<제6장. 도전! 성적표 만들기 대작전!>

선생님: (반짝이는 눈으로) 코딩새싹 🌱, 궁금이 💡! 지난 시간에 딕셔너리라는 보물 상자에 대해 배웠지? 오늘은 그 보물 상자를 활용해서 아주 중요한 임무를 수행할 거야! 바로, 학생들의 성적표를 만드는 대작전이지! 📜✨

코딩새싹 🌱: 와! 성적표요? 제가 잘 할 수 있을까요? ドキドキ (두근두근)

궁금이 💡: 재미있겠어요! 그럼 학생들 이름이랑 점수가 필요하겠네요? 그리고 등급은 어떻게 나눠요? A, B, C… 이런 거요?

선생님: 역시 궁금이는 핵심을 바로 아는구나! 맞아. 우리에게는 두 가지 중요한 정보 묶음, 즉 두 개의 딕셔너리가 필요하단다. 첫 번째는 학생들의 이름과 점수가 담긴 scores 딕셔너리고, 두 번째는 어떤 점수가 어떤 등급인지 알려주는 grades 딕셔너리지!

<1단계: 준비물 챙기기 - 딕셔너리 친구들 소환!>

선생님: 자, 먼저 우리 학생들의 점수부터 확인해볼까?

```
# 학생 이름과 점수를 담은 딕셔너리 생성
scores = {
    "John": 85,
    "Emily": 92,
    "Michael": 78,
    "Jessica": 90,
    "David": 88
}
print("학생들 점수:", scores)
```

코딩새싹 🌱: 아하! "John"이 키고 85가 값이네요! 각 학생 이름에 점수가 짝꿍처럼 연결되어 있어요!

선생님: 그렇지! 이제 등급 기준을 정해볼까? 이 grades 딕셔너리는 '점수 기준'과 '등급'을 짝지어 놓을 거야. 중요한 건, 가장 높은 등급의 기준부터 차례대로 적어줘야 해. 왜냐하면, 우리 코드가 위에서부터 "이 점수 넘었니?" 하고 물어보면서 내려올 거거든.

```
# 점수에 따른 등급을 매기는 딕셔너리 생성 (높은 점수 기준부터!)
grades = {
```

```
    90: "A",  # 90점 이상이면 A
    80: "B",  # 80점 이상이면 B
    70: "C",  # 70점 이상이면 C
    60: "D"   # 60점 이상이면 D
}
print("등급 기준표:", grades)
```

궁금이 💡: 선생님, 그럼 60점 미만은 어떻게 해요? 저기엔 없는데요?

선생님: 오, 아주 중요한 질문이야, 궁금아! 그건 우리가 코드를 만들면서 해결해 줄 거란다. 잠시만 기다려줘!

😉 자, 마지막으로 우리가 만든 학생별 최종 등급을 담아둘 빈 딕셔너리, student_grades도 준비하자!

```
# 학생 등급을 저장할 딕셔너리 초기화
student_grades = {}
print("아직 비어있는 최종 성적표:", student_grades)
```

<2단계: 마법의 주문으로 등급 매기기! - 코드 살펴보기>

선생님: 이제 모든 준비가 끝났으니, 마법의 주문(코드)으로 학생들의 등급을 매겨볼 시간이야! 코드가 조금 길어 보일 수 있지만, 한 줄 한 줄 살펴보면 어렵지 않아.

```
# scores 딕셔너리 순회하며 등급 할당
for student, score in scores.items():  # 1번: 학생과 점수를 한 명씩 꺼내요!
    print(f"\n지금 {student} 학생의 점수 {score}점으로 등급을 매겨볼게요!")
    for grade_score, grade in grades.items(): # 2번: 등급 기준표를 위에서부터 하나씩 봐요!
        print(f"  혹시 {score}점이 {grade_score}점 이상인가요?")
        if score >= grade_score:          # 3번: 학생 점수가 등급 기준 점수보다 높거나 같으면,
            student_grades[student] = grade # 4번: 이 학생의 등급은 바로 이것!
            print(f"  찾았다! {student} 학생의 등급은 {grade}!")
            break                         # 5번: 등급 찾았으니 이 학생은 여기서 끝! 다음 학생으로!
    else:                                 # 6번: 등급 기준표를 다 봤는데도 맞는 게 없으면 (즉, break를 못
        student_grades[student] = "F"     # 7번: 이 학생은 F 등급!
        print(f"  이런... {student} 학생은 D등급 기준인 60점도 못 넘었네요. F등급입니다.")
```

코딩새싹 🌱: 우와! for가 두 번이나 쓰였어요! 첫 번째 for student, score in scores.items():는 학생들

점수 딕셔너리에서 한 명씩 이름이랑 점수를 꺼내는 거죠?

