 프로그램을 작성하십시오.
- ▶ 입력: [1, 2, -8], 출력: -5

```
def sum_list(items):  
    sum_numbers = 0  
    ?  
    return sum_numbers  
print(sum_list([1,2,-8]))
```

List 2.

- ▶ 목록의 모든 항목을 곱셈하는 Python 프로그램을 작성하십시오.
- ▶ 입력: [1, 2, -8], 출력: -16

```
def multiply_list(items):  
    tot = 1  
      
    return tot  
print(multiply_list([1,2,-8]))
```

List 3.

- ▶ 목록에서 가장 큰 수를 출력하는 프로그램을 작성하십시오.
- ▶ 입력: [1, 2, -8, 0], 출력: 2

```
def max_num_in_list( list ):
    max = list[ 0 ]
    ?
    return max
print(max_num_in_list([1, 2, -8, 0]))
```

List 4.

- ▶ 임의의 두 목록이 주어졌을 때, 이들을 이어붙인 목록을 만드는 코드를 작성하십시오.
- ▶ 입력: [1, 2, 3, 0], ["Red", "Green", "Black"]
- ▶ 출력: [1, 2, 3, 0, "Red", "Green", "Black"]

```
list1 = [1, 2, 3, 0]
list2 = ['Red', 'Green', 'Black']
final_list = ?
print(final_list)
```

List 5.

- ▶ 목록에서, 특정 원소의 인덱스를 찾는 코드를 작성하십시오.
- ▶ 입력: [10, 30, 4, -6], 30
- ▶ 출력: 1

```
num = [10, 30, 4, -6]  
print(num.?(30))
```

List 6.

- ▶ 1부터 30까지 숫자로 구성된 목록에서, 첫 5개 원소와 마지막 5개 원소를 출력하는 코드를 작성하십시오.

```
my_list = list(range(1, 31))  
print(my_list[?])  
print(my_list[?])
```


List 7.

- ▶ 1부터 30까지 숫자로 구성된 목록에서, 10부터 19까지 숫자를 출력하는 코드를 작성하십시오.

```
my_list = list(range(1, 31))  
print(my_list[?])
```

List 8.

- ▶ 1부터 10까지 숫자로 구성된 목록에서, 홀수만으로 구성된 목록을 만드는 코드를 작성하십시오.

```
my_list = list(range(1, 11))  
print(my_list[?])
```

List 9.

- ▶ 1부터 30까지 숫자로 구성된 목록에서, 10부터 19 사이 수 중 홀수만으로 구성된 코드를 작성하십시오.

```
my_list = list(range(1, 31))  
print(my_list[?])
```

List 10.

- ▶ 같은 길이의 두 개의 목록이 주어졌을 때, 두 목록의 원소를 차례대로 동시에 출력하는 코드를 작성하십시오.
- ▶ 입력:
 - ▶ [1, 2, 3], ["red", "white", "black"]
 - ▶ 출력:
 - ▶ 1, red
 - ▶ 2, white
 - ▶ 3, black

```
num = [1, 2, 3]
color = ['red', 'white', 'black']
```

?

List 11.

- ▶ 주어진 두 개의 숫자 목록을 나누어 새 목록을 만드는 Python 프로그램을 작성하십시오. (zip의 사용법)

```
def dividing_two_lists(l1,l2):  
    result = [x/y for x, y in zip(l1,l2)]  
    return result  
nums1 = [7,2,3,4,9,2,3]  
nums2 = [9,8,2,3,3,1,2]  
print(dividing_two_lists(nums1, nums2))
```

List 12.

- ▶ 목록이 주어졌을 때, 목록의 원소를 인덱스와 함께 차례로 출력하는 코드를 작성하십시오.
- ▶ 입력: ["red", "white", "black"]
- ▶ 출력:
- ▶ 0, red
- ▶ 1, white
- ▶ 2, black

```
color = ['red', 'white', 'black']
```

?

List 13.

- ▶ 숫자 또는 문자로 구성된 목록이 주어졌을 때, 가장 작은 값을 가지는 5개의 원소를 출력하는 코드를 작성하십시오. (sort 사용법)
- ▶ 입력: [10, 4, 13, 134, 1, 9, 102, 31, 42], 출력: [1, 4, 9, 10, 13]

```
A = [10, 4, 13, 134, 1, 9, 102, 31, 42]
B = sorted(A)[:5]
print(B)
```


List 14.

