

파이썬의 다양한 데이터 친구들과 반복 마법사 for! 📊🔄✨

등장인물:

- **슬기 선생님:** 친절하고 유머러스한 코딩 선생님
- **지혜:** 똑똑하고 질문이 많은 학생
- **코딩이:** 코딩을 처음 배우지만 호기심 많은 학생

슬기 선생님: 여러분, 안녕하세요! 우리가 지금까지 숫자도 다뤄보고, 글자도 다뤄보고, 우리 거북이 친구들도 만들어봤죠? 오늘은 파이썬이 또 어떤 다양한 친구들(자료 형태)을 가지고 있는지, 그리고 아주 강력한 반복 마법사 for에 대해서도 알아볼 거예요!

코딩이: 새로운 친구들이요? 거북이 말고 또 누가 있어요? 반복 마법사는 뭐예요? 똑같은 말을 계속하는 건가요? “왕새우 만세! 왕새우 만세!” 이렇게요?

슬기 선생님: (웃음) 코딩이 학생, 예리한데요? 비슷한 면도 있어요! 먼저 파이썬의 여러 가지 데이터 친구들부터 만나볼까요?

```
a = 10
b = "왕새우"
nayon = ["김나연", 13, "서울", 100, 95, 80]
d = (255, 200, 0)
```

```
print(type(a))
print(type(b))
print(type(nayon))
print(type(d))
```

지혜: 와, a는 숫자 10이고, b는 글자 “왕새우”인 건 알겠어요. type() 함수로 확인해보니 a는 <class 'int'> (정수), b는 <class 'str'> (문자열)이라고 나오네요! 그런데 nayon이랑 d는 처음 보는 모양이에요! nayon은 대괄호 []로 감싸져 있고, d는 소괄호 ()로 감싸져 있어요.

슬기 선생님: 맞아요! nayon처럼 대괄호 []로 여러 가지 정보를 순서대로 묶어 놓은 것을 ****리스트(list)****라고 불러요. 리스트 안에는 지금 “김나연”이라는 문자열, 13이라는 숫자, “서울”이라는 문자열, 그리고 점수들이 들어있죠? 이렇게 여러 종류의 데이터를 한꺼번에 담을 수 있는 아주 유용한 보관함 같은 거예요. type(nayon)을 하니 <class 'list'>라고 나오죠?

코딩이: 오! nayon 리스트 안에는 김나연 학생의 이름, 나이, 사는 곳, 시험 점수까지 다 들어있네요! 그럼 d = (255, 200, 0) 이건 뭐예요? 이것도 숫자 세 개가 묶여 있는데요? type(d)는 <class 'tuple'>이라고 나와요. 튜플?!

슬기 선생님: 네, d처럼 소괄호 ()로 묶인 것은 ****튜플(tuple)****이라고 해요. 리스트랑 비슷하게 여러 값을 순서대로 담을 수 있지만, 튜플은 한번 만들면 안의 내용을 바꿀 수 없다는 특징이 있습니다. 우리가 RGB 색상 값을 쓸 때 (255, 200, 0) 이렇게 썼던 것 기억나요? 그게 바로 튜플이었어요!

지혜: 아하! 리스트는 수정할 수 있는 보관함, 튜플은 한번 정하면 바꿀 수 없는 꾸러미 같은 거네요!

슬기 선생님: 정확해요! 자, 이제 오늘의 하이라이트! 반복 마법사 for를 만나볼 시간이에요. for는 똑같은 일을 여러 번 반복해야 할 때 아주 편리하게 사용할 수 있는 마법 주문입니다. 예를 들어 1부터 9까지의 숫자를 각각 제곱해서 출력하고 싶다고 해볼게요.

```
for i in range(1, 10):  
    print(i * i)
```

코딩이: 와! for i in range(1, 10): 하니까 진짜 1x1=1, 2x2=4, ..., 9x9=81까지 쭉 나왔어요! range(1, 10)은 1부터 9까지 숫자를 만들어주는 건가 봐요!

슬기 선생님: 맞아요! range(1, 10)은 1부터 시작해서 10 바로 앞 숫자, 즉 9까지의 연속된 숫자를 만들어주는 특별한 기능을 해요. list(range(1, 10)) 이라고 써보면 [1, 2, 3, 4, 5, 6, 7, 8, 9] 이렇게 숫자들이 들어있는 리스트가 만들어지는 걸 볼 수 있답니다.

지혜: 선생님, for i in range(1, 10): 다음에 콜론(:)이 꼭 붙어야 하나요? 그리고 다음 줄에 print(i * i)는 왜 앞으로 살짝 들어가 있어요?