선생님: 정확해! `items()` 기억나지? 키와 값을 쌍으로 꺼내주는 마법! 그럼 `student`에는 학생 이름이, `score`에는 그 학생의 점수가 쏙 들어간단다.

궁금이 💡: 그럼 두 번째 `for grade_score, grade in grades.items():`는 아까 그 등급 기준표에서 점수랑 등급을 꺼내는 거네요? 그래서 `if score >= grade_score:` 이 부분에서 학생 점수랑 기준 점수랑 비교하는 거고요!

선생님: (엄지를 척 들며) 바로 그거야! 예를 들어 John의 점수가 85점이면, 등급 기준표를 보면서 '85점이 90점 이상인가? (아니네) → 85점이 80점 이상인가? (맞네!)' 이렇게 찾는 거지.

코딩새싹 🌱: 선생님! `break`는 뭐예요? 갑자기 "멈춰!" 하는 것 같아요!

선생님: 맞아! `break`는 "빙고! 찾았다!" 싶으면 더 이상 안 찾아도 되니까 "여기서 멈추고 이 반복문은 끝!" 하고 빠져나오는 마법 주문이야. John이 85점으로 'B' 등급을 받았으면, 그 밑에 'C' 등급이나 'D' 등급 기준은 볼 필요가 없잖아? 그래서 `break`로 다음 학생으로 넘어가는 거지.

궁금이 💡: 아하! 그럼 아까 제가 질문했던 60점 미만인 학생은 어떻게 되는 거예요? `grades` 딕셔너리에는 F 등급 기준이 없었잖아요! 그때 `else`가 등장하는 건가요?

선생님: 역시 궁금이! 예리하구나! 파이썬의 `for`문은 아주 특별한 `else` 짝꿍을 가질 수 있단다. `for`문 안에서 `break`를 만나지 않고 모든 반복을 다 마쳤을 때, 즉 "등급 기준표를 A부터 D까지 샅샅이 뒤졌는데 맞는 게 하나도 없었네?" 할 때 이 `else` 부분이 실행되는 거야!

코딩새싹 🌱: 와! `for`랑 `else`가 짝꿍이라니 신기해요! 그럼 점수가 너무 낮아서 A, B, C, D 어디에도 해당 안 되면, `break`를 못 만나니까 자동으로 `else`로 가서 F 등급을 받는 거네요!

선생님: (박수를 짝짝) 완벽하게 이해했어! 바로 그거란다. 자, 그럼 이 모든 마법이 끝나면 `student_grades`에는 어떤 결과가 담겨 있을까?

<3단계: 결과 발표! - 완성된 성적표 확인!>

선생님: 두구두구두구… 이제 최종 결과를 확인해볼 시간이야!

```
# 결과 출력
print("\n--- 최종 성적표 ---")
print(student_grades)
```

궁금이 💡: `{ "John": "B", "Emily": "A", "Michael": "C", "Jessica": "A", "David": "B" }` 이렇게 나올 것 같아요! Michael 학생은 78점이니까 C 등급일 거고요!

코딩새싹 🌱: 와! 정말 우리가 직접 학생들 성적표를 만든 거네요! 딕셔너리랑 `for`문, `if`문, 그리고 `break`랑 `else`까지! 배운 걸 다 써본 것 같아요!

선생님: (활짝 웃으며) 아주 훌륭해, 우리 학생들! 오늘 우리는 딕셔너리를 이용해서 조금 더 복잡하지만 아주 유용한 프로그램을 만들어봤어. 각 학생의 점수를 가져오고, 정해진 기준에 따라 등급을 매기고, 그 결과를 새로운 딕셔너리에 착착 정리했지. 이처럼 코딩은 우리가 배운 조각들을 모아서 새로운 문제를 해결하는 멋진 과정이란다!

궁금이 💡: 선생님, 만약에 등급 기준이 바뀌거나 새로운 학생이 추가돼도 이 코드를 조금만 고치면 되겠네요?

선생님: 정확해! 그게 바로 코딩의 힘이지! 상황이 바뀌면 코드도 유연하게 바꿔서 대처할 수 있단다. 오늘 정말 멋진 탐험이었어! 다음엔 또 어떤 재미있는 문제에 도전해볼까?