

<[파이썬 수학] 제2장. 글자들의 기차, 문자열(String) 놀이!>

선생님: (손에 긴 줄에 꿰인 구슬들을 보여주며) 애들아, 안녕! 'String'은 영어로 '줄'이라는 뜻이지? 파이썬에서 문자열(String)은 바로 이 구슬 목걸이처럼, 글자들이 한 줄로 쪽~ 순서를 갖고 엮여있는 것을 말한단다. 마치 글자들이 칸칸이 타고 있는 긴 기차 같기도 하지! 🚂

코딩새싹 🌱: 글자들의 기차요? 재미있는 비유예요! 그럼 "hello"는 h, e, l, l, o 라는 승객들이 탄 기차인 셈이네요!

선생님: 바로 그거야! 그리고 이 기차는 아주 중요한 규칙이 있어. 바로 출발역과 종착역을 꼭 표시해줘야 한다는 거지. 그게 바로 따옴표 "" 또는 ''란다.

문자형은 반드시 큰따옴표 " " 또는 작은따옴표 ' '로 둘러싸야 해.

```
print("hello world")
```

```
print('hello world')
```

궁금이 💡: 선생님, 그럼 만약에 글자 자체에 따옴표를 넣고 싶으면 어떻게 해요? 예를 들어 '나는 "배고파" 라고 말했다' 처럼요!

선생님: 오, 아주 예리한 질문이야! 그럴 땐 바깥쪽과 안쪽의 따옴표 종류를 다르게 써주면 된단다.

왜 에러가 안 날까요? -> 바깥쪽은 작은따옴표, 안쪽은 큰따옴표를 썼기 때문이지!

```
print('I say "i am here"')
```

코딩새싹 🌱: 아하! 서로 다른 따옴표를 쓰면 컴퓨터가 헛갈리지 않는군요!

<숫자 vs 문자열, 같지만 다른 친구!>

선생님: 지난 시간에 우리는 숫자 2와 글자 "2"가 다르다고 배웠지? 이 둘에게 똑같은 마법 지팡이(+, *)를 휘두르면 어떤 일이 일어날까?

숫자 2와 문자 '2'는 자료형이 서로 달라요.

```
print("숫자 2의 타입:", type(2)) # int (정수)
```

```
print("문자 '2'의 타입:", type('2')) # str (문자열)
```

1. 더하기(+) 마법!

```
print("숫자 2 + 3 =", 2 + 3) # 숫자 + 숫자는? 당연히 수학 계산!
```

```
print("문자 '2' + '3' =", '2' + '3') # 문자 + 문자는? 기차를 서로 연결!
```

2. 곱하기(*) 마법!

```
print("숫자 3 * 3 =", 3 * 3) # 숫자 * 숫자는? 구구단!
```

```
print("문자 'good' * 3 =", 'good' * 3) # 문자 * 숫자는? 기차를 복제해서 연결!
```

궁금이 💡: 와! 문자열한테 +를 쓰면 뒤에 착 갖다 붙고, *를 쓰면 여러 번 반복되는군요! 정말 기차를 연결하고 복제하는 것 같아요!

선생님: 그렇지? 하지만 절대로 섞어서는 안 돼! 숫자와 문자열, 즉 서로 다른 자료형끼리는 계산할 수 없단다.

```
a = "2" # a는 문자 기차
```

```
b = 3 # b는 숫자
```

```
# print(a + b) # 이 코드를 실행하면? TypeError가 발생! "타입이 다른데 어떻게 더해?"
```

코딩새싹 🌱: 아! "문자 2랑 숫자 3은 더할 수 없어!" 하고 파이썬이 화를 내는군요!

<인덱싱(Indexing): 원하는 칸의 승객 찾아내기!>

선생님: 글자 기차는 각 칸마다 번호가 매겨져 있어. 이 번호를 **인덱스(index)**라고 부르지. 이 번호를 알면 우리가 원하는 칸의 글자(승객)를 콧 집어서 불러낼 수 있단다! 중요한 건, **첫 번째 칸의 번호는 1이 아니라 0부터 시작**한다는 거야!

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

```
# H e l l o   W o r l d (글자)
```

```
# 0 1 2 3 4 5 6 7 8 9 10 (인덱스 번호)
```

```
print("0번 칸의 승객은?", my_string[0]) # H
```

```
print("6번 칸의 승객은?", my_string[6]) # W
```

궁금이 💡: 0부터 시작하는 건 좀 헷갈리지만, 알 것 같아요! 그런데 뒤에서부터 셀 수는 없나요? 마지막 글자를 찾고 싶을 때요!

선생님: 물론 있단다! 뒤에서부터 셀 때는 -1을 사용해! -1이 맨 뒤 칸, -2가 뒤에서 두 번째 칸이지.

