

<제7장. 상자 속의 상자! 중첩 딕셔너리의 비밀!>

선생님: (눈을 반짝이며) 코딩새싹 🌱, 궁금이 💡! 우리가 지금까지 사용한 딕셔너리 보물 상자는 한 칸짜리였지? 그런데 만약 보물 상자 안에 또 다른 작은 보물 상자들이 여러 개 들어있다면 어떨까? 그리고 그 작은 상자 안에는 번쩍이는 보석 목걸이(리스트)나 친구들의 비밀 쪽지(딕셔너리)가 또 들어있다면?



코딩새싹 🌱: 와! 상상만 해도 신기해요! 그럼 엄청나게 많은 정보를 한 곳에 다 담을 수 있겠는데요? 마치 마법사 할아버지의 비밀 방 같아요! 🧙‍♂️ 🔮 📖

선생님: 바로 그거야! 이것 바로 **중첩 딕셔너리(Nested Dictionary)**라고 부른다. 딕셔너리 안에 또 다른 딕셔너리나 리스트 같은 값들이 들어가는 구조지. 마치 러시아 인형 마트료슈카처럼 말이야! 까도 까도 계속 나오는 신기한 구조!

궁금이 💡: 그럼 그 안에 있는 정보는 어떻게 꺼내요? 열쇠가 엄청 많이 필요할 것 같아요! 🔑 🔑 🔑 🔑 🔑 🔑 🔑

선생님: 아주 좋은 질문이야, 궁금아! 열쇠를 여러 개 사용한다고 생각하면 돼. 차근차근 하나씩 열고 들어가면 된단다. 자, 여기 어마어마한 사용자 정보가 담긴 users 딕셔너리를 한번 보자! (선생님은 먼저 간단한 예시로 설명하기 시작한다)

사용자 정보를 담은 (작은 예시) 딕셔너리

```
users = {  
    "john": { # 첫 번째 사용자 "john"의 정보 상자  
        "name": "John Doe",  
        "age": 30,  
        "email": "john@example.com",  
        "interests": ["reading", "traveling"], # 관심사는 여러 개일 수 있으니 리스트!  
        "friends": [ # 친구 정보도 여러 명일 수 있으니 리스트!  
            { # 각 친구 정보는 또 작은 정보 상자(딕셔너리)!  
                "name": "Jane Smith",  
                "age": 28,  
                "email": "jane@example.com"  
            },  
            {
```

```

        "name": "Mike Johnson",

        "age": 32,

        "email": "mike@example.com"

    }

]

},

"emily": { # 두 번째 사용자 "emily"의 정보 상자

    "name": "Emily Brown",

    "age": 25,

    "email": "emily@example.com",

    "interests": ["painting", "photography"],

    "friends": [

        {

            "name": "David Lee",

            "age": 27,

            "email": "david@example.com"

        }

    ]

}

}

```

코딩새싹 🌱: 와아... 정말 복잡해 보여요! users 라는 큰 상자 안에 "john"이랑 "emily"라는 이름표가 붙은 중간 상자가 있고, 그 중간 상자 안에 또 "name", "age" 같은 이름표들이 붙어있어요! "interests"는 리스트고, "friends"는... 리스트 안에 또 딕셔너리가 들어있어요! 🤖

선생님: 하하! 맞아, 처음 보면 조금 어지러울 수 있지만, 구조를 알면 아주 체계적입니다. 마치 큰 도서관(users)에 가서, 'John 섹션'("john")으로 간 다음, 거기서 '이름 카드'("name")를 찾는 것과 같지!

<1단계: 보물 상자 속 보물 찾아내기! - 중첩 데이터 접근>

선생님: 자, 그럼 John의 이름을 찾아볼까? users라는 큰 상자에서 "john" 열쇠로 중간 상자를 열고, 그 안에서 "name" 열쇠로 이름을 꺼내는 거야!

사용자 정보 접근 및 출력

```
print(users["john"]["name"]) # 출력: John Doe
```

궁금이 💡: 아하! 대괄호 []를 연달아 쓰는군요! users["john"] 이게 John의 모든 정보가 담긴 딕셔너리고, 거기서 다시 ["name"]으로 이름을 꺼내는 거네요! 그럼 Emily의 나이는... users["emily"]["age"] 이렇게 하면 될까요?

선생님: (미소 지으며) 정확해! 바로 그거야! 어디 한번 확인해볼까?

```
print(users["emily"]["age"]) # 출력: 25
```

코딩새싹 🌱: 진짜 25가 나왔어요! 그럼 John의 관심사들("interests")은 어떻게 꺼내요? 저건 리스트인데...

선생님: 좋은 질문! John의 관심사들은 리스트 형태로 저장되어 있으니, 그대로 리스트가 나올 거야.

```
print(users["john"]["interests"]) # 출력: ["reading", "traveling"]
```

선생님: 자, 이제 최고 난이도 퀴즈! Emily의 친구 목록에서 첫 번째 친구의 이메일 주소를 꺼내려면 어떻게 해야 할까? 힌트: 친구 목록("friends")은 리스트고, 리스트 안의 각 친구 정보는 딕셔너리란다!