- ▶ 서브목록들의 목록에서, 각 서브목록의 길이순으로 서브목록을 정렬하는 코드를 작성하십시오. (sort의 사용법)
- ▶ 입력: `[[1, 2, 3], [4, 5, 6, 7], [7, 8]]`
- ▶ 출력: `[[7, 8], [1, 2, 3], [4, 5, 6, 7]]`

```
A = [[1, 2, 3], [4, 5, 6, 7], [7, 8]]  
B = sorted(A, key=len)  
print(B)
```


List 15.

- ▶ 단어들의 목록에서 길이가 n 보다 긴 단어들로 구성된 리스트를 반환하는 함수를 작성하십시오.
- ▶ 입력: $n=3$, ["The", "quick", "brown", "fox"], 출력: ["quick", "brown"]

```
def long_words(n, ls):  
    word_len = []  
      
    return word_len  
print(long_words(3, ["The", "quick", "brown", "fox"]))
```

List 16.

- ▶ 두 목록이 주어졌을 때, 공통인자가 있으면 True, 없으면 False를 반환하는 함수를 작성하십시오.

```
def common_data(list1, list2):  
    result = False  
      
    return result  
print(common_data([1,2,3,4,5], [5,6,7,8,9]))  
print(common_data([1,2,3,4,5], [6,7,8,9]))
```

List 17.

- ▶ 목록에서 등장하는 원소로 구성된 목록을 출력하는데, 두 번 이상 등장하는 원소는 한 번만 출력하십시오(unique list라고 부름).
- ▶ 입력: [10, 20, 30, 40, 20, 50, 60, 40]
- ▶ 출력:[10, 20, 30, 40, 50, 60]

```
my_list = [10, 20, 30, 40, 20, 50, 60, 40]
my_new_list = []
print("Original List : ",my_list)
```

?

```
print("List of unique numbers : ",my_new_list)
```

List 18.

- ▶ 목록에서 연속된 중복을 제거한 목록을 출력하는 코드를 작성하십시오.
- ▶ 입력: [1, 2, 3, 4, 4, 5, 6, 6, 6]
- ▶ 출력: [1, 2, 3, 4, 5, 6]

```
ls = [1, 2, 3, 4, 4, 5, 6, 6, 6]  
new_ls = []
```

```
?
```

```
print(new_ls)
```


List 19.

- ▶ 6개의 숫자로 구성된, 2x3짜리 2차원 목록에서 각 행의 합을 출력하는 코드를 작성하십시오.
- ▶ 입력: `[[1, 2, 3], [4, 5, 6]]`
- ▶ 출력: `6, 15`

```
A = [[1, 2, 3], [4, 5, 6]]  
results = []
```

?

```
print(results)
```

List 20.

- ▶ 여러개의 목록이 주어졌을 때, 각 목록의 원소를 다 이어 붙여서 하나의 목록으로 만드는 코드를 작성하십시오. (힌트: extend)
- ▶ 입력: `[[1, 2, 3], [4, 5, 6], [7, 8]]`
- ▶ 출력: `[1, 2, 3, 4, 5, 6]`

```
A = [[1, 2, 3], [4, 5, 6], [7, 8]]  
B = []  


?

  
print(B)
```

List 21.

- ▶ 서브목록들의 목록에서, 각 서브목록의 첫 번째 원소들의 합을 출력하는 코드를 작성하십시오. (한 줄에 for문 작성하기)
- ▶ 입력: `[[1, 2], [2, 3, 4], [8, 4], [10, 11, 1, 2], [3, 10]]`
- ▶ 출력: 24

```
lst = [[1, 2], [2, 3, 4], [8, 4], [10, 11, 1, 2], [3, 10]]  
first_elems = ?  
print(sum(first_elems))
```

List 22.

- ▶ 서브목록들의 목록에서, 각 서브목록의 마지막 원소들 중, 값이 3보다 큰 원소들의 합을 출력하는 코드를 작성하십시오.
- ▶ 입력: `[[1, 2], [2, 3, 4], [8, 4], [10, 11, 1, 2], [3, 10]]`
- ▶ 출력: 18

```
lst = [[1, 2], [2, 3, 4], [8, 4], [10, 11, 1, 2], [3, 10]]  
last_elems = ?  
print(sum(last_elems))
```

List 23.

