

<[파이썬 수학] 제4장. 마법 주문과 특별 능력! 함수와 메소드>

선생님: (공구함과 장난감 로봇을 양손에 들고) 애들아, 안녕! 지난 시간에 우리는 `2 + '3'`처럼 타입이 다른 친구들은 계산할 수 없어서, `2 + int('3')`처럼 형변환 마법을 써야 한다고 배웠지? 그럼 이건 어떨까?

왜 에러가 "안" 날까요?

```
a = "3"
b = 7
print(a + str(b))
```

코딩새싹 🌱: 어? 지난번엔 에러가 났었는데... 이번엔 `str(b)`로 숫자 7을 글자 '7'로 변신시켰군요! 그래서 글자 기차 "3"이랑 글자 기차 '7'이 연결돼서 "37"이 나온 거예요!

선생님: 정확해! 지난 시간에 배운 형변환 마법을 완벽하게 이해했구나! 이처럼 우리는 `int()`, `str()`, `type()`, `print()` 같은 마법 주문들을 써왔어. 이런 주문들을 파이썬 세계에서는 **함수(Function)**라고 부른다. 특히 파이썬이 기본적으로 제공하는 이런 함수들을 **내장 함수(built-in function)**라고 하지. 마치 이 공구함에 들어있는 망치나 드라이버처럼, 누구나 필요할 때 꺼내 쓸 수 있는 공용 도구 같은 거야.

궁금이 💡: 공용 도구요? 그럼 공용이 아닌 도구도 있나요?

선생님: 아주 예리한 질문이야! 바로 그게 오늘 배울 **메소드(Method)**란다. 메소드는 특정 물건(객체)만 쓸 수 있는 전용 특별 능력 같은 거야.

<객체, 속성, 그리고 메소드>

선생님: 파이썬 세상에 있는 모든 것, 예를 들어 우리가 만든 숫자, 문자열, 리스트 등은 모두 **객체(object)**라고 불러. 그리고 이 객체들은 두 가지를 가지고 있지. 바로 **속성(attribute)**과 **메소드(method)**야.

- **속성:** 그 객체의 정보나 상태. (예: 사람의 '이름', '키', '나이')
- **메소드:** 그 객체가 할 수 있는 특별한 행동이나 기능. (예: 사람이 '달린다()', '말한다()')

예를 들어, 이 로봇(객체)이 있다면, 이 로봇의 색깔이나 만들어진 날짜는 '속성'이고, 이 로봇만 할 수 있는 '레이저 발사하기()', '하늘 날기()' 같은 건 '메소드'가 되는 거지!

궁금이 💡: 아하! 그럼 함수는 공용 공구함에 있는 도구고, 메소드는 특정 객체에만 달려있는 특별한 버튼 같은 거네요!

선생님: 완벽한 비유야! 그래서 메소드를 쓸 때는 반드시 "어떤 객체의" 특별 능력인지 밝혀줘야 해. 바로

객체.메소드() 이렇게 점(.)을 찍어서 말이지!

<문자열 객체의 특별한 능력, 문자열 메소드!>

선생님: 자, 그럼 우리의 글자 기차, 즉 문자열 객체는 어떤 특별 능력을 가지고 있는지 만나볼까?

```
my_string = "Hello World"
```

```
# 능력 1: .upper() - 모든 글자를 큰 소리로 외치기! (대문자로 바꾸기)
```

```
print("대문자로 외치기:", my_string.upper())
```

```
# 능력 2: .lower() - 모든 글자를 속삭이기! (소문자로 바꾸기)
```

```
print("소문자로 속삭이기:", my_string.lower())
```

```
# 능력 3: .count('o') - 'o'라는 글자가 몇 개나 있는지 세어보기!
```

```
print("'o' 글자 개수 세기:", my_string.count("o"))
```

```
# 능력 4: .split() - 띄어쓰기를 기준으로 글자 기차를 여러 칸으로 싹둑! 나누기!
```

```
print("띄어쓰기로 나누기:", my_string.split())
```

코딩새싹 🌱: 와! my_string.upper() 하니까 정말 다 대문자로 바뀌었어요! 점(.)을 찍어서 “my_string아, 너의 어퍼 능력을 보여줘!” 하고 명령하는 것 같아요! split()을 하니까 리스트가 되었네요!

선생님: 맞아! split() 메소드는 문자열을 특정 기준으로 나눠서 리스트로 만들어주는 아주 유용한 능력이란다. 꼭 띄어쓰기가 아니더라도 우리가 기준을 정해줄 수 있어.

```
fish = "새우-오징어-갈치"
```

```
# 이번엔 '-'를 기준으로 나눠볼까?
```

```
fish_list = fish.split("-")
```

```
print(fish_list)
```

```
print("리스트의 첫 번째 항목:", fish_list[0]) # 이제 리스트가 되었으니 인덱싱도 가능!
```

궁금이 💡: len() 같은 함수는 len(my_string) 이렇게 쓰는데, count() 같은 메소드는 my_string.count('o') 이렇게 쓰는 이유를 이제 알겠어요! len()은 문자열의 길이도 재고, 리스트의 길이도 잴 수 있는 공용 도구지만,

count()는 문자열 객체에 속한 특별한 세기 능력이기 때문이군요!

선생님: (감탄하며) 바로 그거야! 그 차이를 이해했다면 오늘 수업의 핵심을 깨우친 거란다!

<도전! 함수와 형변환 콜라보 퀴즈!>

선생님: 자, 그럼 오늘 배운 것과 지난 시간에 배운 것을 합쳐서 문제를 해결해볼까?

[퀴즈] 숫자 23459가 몇 자리 수인지 알아내시오!

힌트: 글자의 길이를 재는 공용 도구 len() 함수를 쓰고 싶는데... len()은 숫자에 쓸 수 있을까?

코딩새싹 🌱: 으음... len()은 글자 기차의 길이를 재는 거니까 숫자에겐 바로 쓸 수 없을 것 같아요! 그럼...
아! 변신 마법!

궁금이 💡: 맞아요! 숫자 23459를 str() 마법으로 글자 기차 "23459"로 변신시킨 다음에, len()이라는 공용
도구(함수)로 길이를 재면 되겠어요!

선생님: 정답! 완벽한 해결책이야!

```
num = 23459
```

```
# 1단계: 숫자를 문자열로 형변환!
```

```
str_num = str(num)
```

```
# 2단계: 문자열이 됐으니 len() 함수로 길이를 잰다!
```

```
num_length = len(str_num)
```

```
print(f"숫자 {num}은(는) {num_length}자리 수입니다!")
```

```
# 이걸 한 줄로 합치면?
```

```
print(f"숫자 {num}은(는) {len(str(num))}자리 수입니다!")
```

코딩새싹 🌱: 와! 형변환 마법이랑 함수를 같이 쓰니까 이런 문제도 풀 수 있네요! 정말 신기해요!

선생님: (미소 지으며) 그렇지? 오늘 우리는 누구나 쓸 수 있는 강력한 공용 도구 '함수'와, 특정 데이터만 쓸 수 있는 '메소드'에 대해 배웠어. 이 둘의 차이를 이해하고, 형변환 마법과 함께 적절히 사용하면 여러분은 파이썬 세상의 어떤 문제든 해결할 수 있는 훌륭한 문제 해결사가 될 수 있을 거란다! 오늘 탐험도 정말 멋졌어! 🌟