

주요 데이터 구조 - 리스트

리스트(List)

- 파이썬에서 가장 자주 사용되는 데이터 구조 중 하나
- 여러 값들을 순서대로 저장하는데 사용
- 리스트는 여러 원소를 순서대로 포함하며, 각 원소는 인덱스를 통해 접근
- 인덱스는 0부터 시작하여 리스트의 각 원소에 대응

특징

- 순서가 있는 컬렉션
- 요소의 추가, 삭제, 수정이 **자유로움**
- 동일한 요소의 중복 저장 가능
- 다양한 데이터 타입을 하나의 리스트에 저장할 수 있음

```
fruits = ["사과", "바나나", "오렌지"] numbers = [1, 2, 3, 4, 5] mixed = [1, "안녕", 3.14, True]
print(f"과일 리스트: {fruits}") print(f"숫자 리스트: {numbers}") print(f"혼합 리스트: {mixed}")
print(f"과일 첫번째 과일 : {fruits[0]}") print(f"과일 두번째 과일 : {fruits[1]}") print(f"과일 세번째 과일 : {fruits[2]}")
for fruit in fruits: print(f"과일: {fruit}")
print(len(fruits)) # 리스트의 길이
for i in range(3): print(f"과일: {fruits[i]}")
for i in range(len(fruits)): print(f"과일 {fruits[i]}")
for i in range(len(fruits)): print(f"{i+1}번째 과일 {fruits[i]}")
fruits = ["사과", "바나나", "오렌지"] #인덱스와 데이터를 함께 출력
for i, fruit in enumerate(fruits): print(f"{i+1}번째 과일: {fruit}")
```

리스트 요소 수정

- 리스트의 요소를 수정하려면, 해당 요소의 인덱스를 사용하여 새 값을 할당
- 리스트가 변경 가능(mutable)한 데이터 타입이기 때문에 가능한 작업

```
my_list = [10, 20, 30, 40, 50] my_list[1] = 25 # 두 번째 요소 값을 25로 수정
print(my_list) # 결과: [10, 25, 30, 40, 50] my_list[4] = 100 # 다섯 번째 요소 값을 100으로 수정
print(my_list) # 결과: [10, 25, 30, 40, 100]
```

리스트 결합

- 리스트에서 `+` 연산자를 사용하는 것은 두 개 이상의 리스트를 결합하여 새로운 리스트를 생성하는 방법
- 이 연산자는 기존의 리스트들은 변경하지 않고, 새로운 리스트를 반환

```
old_list = [5, 6, 7] new_list = [1, 2, 3, 4] result = old_list + new_list
print(result) # 출력: [5, 6, 7, 1, 2, 3, 4]
```

리스트 요소 추가 및 삭제

- 영구적인 추가를 위해 `append()` 사용

```
my_list = [] my_list = [10, 20, 30, 40, 50] my_list.append(6) # 리스트 끝에 6 추가
print(my_list) my_list.insert(1, 9) # 인덱스 1에 9 삽입
print(my_list) my_list.remove(20) # 데이터를 날린다.
print(my_list) del my_list[0] # 첫 번째 인덱스 기준 요소 삭제
print(my_list)
```

함수	설명	예시
<code>append(x)</code>	리스트의 끝에 요소 <code>x</code> 를 추가	<code>lst.append(4)</code>
<code>extend(iterable)</code>	<code>iterable</code> 의 모든 항목을 리스트에 추가	<code>lst.extend([4, 5])</code>
<code>insert(i, x)</code>	지정한 위치 <code>i</code> 에 요소 <code>x</code> 를 삽입	<code>lst.insert(2, 'a')</code>
<code>remove(x)</code>	리스트에서 첫 번째로 나오는 요소 <code>x</code> 를 제거	<code>lst.remove(3)</code>
<code>pop([i])</code>	지정된 위치 <code>i</code> 의 요소를 제거하고 반환 (기본은 마지막 요소)	<code>lst.pop()</code> or <code>lst.pop(1)</code>
<code>clear()</code>	리스트의 모든 요소를 제거	<code>lst.clear()</code>
<code>index(x[, start[, end]])</code>	요소 <code>x</code> 의 첫 번째 인덱스를 반환	<code>lst.index(2)</code>
<code>count(x)</code>	리스트에서 요소 <code>x</code> 의 개수를 반환	<code>lst.count(2)</code>
<code>sort(key=None, reverse=False)</code>	리스트를 정렬	<code>lst.sort()</code>
<code>reverse()</code>	리스트의 요소 순서를 반대로 변경	<code>lst.reverse()</code>
<code>copy()</code>	리스트의 얇은 복사본을 반환	<code>new_list = lst.copy()</code>

```
# 아래는 5명의 학생들의 성적을 입력 받아서 # 최고값, 최소값, 평균값, 90점 이상의 count 프로그램
# 5명의 학생들의 성적을 입력 -> list [] 에 저장 STUDENTS = 5
lst = [] count = 0
for i in range(STUDENTS):
    value = int(input(f"{i+1}번째 학생의 성적을 입력하세요."))
    lst.append(value)
# 최고값, 최소값, 평균값 계산
max, min, sum
print(f"최대 점수 : {max(lst)}")
print(f"최대 점수 : {min(lst)}")
print(f"평균 점수 : {sum(lst)/len(lst)}")
```