

<[파이썬 수학] 제7장. 마법 주머니의 특별한 기술들!>

선생님: (마법 주머니를 흔들며) 애들아, 안녕! 지난 시간에 배운 우리의 마법 주머니, 리스트는 그냥 물건을 담기만 하는 게 아니란다. 이 주머니 자체가 아주 똑똑해서, 내용물을 스스로 정리하거나, 개수를 세거나, 물건을 추가하는 등 여러 가지 특별한 기술(메소드)을 가지고 있지!

<1. 마법 주머니의 기본 기술들 (List Methods)>

선생님: 먼저, 가장 많이 쓰이는 기본 기술들을 만나볼까?

기술 1: `.append()` - 주머니에 물건 하나 더 넣기!

```
fruits = ["사과", "바나나", "애플", "망고"]

print("기술 사용 전:", fruits)

fruits.append("토마토") # "토마토"를 주머니 맨 뒤에 쏙!

fruits.append(100)      # 숫자도 쏙!

print("append 기술 사용 후:", fruits)
```

기술 2: `.count()` - 주머니 안에 이 물건이 몇 개나 들었지?

```
names = ["왕새우", "유재석", "이성주", "정재윤", "정재훈", "유재석"]

name_count = names.count("유재석") # "유재석"이라는 이름이 몇 개인지 세어줘!

print("\n'유재석'은 몇 명?", name_count)
```

기술 3: `.sort()` - 주머니 속 물건들을 가나다/크기 순으로 정리해!

기술 4: `.reverse()` - 주머니 속 물건 순서를 반대로 뒤집어!

```
nums = [100, 34, 56, 45, 80, 90]

print("\n정리 전 숫자들:", nums)

nums.sort() # 오름차순(작은 수부터 큰 수 순서)으로 정리!

print("sort 기술 사용 후:", nums)

nums.reverse() # 현재 순서를 그대로 뒤집기! (결과적으로 내림차순이 되겠지?)

print("reverse 기술 사용 후:", nums)
```

코딩새싹 🌱: 와! `.append()` 하니까 정말 맨 뒤에 추가되네요! `.sort()`는 순서대로, `.reverse()`는 거꾸로!
정말 주머니가 스스로 정리하는 것 같아요!

<2. 중요한 비밀: 제자리에서 일하는 기술 vs 새로운 걸 주는 기술>

궁금이 💡: 선생님, 아까 코드에 `a = names.sort()` 라는 부분이 있었는데, 그 다음에 `print(a)`를 하면 뭐가 나와요? 정렬된 리스트가 나올까요?

선생님: 오! 오늘 배울 내용 중 가장 중요한 질문이야! 다들 주목! `.sort()`나 `.reverse()` 같은 기술들은 아주 특이한 성격이 있어. 마치 방을 청소하는 로봇 같달까? 내가 “방 청소해!”라고 명령하면, 로봇은 방을 깨끗하게 만들 뿐, 나에게 무언가를 가져다주지는 않지?

```
names = ["왕새우", "유재석", "이성주", "정재윤", "정재훈", "유재석"]
```

```
print("\n정렬 전 names:", names)
```

```
a = names.sort() # names 리스트 자체를 제자리에서 정렬시켜! 그리고 아무것도 돌려주지 않아.
```

```
print("정렬 후 names:", names) # names 리스트는 바뀌었지만,
```

```
print("a에는 뭐가 들어있을까?:", a) # a는 받은 게 없어서 None (아무것도 없음)이 출력돼!
```

코딩새싹 🌱: 앗! `a`를 출력하니까 정말 `None`이 나왔어요! `names` 리스트는 정렬이 됐는데!

선생님: 맞아. `.sort()`는 리스트 그 자체를 바꿔버리는 ‘제자리에서 일하는 기술’이야. 만약 원본은 그대로 두고, 정렬된 새로운 리스트를 만들고 싶다면 어떻게 할까? 두 가지 방법이 있어.

방법 1: 주머니 복사해서 정리하기!

```
names1 = ["왕새우", "유재석", "이성주"]
```

```
names2 = names1 + [] # 빈 리스트를 더해서 똑같은 주머니를 하나 더 복사!
```

```
names2.sort() # 복사한 주머니만 정리!
```

```
print("\n원본 names1:", names1) # 원본은 그대로!
```

```
print("복사해서 정렬한 names2:", names2)
```

방법 2: `sorted()` 라는 공용 도구(함수) 사용하기!

