<[파이썬 수학] 제11장. 생각하는 컴퓨터, if 조건문!>

선생님: (갈림길 표지판을 들고) 얘들아, 안녕! 우리가 길을 가다 보면 '왼쪽으로 갈까, 오른쪽으로 갈까?' 하고 선택을 해야 할 때가 있지? 컴퓨터에게도 이렇게 **특정 조건에 따라 다른 행동을 하도록** 가르칠 수 있단다. 바로 **if 조건문**이라는 마법을 사용해서 말이야!

코딩새싹 🌱: if는 '만약 ~라면' 이라는 뜻이죠? "만약 배가 고프면, 밥을 먹는다!" 처럼요?

선생님: 바로 그거야! 지난 시간에 우리는 True와 False라는 불리언 값을 얻는 법을 배웠지? if문은 바로 그 결과를 사용해. if 뒤에 오는 조건이 True이면, 정해진 행동을 하고, False이면 그냥 지나친단다.

<1. 가장 기본적인 선택: if 문>

선생님: if문의 기본 구조는 아주 간단해. if 조건문: 이라고 쓰고, 그 조건이 True일 때 실행할 명령들은 **반드시 들여쓰기**를 해줘야 해!

```
# money가 5000원 "이상"일 때만 "택시를 탄다"고 출력하고 싶다면?
# (예제 코드에서는 money > 5000 이었지만, money >= 5000 으로 해볼게!)
money = 6000

if money >= 5000: # 이 조건이 True인가? (6000 >= 5000 -> True!)
  print("돈이 충분하네! 택시를 탄다.") # True니까 이 문장이 실행된다!

money = 4000
if money >= 5000: # 이 조건이 True인가? (4000 >= 5000 -> False!)
  print("돈이 충분하네! 택시를 탄다.") # False니까 이 문장은 그냥 지나친다!
```

궁금이 💡: 아하! if문은 조건이 True일 때만 실행되는 '비밀 통로' 같은 거네요! False일 땐 그 통로의 존재조차 모르는 거고요!

선생님: 정확한 비유야! 이 if문은 for 반복문과 함께 쓰면 정말 강력한 힘을 발휘한단다.

[미션] 1에서 20까지 정수 중에 nums 리스트에 없는 숫자들만 모아서 새 리스트 만들기!

```
nums = [4, 7, 13, 17]
new_nums = []
```

```
for i in range(1, 21): # 1부터 20까지 숫자를 하나씩 꺼내서
  if i not in nums: # 만약 그 숫자가 nums 리스트에 "없다면" (이 조건이 True라면)
    new_nums.append(i) # new_nums 리스트에 추가해!

print(new_nums)
```

<2. 둘 중 하나를 선택해! if-else 문>

선생님: '만약 조건이 True이면 A를 해!' 라고 시켰는데, 만약 조건이 False일 경우엔 'B를 해!' 라고 알려주고 싶을 때도 있겠지? 그럴 때 else를 함께 쓴단다. else는 '그렇지 않으면' 이라는 뜻이야.

```
# 내가 가진 돈이 3000원 이상이면 택시를 타고, "그렇지 않으면" 지하철을 탄다.
money = 2500

if money >= 3000: # 조건이 True인가? (2500 >= 3000 -> False!)
  print("택시 탄다.")
else: # if 조건이 False였으므로, 이쪽으로 온다!
  print("아쉽지만 지하철을 탄다.")
```

코딩새싹 🌱: else에는 조건을 안 써줘도 되네요?