슬기 선생님: 아주 중요한 부분을 발견했네요, 지혜 학생! for 반복문을 쓸 때는 약속이 있어요. 첫째, for 변수 in 반복할_범위: 처럼 쓰고, 맨 마지막에는 반드시 **콜론(:)**을 찍어줘야 해요. “이제부터 이 범위만큼 반복할 거야!” 라는 신호 같은 거죠. 둘째, for 문 다음 줄부터 반복하고 싶은 명령어들은 반드시 똑같은 간격만큼 안으로 들여쓰기를 해야 해요. 보통 스페이스바 네 번 또는 탭(Tab) 키 한 번으로 들여쓰답니다. 이렇게 들여쓰기 된 부분을 하나의 **코드 블록(code block)**이라고 불러요. “여기까지가 한 묶음으로 반복될 명령어들이야!” 라는 뜻이죠. 만약 여러 줄을 한꺼번에 들여쓰고 싶으면, 마우스로 그 줄들을 쭉 선택(드래그)한 다음에 탭 키를 누르면 한 번에 쑥 들어간답니다.

코딩이: 아하! 콜론이랑 들여쓰기가 반복 마법의 비밀이었군요! 그럼 my_list = ["박온유", "박주한", "최지훈", "왕새우"] 이 리스트에 있는 이름을 하나씩 꺼내서 “잘생겼다”랑 같이 출력하려면요?

슬기 선생님: 그것도 for 마법으로 간단하게 할 수 있어요!

```
my_list = ["박온유", "박주한", "최지훈", "왕새우"]
```

```
for name in my_list:  
    print(name)  
    print("잘생겼다.")
```

지혜: for name in my_list: 이렇게 하니까 name이라는 변수에 “박온유”가 한 번, “박주한”이 한 번, 이렇게 리스트 안의 이름들이 차례대로 쭉쭉 들어가면서 print(name)과 print("잘생겼다.")가 반복되네요! 정말 편리한데요!

코딩이: 선생님! 만약에요... 제가 들여쓰기를 깜빡하고 이렇게 쓰면 어떻게 돼요?

코딩이의 실수 예시

```
my_list = ["박온유", "박주한", "최지훈", "왕새우"] # 이 줄은 문제 없음
```

```
for name in my_list:
print(name) # 어? 들여쓰기를 안 했네!
    print("잘생겼다.") # 이 줄은 또 들여쓰기가 이상하고...
```

슬기 선생님: (코딩이의 코드를 보며) 아하! 코딩이 학생이 아주 중요한 실수를 해줬어요! 이렇게 for name in my_list: 다음에 와야 할 명령어들이 제대로 들여쓰기 되어 있지 않으면, 파이썬은 “어? for 문 다음에 반복할 명령어가 어디 갔지? 아니면 얼마나 들여써야 하는지 모르겠어!” 하면서 당황하게 된답니다. 그래서 이런 오류 메시지를 보여줄 거예요.

```
IndentationError: expected an indented block after 'for' statement on line 3
```

지혜: IndentationError! 들여쓰기 오류라는 뜻이네요! “3번째 줄에 있는 for 문 다음에는 들여쓰기 된 코드 블록이 와야 해!” 라고 알려주는 거군요!

슬기 선생님: 맞아요! 파이썬에서 들여쓰기는 단순한 꾸미기가 아니라, 코드의 구조를 알려주는 아주 중요한 약속이에요. 그래서 for 문이나 나중에 배울 if 문 다음에는 반드시 들여쓰기로 코드 블록을 만들어줘야 합니다. 정확히는 이렇게 써야겠죠.

```
my_list = ["박온유", "박주한", "최지훈", "왕새우"]
```

```
for name in my_list:
    print(name)          # 스페이스바 4번 또는 탭 1번으로 들여쓰기
    print("잘생겼다.") # 위와 똑같이 들여쓰기
```

코딩이: 아! 들여쓰기가 틀리면 컴퓨터가 못 알아듣는구나! 이제부터 콜론이랑 들여쓰기 꼭꼭 챙길게요! 그럼 “왕새우 만세”를 10번 외치게 하려면요? range(10) 쓰면 되나요?

슬기 선생님: 네, 맞아요! range(10)은 0부터 9까지 총 10개의 숫자를 만들어주니까, 10번 반복할 때 아주 유용하게 쓸 수 있어요.

```
# "이름 만세"를 10번 출력하세요.
```

```
for i in range(10):
    print("왕새우 만세")
```

```
print("하하하") # 이 줄은 for 반복문 밖에 있으니 한 번만 실행돼요!
```

지혜: for i in range(10): 하니까 print("왕새우 만세")가 정말 10번 반복됐어요! 그리고 마지막 print("하하하")는 들여쓰기가 안 되어 있으니 반복문이 끝나고 딱 한 번만 실행되는 거군요!

슬기 선생님: 정확합니다! 오늘 우리는 리스트, 튜플 같은 새로운 자료형 친구들도 만나봤고, 강력한 반복 마법사 for와 그의 단짝 range(), 그리고 아주 중요한 콜론(:)과 들여쓰기 규칙까지 배웠어요. 정말 많은 것을 배웠죠? 이제 반복적인 작업은 for 마법사에게 맡겨보세요!

코딩이 & 지혜: 네! 선생님! 이제 반복 작업도 문제없어요! for 최고!