

<[파이썬 수학] 제10장. OX 퀴즈의 달인, 불리언(Boolean)과 비교!>

선생님: (O, X가 그려진 팻말을 들고) 애들아, 안녕! 우리가 푸는 퀴즈 중에는 “맞으면 O, 틀리면 X”로 답하는 OX 퀴즈가 있지? 컴퓨터 세상에도 딱 두 가지 대답만 존재하는 아주 특별한 자료형이 있단다. 그 이름은 바로 불리언(Boolean)!

코딩새싹 🌱: 불리언이요? 그럼 대답이 딱 두 개밖에 없는 거예요?

선생님: 맞아! 바로 True (참, 맞다!) 와 False (거짓, 아니다!), 이 두 가지뿐이지. 마치 신호등의 초록불(True)과 빨간불(False)처럼, 컴퓨터의 모든 판단은 결국 이 두 가지로 귀결된단다. 이 이름은 '조지 불'이라는 수학자 아저씨의 이름에서 따왔다고 해.

궁금이 💡: 그럼 파이썬은 어떤 걸 True로 보고, 어떤 걸 False로 봐요? 기준이 있나요?

선생님: 아주 좋은 질문이야! 파이썬은 '내용물이 있거나 0이 아니면 True', '텅 비어있거나 0이면 False'라고 생각하는 경향이 있어. 이걸 Truthy 와 Falsy 라고 부르지. bool()이라는 진실 탐지기 함수로 확인해볼 수 있단다!

- Falsy (거짓 같은 것들): False, None(아무것도 없음), 숫자 0, 비어있는 문자열 "", 비어있는 리스트 []
- Truthy (참 같은 것들): True, 0이 아닌 모든 숫자, 글자가 있는 문자열, 뭐든 들어있는 리스트

```
print("텅 빈 리스트 [] 는?", bool([]))      # False
print("숫자 1이 든 리스트 [1] 은?", bool([1])) # True
print("숫자 0은?", bool(0))                  # False
print("숫자 -100은?", bool(-100))            # True
print("글자 '안녕'은?", bool("안녕"))        # True
```

<1. 진실의 저울! 비교 연산자(Comparison Operators)>

선생님: 그럼 True나 False라는 대답은 언제 나오는 걸까? 바로 우리가 비교 연산자라는 '진실의 저울'을 사용할 때야! 이 저울은 두 값을 비교해서, 그 결과가 맞으면 True, 틀리면 False를 우리에게 알려준단다.

== : 양쪽 값이 '같다'면 True (주의! 변수에 값을 넣는 = 와는 달라!)

```
print("2 == (1 + 1) 인가요?", 2 == (1 + 1)) # True
```

!= : 양쪽 값이 '다르다'면 True

```
print("2 != (1 + 1 + 1) 인가요?", 2 != (1 + 1 + 1)) # True
```

```
# >, >=, <, <= : 크거나, 크거나 같거나, 작거나, 작거나 같은지 비교
print("2 >= 4 인가요?", 2 >= 4) # False
print("1 < 2 인가요?", 1 < 2) # True
```

코딩새싹 🌱: ==는 등호를 두 개나 쓰는군요! 변수에 값을 넣는 =랑 헷갈리지 말라고 그런가 봐요!

선생님: 정확해! 그리고 여기에 **not** 이라는 '청개구리 마법'을 쓸 수도 있어! not은 True는 False로, False는 True로 결과를 반대로 뒤집어 준단다.

```
# 2 == (1 + 1) 은 True 지만, 그 앞에 not을 붙이면?
print("not (2 == 1 + 1) 인가요?", not (2 == 1 + 1)) # not True -> False
```

<2. 보물찾기 게임! 포함 연산자 (in, not in)>

선생님: 이번엔 '이 안에 그거 있니?' 하고 물어보는 보물찾기 연산자, **in** 과 **not in**을 배워볼 거야. 문자열이나 리스트 같은 데이터 묶음 안에 우리가 찾는 값이 있는지 없는지 확인해서 True나 False로 알려주지.

```
my_str = "Be careful"

# "Be careful" 이라는 글자 기차 안에 'B'라는 승객이 타고 있니?
print("'B' in my_str ?", "B" in my_str) # True

# 'b'는 있니? (파이썬은 대소문자를 구분한단다!)
print("'b' in my_str ?", "b" in my_str) # False
```

```
nums = [12, 34, 56]

# [12, 34, 56] 이라는 마법 주머니 안에 56이라는 물건이 있니?
print("56 in nums ?", 56 in nums) # True

# 100은 없지?
print("100 not in nums ?", 100 not in nums) # 100이 없는 게 True (맞다)
```

코딩새싹 🌱: in을 쓰니까 정말 보물찾기 하는 것 같아요! “이 안에 이거 있어?” 하고 바로바로 물어볼 수 있네요!

궁금이 💡: 선생님! 그럼 숫자 안에도 숫자가 있는지 확인할 수 있어요? 예를 들어 숫자 123 안에 1이 있는지요?
1 in 123 이렇게요!

선생님: (미소 지으며) 이야, 정말 훌륭한 질문이야! 직접 해보면 어떻게 될까?

```
my_int = 123
```

```
# print(1 in my_int) # 이 코드를 실행하면? TypeError가 발생!
```

선생님: 파이썬이 “int 타입의 객체는 보물상자가 아니라서 안을 들여다볼 수 없어!” 하고 에러를 내는구나. in 연산자는 문자열이나 리스트처럼 여러 개의 항목을 담고 있는 **순서가 있는 데이터(sequence)**에만 사용할 수 **있단다**. 숫자 123은 하나의 덩어리지, 1, 2, 3이 들어있는 주머니가 아니기 때문이야. 만약 꼭 하고 싶다면? str() 변신 마법을 써서 '1' in str(123) 이라고 물어봐야겠지!

궁금이 💡: 아하! in 연산자는 컨베이어 벨트나 마법 주머니 같은 곳에만 쓸 수 있는 거군요! 이제 알겠어요!

선생님: (뺨말을 내려놓으며) 아주 좋아! 오늘 우리는 컴퓨터의 판단 기준이 되는 True와 False, 즉 불리언에 대해 배웠고, 비교 연산자와 포함 연산자를 통해 이 불리언 값을 얻어내는 방법을 익혔어. 이 ‘판단’ 능력은 다음 시간에 배울 if 조건문의 가장 중요한 열쇠가 될 거란다! 오늘 수업도 정말 훌륭했어! 🌟