선생님: 맞아! else는 if 조건이 False인 **모든 나머지 경우**를 뜻하기 때문에 따로 조건을 쓸 필요가 없단다. 자, 그럼 이걸로 짝수와 홀수를 나눠 담는 프로그램을 만들어볼까?

```
even_list = []
odd_list = []

for i in range(1, 31): # 1부터 30까지
  if i % 2 == 0: # 2로 나눈 나머지가 0이면 (짝수이면)
    even_list.append(i)
  else: # 그렇지 않으면 (홀수이면)
    odd_list.append(i)

print("짝수 모음:", even_list)
```

```
print("홀수 모음:", odd_list)
```

<3. 여러 갈래의 선택지! if-elif-else 문>

선생님: 그런데 세상에는 'A 아니면 B' 처럼 선택지가 두 개만 있는 건 아니지. '택시', '버스', '도보' 처럼 여러 선택지가 있을 땐 어떻게 할까? 그럴 때 바로 **elif** 가 등장한단다! elif는 "else if", 즉 '그게 아니라, 만약 ~라면'의 줄임말이야.

```
# 돈에 따라 여러 가지 교통수단을 선택해볼까?
money = 4500

if money >= 5000: # 1. 5000원 이상인가? (4500 >= 5000 -> False!) -> 다음으로!
  print("택시 타고 간다.")
elif money >= 3000: # 2. 그렇다면 3000원 이상인가? (4500 >= 3000 -> True!) -> 여기 실행하고 끝!
  print("버스 타고 간다.")
else: # 위 조건들이 모두 False일 때만 실행
  print("걸어간다.")
```

궁금이 💡: 아하! if문을 먼저 체크하고, False면 그 다음 elif를 체크하고… 순서대로네요! 만약 중간에 elif가 True가 되면, 그 뒤에 다른 elif나 else는 아예 쳐다보지도 않는 거군요!

선생님: 완벽하게 이해했어! 순서가 아주 중요하고, 딱 하나만 선택되면 나머지는 무시된단다.

<최종 도전! 자동 성적 계산기 만들기!>

선생님: 자, 이제 우리가 배운 for 반복문과 if-elif-else를 모두 합쳐서, 학생들의 평균 점수와 성적(A, B, C, D, F)을 자동으로 계산해서 추가해주는 엄청난 프로그램을 만들어보자!

```
my_class = [
    ["이름", "국어점수", "수학점수", "영어점수"], # 제목 행 (평균, 성적은 비어있음)
    ["홍길동", 50, 80, 70],
    ["데카르트", 100, 100, 100],
    ["맹구", 10, 20, 10]
]
```

```
# 제목 행을 제외하고, 각 학생의 정보(리스트)를 반복한다.
for student_data in my_class[1:]:
  # 1. 평균 계산해서 추가하기
  scores = student_data[1:]
  avg = sum(scores) / len(scores)
  student_data.append(round(avg, 2)) # 계산된 평균을 학생 리스트에 추가!

  # 2. 성적 매기기 (if-elif-else 구조 사용!)
  if avg >= 90:
    grade = "A"
  elif avg >= 80:
    grade = "B"
  elif avg >= 70:
    grade = "C"
  elif avg >= 60:
    grade = "D"
  else:
    grade = "F"

  student_data.append(grade) # 부여된 성적을 학생 리스트에 추가!

  # 3. 결과 출력하기
  print(f"{student_data[0]}의 성적: {student_data[-1]}")

print("\n--- 최종 결과 리스트 ---")
print(my_class)
```

코딩새싹 🌱: 와! for문 안에서 if문을 쓰니까 이렇게 복잡한 일도 자동으로 처리할 수 있네요! 홍길동, 데카르트, 맹구의 평균이랑 성적이 리스트에 착착 추가됐어요!

궁금이 💡: 정말 대단해요! 불리언으로 조건을 판단하고, if문으로 행동을 결정하고, for문으로 그걸

반복하니까… 정말 컴퓨터가 스스로 생각하고 일하는 것처럼 보여요!

선생님: (활짝 웃으며) 바로 그거야! 오늘 배운 `if` 조건문은 프로그램을 '지능적'으로 만들어주는 핵심적인 마법이란다. 조건을 판단해서 다른 행동을 하게 만드는 능력! 이것만 있으면 여러분은 훨씬 더 똑똑하고 유용한 프로그램을 만들 수 있을 거야. 오늘 정말 어려운 내용을 배우느라 수고 많았어! 🌟