궁금이 💡: 으음... 일단 users["emily"]["friends"]까지 가면 Emily의 친구들 목록(리스트)이 나오겠죠? 그 리스트에서 첫 번째 친구니까... 리스트의 첫 번째는 [0]으로 꺼내지 않나요? 그럼 users["emily"]["friends"][0]! 여기까지 하면 첫 번째 친구의 정보가 담긴 딕셔너리가 나오겠네요!

코딩새싹 🌱: 아! 그럼 그 딕셔너리에서 이메일 주소를 꺼내려면 다시 ["email"]을 붙이면 되겠다! users["emily"]["friends"][0]["email"] 맞아요, 선생님?

선생님: (박수를 치며) 브라보! 우리 학생들이 이렇게 똑똑하다니! 정말 대단해! 정확히 맞혔어!

```
print(users["emily"]["friends"][0]["email"]) # 출력: david@example.com
```

선생님: 이렇게 한 단계씩, 마치 탐험가가 지도를 보며 길을 찾아가듯 필요한 정보에 접근하면 된단다.

<2단계: 보물 상자 꾸미기! - 중첩 데이터 수정>

선생님: 정보를 꺼내는 것뿐만 아니라, 수정하거나 추가할 수도 있어. 예를 들어, John의 관심사에 '음악 감상(music)'을 추가해볼까? John의 관심사는 리스트니까, 리스트에 항목을 추가하는 .append()를 사용하면 돼!

새로운 관심사 추가

```
users["john"]["interests"].append("music")
```

```
print(users["john"]["interests"]) # 출력: ["reading", "traveling", "music"]
```

코딩새싹 🌱: 와! ["reading", "traveling", "music"]으로 바뀌었어요! .append()는 리스트 끝에 추가하는 마법 주문이었죠!

궁금이 💡: 그럼 Emily의 첫 번째 친구 David Lee의 나이를 한 살 올려서 28살로 바꿔줄 수도 있나요?

선생님: 물론이지! 아까 David Lee의 이메일을 찾았던 것처럼, 나이 정보까지 찾아가서 새로운 값을 넣어주면 돼.

```
# 친구 정보 업데이트
```

```
users["emily"]["friends"][0]["age"] = 28 # David Lee의 나이를 28로 변경
```

```
print(users["emily"]["friends"][0]) # 출력: {"name": "David Lee", "age": 28, "email": "david@example.com"}
```

코딩새싹 🌱: 와! 정말 David Lee의 나이가 28로 바뀌었어요! 필요한 곳까지 콕 집어서 찾아가서 바꿀 수 있다니, 정말 강력한데요!

<3단계: 거대한 정보의 바다! - 확장된 사용자 데이터>

선생님: 맞아. 이런 중첩 구조는 아주 많은 양의 복잡한 데이터를 체계적으로 관리할 때 정말 유용하단다. (선생님은 사용자가 제공한 더 큰 users 딕셔너리 예시를 보여주며) 봐봐, 이렇게 사용자 정보가 수십, 수백 명이 되어도 동일한 구조를 사용하면 각자의 이름, 나이, 친구 정보까지 깔끔하게 정리하고 필요할 때마다 쓱쓱 꺼내 쓸 수 있겠지?

```
# (여기서 사용자가 제공한 더 큰 users 딕셔너리가 있다고 상상해 주세요!)
```

```
# users = {  
  
#     "john": { ... },  
  
#     "emily": { ... },  
  
#     "michael": { ... },  
  
#     # ... 수많은 사용자 정보 ...  
  
#     "grace": { ... }  
  
# }
```

궁금이 💡: 와! 정말 많은 정보가 들어있어도, 규칙만 알면 원하는 정보를 딱 찾을 수 있겠어요! 마치 거대한 백과사전 같아요!

선생님: (흐뭇하게 웃으며) 그렇지! 오늘 우리는 딕셔너리 안에 또 딕셔너리가 들어가는 '중첩 딕셔너리'라는

아주 멋진 구조를 배웠어. 처음엔 복잡해 보였지만, 열쇠를 차근차근 사용해서 원하는 정보에 접근하고 수정하는 방법을 익혔지? 이 능력은 앞으로 여러분이 더 크고 복잡한 프로그램을 만드는 데 아주 중요한 밑거름이 될 거란다!

코딩새싹 🌱: 네, 선생님! 이제 어떤 복잡한 정보라도 딕셔너리로 잘 정리할 수 있을 것 같아요! 상자 속의 상자, 정말 재미있었어요!

궁금이 💡: 저도요! 이제 친구들 정보를 이렇게 정리해서 저만의 비밀 연락처를 만들어보고 싶어요! 물론, 친구들 허락은 받아야겠지만요! 😊

선생님: 하하, 아주 좋은 생각이야! 코딩은 이렇게 우리가 배운 것을 실생활에 적용해보면서 더욱 발전하는 거란다. 오늘도 정말 수고 많았어, 우리 꼬마 탐험가들!