- ▶ 목록들의 목록에서, 일정 범위의 값만 포함하는 목록들의 목록을 계산하는 함수를 작성하십시오.
- ▶ 출력: [[13, 14, 15, 17]]

```
def remove_list_range(input_list, left_range, righth_range):  
    ?  
    return result  
list1 = [[2], [0], [1, 2, 3], [0, 1, 2, 3, 6, 7], [9, 11],  
         [13, 14, 15, 17]]  
left_range = 13  
righth_range = 17  
print(remove_list_range(list1, left_range, righth_range))
```

List 24.

- ▶ 세 개의 목록이 주어졌을 때, 공통된 원소를 가진 인덱스를 출력하는 함수를 작성하십시오. (힌트: zip)
- ▶ 출력: [1, 7]

```
def extract_index_ele(l1, l2, l3):  
    result = []
```

?

```
    return result
```

```
nums1 = [1, 1, 3, 4, 5, 6, 7]  
nums2 = [0, 1, 2, 3, 4, 5, 7]  
nums3 = [0, 1, 2, 3, 4, 5, 7]  
print(extract_index_ele(nums1, nums2, nums3))
```

List 25.

- ▶ 목록이 주어졌을 때, 특정 구간을 뒤집는(reverse) 함수를 작성하십시오.
- ▶ 출력: [10, 20, 50, 40, 30, 60, 70, 80]

```
def reverse_list_in_location(lst, start_pos, end_pos):
```

?

```
    return lst
```

```
nums = [10,20,30,40,50,60,70,80]
```

```
start_pos = 2
```

```
end_pos = 4
```

```
print(reverse_list_in_location(nums,start_pos,end_pos))
```

List 26.

- ▶ 주어진 목록에서 연속된 모든 항목 쌍의 목록을 계산하는 Python 프로그램을 작성하십시오.
- ▶ 입력: [1, 1, 2, 3, 3, 4, 4, 5]
- ▶ 출력: [(1, 1), (1, 2), (2, 3), (3, 3), (3, 4), (4, 4), (4, 5)]

```
def pairwise(l1):  
    temp = []  
    ?  
    return temp  
l1 = [1,1,2,3,3,4,4,5]  
print(pairwise(l1))
```


List 27.

- ▶ 3x4 행렬이 2차원 목록으로 주어졌을때, 특정 열에 있는 원소의 합을 출력하는 코드를 작성하십시오.
- ▶ 입력: [1, 1, 2, 3, 3, 4, 4, 5]
- ▶ 출력: [(1, 1), (1, 2), (2, 3), (3, 3), (3, 4), (4, 4), (4, 5)]

```
def sum_column(nums, C):  
    result = ?  
    return result  
  
nums = [[1,2,3,2],  
        [4,5,6,2],  
        [7,8,9,5]]  
  
column = 0  
print(sum_column(nums, column))
```

List 28.

- ▶ 주어진 목록 목록에서 하위 목록의 최대 합계와 최소 합계를 계산하는 Python 프로그램을 작성하십시오. (힌트: max, min, key)
- ▶ 출력: 14, 6

```
def max_min_sublist(lst):
```

```
    ?
```

```
    return max_result, min_result
```

```
nums = [[1,2,3,5], [2,3,5,4], [0,5,4,1], [3,7,2,1],  
        [1,2,1,2]]
```

```
result = max_min_sublist(nums)
```

List 29.

- ▶ 주어진 목록에서 두 개의 하위 목록을 교환하는 Python 프로그램을 작성하십시오.
- ▶ 출력: [0, 4, 5, 3, 4, 1, 2, 6, 7, 8, 9, 10]

```
nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  
      ?  
print("\nSwap two sublists of the said list:")  
print(nums)
```

List 30.

- ▶ 주어진 행렬의 차원을 찾는 Python 프로그램을 작성하십시오.
- ▶ 출력: 2, 2

```
def matrix_dimensions(test_list):
```

```
    ?
```

```
    return row,column
```

```
lst = [[1,2],[2,4]]  
print("Dimension of the said matrix:")  
print(matrix_dimensions(lst))
```