`sorted()` 함수는 원본은 절대 건드리지 않고, 정렬된 새로운 리스트를 만들어서 우리에게 돌려준단다.

```
nums = [100, 34, 56, 45, 80, 90]
```

```
new_nums = sorted(nums) # sorted() 함수는 정렬된 '새 리스트'를 돌려줘!
```

```
print("\n원본 nums:", nums)
```

```
print("sorted() 함수로 만든 새 리스트:", new_nums)
```

궁금이 💡: 아하! .sort() 메소드는 리스트 자체를 바꿔버리고, sorted() 함수는 원본은 놔두고 새로운 결과물을 주는 거군요! 이제 확실히 알겠어요!

<3. 주머니 속의 주머니: 중첩 리스트(Nesting Lists)>

선생님: 우리의 마법 주머니는 너무나도 위대해서, 주머니 안에 또 다른 작은 주머니들을 담을 수도 있단다! 이걸 중첩 리스트라고 해!

학생들의 성적을 담은 중첩 리스트

```
my_class = [
    ["이름", "국어점수", "수학점수", "영어점수", "평균"], # 0번 주머니 (제목들)
    ["홍길동", 50, 80, 70], # 1번 주머니 (홍길동 정보)
    ["데카르트", 100, 100, 100], # 2번 주머니 (데카르트 정보)
    ["맹구", 10, 20, 10] # 3번 주머니 (맹구 정보)
]
```

코딩새싹 🌱: 와! 리스트 안에 리스트가 들어있어요! 그럼 저기서 '수학'이라는 글자만 꺼내려면 어떻게 해요?

선생님: 이중 인덱싱을 쓰면 되지! 먼저 큰 주머니에서 원하는 작은 주머니를 꺼내고, 그 작은 주머니에서 다시 원하는 물건을 꺼내는 거야!

'수학'은 0번 주머니 안에 들어있는 1번 물건이지!

```
print("\n'수학' 꺼내기:", my_class[0][1])
```

그럼 '데카르트'의 '수학점수'를 이용해서 문장을 만들어볼까?

이름: 데카르트, 수학점수: 100점

```
print(f"{my_class[0][0]}: {my_class[2][0]}, {my_class[0][2]}점수: {my_class[2][2]}점")
```

<도전! 중첩 리스트 실전 퀴즈!>

선생님: 자, 이제 마지막 도전! 홍길동 학생과 맹구 학생의 점수 평균을 계산해서, 각 학생의 정보가 담긴 작은 주머니에 .append() 기술로 추가해보자!

궁금이 💡: 좋아요! 먼저 홍길동 학생의 점수들만 꺼내야겠어요!

1. 홍길동의 점수들만 슬라이싱으로 꺼내기! (1번 주머니의 1번~3번 물건)

```
scores = my_class[1][1:4] # [50, 80, 70]
```

```
print("\n홍길동의 점수들:", scores)
```

```
# 2. sum()과 len() 함수로 평균 계산하기!
```

```
total = sum(scores)
```

```
avg = total / len(scores)
```

```
# 3. round() 함수로 소수점 정리하고, append() 기술로 홍길동 주머니에 추가!
```

```
my_class[1].append(round(avg, 2))
```

```
print("홍길동 평균 추가 후:", my_class[1])
```

```
# --- 맹구도 똑같이 해볼까? ---
```

```
m_scores = my_class[3][1:] # 1번부터 끝까지!
```

```
m_avg = sum(m_scores) / len(m_scores)
```

```
my_class[3].append(round(m_avg, 2))
```

```
print("맹구 평균 추가 후:", my_class[3])
```

```
print("\n--- 최종 결과 ---")
```

```
print(my_class)
```

코딩새싹 🌱: 와! 정말 평균 점수가 각 학생 리스트의 맨 뒤에 추가됐어요! 우리가 배운 인덱싱, 슬라이싱, 함수, 메소드를 전부 다 쓴 것 같아요!

선생님: (활짝 웃으며) 바로 그거야! 오늘 우리는 리스트가 가진 여러 특별한 기술들과, 리스트 안에 리스트를 담는 법, 그리고 그것들을 활용해 복잡한 데이터를 처리하는 방법까지 배웠어. 이제 여러분은 이 마법 주머니, 리스트를 정말 자유자재로 다룰 수 있게 된 거란다! 오늘 정말 멋진 탐험이었어! 🌟