

<[파이썬 수학] 제5장. 문장을 꾸미는 마법, 프린트 포매팅!>

선생님: (손에 예쁘게 장식된 편지지를 들고) 애들아, 안녕! 우리가 컴퓨터에게 `print()`로 말을 시킬 때, 그냥 내용만 전달하는 것보다 이렇게 예쁜 편지지에 보기 좋게 써서 보여주면 더 좋지 않을까? 오늘은 바로 이 `print()` 함수를 더 멋지게 꾸미는 방법을 배울 거란다!

<1. print 함수의 작은 비밀들>

선생님: 먼저, `print()` 함수의 작은 비밀 두 가지를 다시 한번 확인해볼까?

1. `print()`는 말을 한 번 할 때마다 자동으로 줄을 바꿔준다. (자동 '엔터' 기능!)
2. 만약 우리가 강제로 줄을 더 바꾸고 싶으면, 줄바꿈 문자 `\n`을 사용하면 되지!

```
a = 10
```

```
b = 2
```

```
print(a + b) # 12를 출력하고 자동으로 한 줄 띄움
```

```
print("\n") # \n으로 빈 줄을 하나 더 만듦
```

```
print(a - b) # 8을 출력하고 자동으로 한 줄 띄움
```

코딩새싹 🌱: 아! `\n`이 '줄바꿈'이라는 뜻이었죠! 정말 빈 줄이 하나 더 생겼어요!

<2. 문장 만들기 대작전: + vs 포매팅>

선생님: 우리가 "왕새우의 수학점수는 100점이다." 같은 문장을 만들 때, 예전에는 이렇게 + 기호로 글자랑 숫자를 연결했었지. 그런데 숫자는 `str()`로 변신시켜줘야 하고, 띄어쓰기도 신경 써야 해서 조금 불편했어.

```
print(name + "의 수학점수는 " + str(score) + "점" ) (조금 복잡한 옛날 방식)
```

궁금이 💡: 맞아요! 변수가 많아지면 +랑 `str()`이 너무 많아져서 헷갈릴 것 같아요! 더 좋은 방법이 있나요?

선생님: 물론! 그래서 등장한 것이 바로 문장의 형식을 미리 만들어두는 **포매팅(Formatting)** 이란다. 먼저, 조금 오래된 마법인 `.format()` 메소드를 만나보자!

<방법 1. 구멍 뚫고 채우기! .format() 메소드>

선생님: `.format()`은 문장에 {0}, {1}처럼 번호가 적힌 구멍을 뽕 뚫어놓고, 나중에 그 번호에 맞는 값을 채워 넣는 방식이야.

```
name = "최석민"
```

```
home = "화성"
```

```
score = 999
```

```
# {0} 자리에는 name, {1} 자리에는 score, {2} 자리에는 home을 채워줘!
```

```
print("{0}의 수학점수는 {1}점이다.\n{0}의 고향은 {2}이다.".format(name, score, home))
```

코딩새싹 🌱: 와! +로 연결하는 것보다 훨씬 깔끔해 보여요! {0} 자리에 name이 두 번이나 쓰였네요!

<방법 2. 최신 마법! 더 강력하고 편한 f-string!>

선생님: .format()도 좋지만, 요즘 파이썬 마법사들은 더 강력하고 쓰기 편한 **f-string**이라는 최신 마법을 주로 사용한다! 문자열 앞에 f를 붙여주기만 하면 돼!

궁금이 💡: f-string은 어떻게 다른데요?

선생님: f-string은 구멍에 번호를 매길 필요 없이, 중괄호 {} 안에 변수 이름을 바로 쏙! 넣어주면 된단다. 마치 마법 종이에 변수 상자를 그대로 넣으면 알아서 내용물을 보여주는 것과 같지!

```
name = "데카르트"
```

```
home = "프랑스"
```

```
score = 100
```

```
# f-string을 사용하면 변수를 { } 안에 바로 쓸 수 있어!
```

```
formatted_str = f"이름은 {name}이고, 고향은 {home}, 수학 점수는 {score}점입니다."
```

```
print(formatted_str)
```

코딩새싹 🌱: 우와! 이게 훨씬 편하고 보기 좋은데요? {name}이라고 쓰니까 바로 “데카르트”가 나왔어요!

<f-string의 슈퍼파워: 정렬과 숫자 꾸미기!>

선생님: f-string의 진짜 슈퍼파워는 지금부터야! 글자를 정렬하거나, 숫자를 원하는 형태로 꾸밀 수도 있거든. 중괄호 안에 {변수이름:꾸미기규칙} 형식으로 써주면 돼.

1. 자리 확보와 정렬하기!

마치 책꽂이에 책을 정리하듯, 글자를 원하는 칸 수에 맞춰 정렬할 수 있어!

- <: 왼쪽 정렬
- >: 오른쪽 정렬
- ^: 가운데 정렬

```
name = "왕새우아빠"
```

```
book = "일리아스"
```

```
price = 21000
```

이름은 10칸을 확보해서 가운데(^), 책은 10칸을 확보해서 왼쪽(<), 가격은 10칸을 확보해서 오른쪽(>) 정렬!

```
print(f'"{name:^10}"가 좋아하는 책은 "{book:<10}"이고, 가격은 "{price:>10}"입니다.')
```

궁금이 💡: 와! 따옴표를 넣어서 보니까 정말 정렬이 됐어요! name은 가운데, book은 왼쪽, price는 오른쪽에 붙어있네요!

2. 소수점 정리하기!

컴퓨터가 계산한 소수는 너무 길어서 지저분해 보일 때가 있지? f-string으로 깔끔하게 정리할 수 있어!

```
num = 2 / 3 # 0.66666666...
```

```
name = 124.222222
```

:.2f 는 '소수점(.) 아래 두(2) 자리까지만 보여주는 실수(f)로 만들어줘!' 라는 뜻이야.

```
print(f"2/3를 소수 둘째 자리까지 반올림하면? {num:.2f}")
```

```
print(f"내 이름은 사실 숫자... {name:.2f}입니다.")
```

코딩새싹 🌱: {num:.2f}라고 하니까 정말 0.67이라고 반올림까지 해서 깔끔하게 나왔어요! f는 float(실수)를 뜻하는 거군요!

궁금이 💡: 선생님, 아까 코드에 f"소수 둘째 자리까지 출력 : {num:2}" 이런 것도 있었는데, 이건 :.2f랑 다른 건가요?

선생님: 오, 아주 중요한 걸 발견했구나! 그냥 {num:2}라고 쓰면 '최소 2칸의 공간을 확보해줘'라는 뜻이라서 소수점을 정리하는 기능과는 다르단다. 소수점 자릿수를 정리할 때는 꼭 마침표(.)를 사용해서 :.2f 처럼 써줘야 한다는 걸 기억해두렴!

선생님: (편지지를 접으며) 자, 오늘 우리는 print 함수를 그냥 쓰는 것에서 넘어서, .format()과 f-string, 특히 f-string의 강력한 꾸미기 기능들을 배웠어. 이제 여러분은 단순히 결과를 출력하는 것을 넘어, 다른 사람이 보기 좋게, 깔끔하게 정리된 정보를 보여줄 수 있는 능력을 갖게 된 거란다! 정말 멋진 마법이지?

코딩새싹 🌱: 네, 선생님! 이제 f-string으로 뭐든지 예쁘게 출력할 수 있을 것 같아요! 정말 재미있었어요!