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

```
# d l r o W   o l l e H (뒤에서부터)
```

```
# -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 (음수 인덱스)
```

```
print("뒤에서 첫 번째 승객은?", my_string[-1]) # d
print("뒤에서 다섯 번째 승객은?", my_string[-5]) # W

print("왕새우는 맛있다."[-2]) # '다'
```

코딩새싹 🌱: 와! -1이 마지막이라니! 편한데요? 그럼 숫자도 이렇게 인덱싱 할 수 있어요? 123의 첫 번째 숫자인 1을 꺼내고 싶어요!

선생님: 좋은 시도지만, 아쉽게도 그건 안 된단다. 숫자형 자료는 순서나 칸의 개념이 없는 하나의 덩어리라서 인덱싱을 할 수 없어! 오직 글자 기차(문자열)만 가능하지.

<슬라이싱(Slicing): 기차 칸 여러 개 잘라내기!>

선생님: 이번엔 승객 한 명만 부르는 게 아니라, '몇 번 칸부터 몇 번 칸까지' 여러 칸을 통째로 잘라내는 **슬라이싱(Slicing)**을 배워볼 거야! '얇게 썰다'는 뜻처럼, 문자열의 일부를 싹둑 잘라내는 거지. 콜론(:)을 사용한다!

기차[시작번호 : 끝번호 : 걸음수]

궁금이 💡: 끝번호까지 잘라내는 건가요?

선생님: 아주 중요한 포인트야! 슬라이싱은 **끝번호 바로 전 칸까지만** 잘라낸다는 걸 꼭 기억해야 해! 자, 직접 보자!

```
my_string = "python study"

# p y t h o n   s t u d y
# 0 1 2 3 4 5 6 7 8 9 10 11

# 0번 칸부터 6번 칸 "전"까지 잘라줘! -> 즉, 0~5번 칸
print("my_string[0:6] =", my_string[0:6]) # python

# 시작번호를 생략하면? 처음부터!
# 끝번호를 생략하면? 끝까지!
# "study"만 잘라내려면? 7번 칸부터 끝까지!
```

```
print("my_string[7:] =", my_string[7:]) # study
```

코딩새싹 🌱: 아! 끝번호는 포함이 안 되는군요! [7:] 이렇게 끝번호를 안 쓰니까 끝까지 다 나왔어요! 그럼 '걸음수'는 뭐예요?

선생님: '걸음수(step)'는 몇 칸씩 건너뛰면서 글자를 가져올지 정하는 거야.

```
my_string = "python study"
```

```
# 0번 칸부터 6번 칸 전까지, 2칸씩 건너뛰면서!
```

```
# p(0) -> t(2) -> o(4)
```

```
print("my_string[:6:2] =", my_string[:6:2]) # pto
```

```
# 여기서 아주 유용한 비밀 기술! 걸음수를 -1로 하면?
```

```
# 기차를 거꾸로! 한 칸씩 뒤로 가면서 모두 가져와!
```

```
print("my_string[::-1] =", my_string[::-1]) # yduts nohtyp
```

궁금이 💡: [::-1] 하니까 글자가 완전히 거꾸로 됐어요! 이거 정말 신기한데요!

<도전! 문자열 퀴즈!>

선생님: 자, 이제 배운 인덱싱과 슬라이싱 기술로 퀴즈를 풀어볼까?

[퀴즈] “너는 정말 똑똑하구나” 라는 문장에서 “는정 똑구”만 나타내시오!

힌트: 한 칸씩 건너뛰면 될 것 같은데... 시작은 어디서 해야 할까?

코딩새싹 🌱: 으음... '는'이 1번 칸이니까 1번부터 시작하고... 두 칸씩 건너뛰면 될 것 같아요!

```
words = "너는 정말 똑똑하구나"
```

```
# 너 는 정 말 똑 똑 하 구 나
```

```
# 0 1 2 3 4 5 6 7 8 9 10
```

```
# 1번 칸에서 시작해서, 끝까지, 2칸씩 건너뛰기!
```

```
print(words[1::2])
```

선생님: (엄지를 척 들며) 정답! 완벽해! '는'(1), '정'(3), ' '(5), '똑'(7), '구'(9) 이렇게 정확히 골라냈구나!

궁금이 💡: 문자열이 글자들의 기차 같다고 생각하니까 인덱싱이랑 슬라이싱이 훨씬 쉽게 이해됐어요!

선생님: 아주 좋아! 오늘 우리는 문자열이라는 글자 기차를 다루는 여러 가지 방법을 배웠어. 따옴표의 규칙부터, 숫자와는 다른 연산법, 그리고 원하는 글자를 콕 집어내거나(인덱싱) 잘라내는(슬라이싱) 기술까지! 이 기술들은 앞으로 프로그램을 만들 때 정말 많이 쓰일 테니 꼭 기억해두렴! 오늘 탐험도 정말 훌륭했어! 🌟