

파이썬 거북이와 함께 그림 그리기! 🐢🎨

등장인물:

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

슬기 선생님: 여러분, 안녕하세요! 지난 시간에는 파이썬으로 글자도 쓰고 계산도 해봤죠? 오늘은 더 신나는 걸 해볼 거예요! 바로 파이썬 거북이를 불러내서 그림을 그리는 거랍니다!

코딩이: 거북이요? 진짜 거북이가 컴퓨터에서 나와요? 느릿느릿 움직이나요?

슬기 선생님: (웃음) 진짜 거북이는 아니지만, 우리가 명령하는 대로 그림을 그려주는 아주 특별한 거북이 친구예요! 자, 먼저 우리 거북이 친구를 코랩으로 데려와야겠죠? 첫 번째 줄에 이렇게 써볼까요?

```
!pip install ColabTurtlePlus
```

지혜: 선생님! 느낌표는 뭐고 pip는 뭐예요? install은 설치한다는 뜻 같은데... ColabTurtlePlus는 거북이 이름인가요?

슬기 선생님: 와, 지혜 학생! 정말 날카로운 질문이에요! 여기서 느낌표(!)는 코랩에게 “이건 파이썬 명령어가 아니라 특별한 시스템 명령어예요!”라고 알려주는 신호예요. pip는 파이썬에 필요한 여러 가지 도구들을 설치해주는 로봇 친구라고 생각하면 돼요. 그리고 install ColabTurtlePlus는 “콜랩 터틀 플러스를 설치해줘!”라는 뜻입니다. ColabTurtlePlus가 바로 우리가 오늘 그림 그릴 때 사용할 거북이 도구의 이름이에요.

코딩이: 아하! 로봇 친구한테 거북이 도구를 가져오라고 시키는 거군요!

슬기 선생님: 맞아요, 코딩이 학생! 이제 거북이 도구를 가져왔으니, 우리 프로그램에서 이 도구를 사용하겠다고 알려줘야 해요. 다음 줄을 보세요.

```
from ColabTurtlePlus.Turtle import *
```

지혜: 선생님, from이랑 import는 또 뭐예요? 그리고 ColabTurtlePlus.Turtle 다음에 점(.)은 왜 찍어요? 별표(*)는 무슨 뜻이에요? 그리고 Turtle은 왜 대문자로 시작해요? 질문이 너무 많죠?

슬기 선생님: (웃음) 괜찮아요, 지혜 학생! 원래 똑똑한 학생은 질문이 많은 법이죠! 하나씩 설명해 줄게요. from ColabTurtlePlus.Turtle은 “우리가 설치한 ColabTurtlePlus 도구 상자 안에 있는 Turtle 파일에서”라는 뜻이에요. ColabTurtlePlus라는 큰 도구 상자 안에 Turtle이라는 작은 파일이 들어있는 거죠. 그리고 점(.)은 “~안에 들어있는” 또는 “~의” 라는 뜻으로 생각하면 쉬워요. 그래서 ColabTurtlePlus.Turtle은 “ColabTurtlePlus 도구 상자 안에 있는 Turtle 파일”이 되는 거죠. import *는 “그 파일 안에 있는 모든 것을 가져와서 쓸 수 있게 해줘!”라는 뜻이에요. 별표(*)는 ‘모든 것’이라는 의미입니다. 마치 마법 주문 같죠? 마지막으로 Turtle이 대문자로 시작하는 건, 약속이에요. 이렇게 여러 기능이 모여있는 특별한 도구나 설계를 만들 때는 첫 글자를 대문자로 써서 “이건 아주 중요한 거야!”라고 표시한답니다.

코딩이: 와... 뭔가 엄청 복잡한 주문 같은데요? 프롬 콜랩터틀플러스뽀터틀 임폴트 별! 외워야 하나요?

슬기 선생님: (웃음) 다 외울 필요는 없어요! 우리가 “거북아, 나와라!” 하고 주문을 외우는 거라고 생각하면 돼요. 이 주문 덕분에 이제 거북이의 여러 가지 신기한 능력을 사용할 수 있게 된답니다.

슬기 선생님: 자, 거북이를 부르기 전에 그림 그릴 도화지를 깨끗하게 해볼까요?

```
clearscreen()
```

지혜: clearscreen()은 화면을 깨끗하게 한다는 뜻인가요? 뒤에 괄호()는 왜 붙어요?

슬기 선생님: 맞아요! clearscreen은 “화면을 깨끗하게 지워줘!”라는 명령어예요. 그리고 명령어 뒤에 붙는 괄호()는 “지금 바로 이 명령을 실행해!”라는 뜻을 가진 버튼 같은 거예요. 우리가 TV 리모컨에서 전원 버튼을 누르는 것처럼요!

코딩이: 아하! 깨끗한 도화지 준비 완료! 이제 거북이 나와요?

슬기 선생님: 네, 드디어 주인공 등장입니다! 우리 아기 거북이를 만들어 볼게요.

```
baby = Turtle()
```

지혜: baby는 거북이 이름인가요? 그리고 Turtle()은 아까 import 할 때 봤던 대문자 Turtle이랑 똑같이 생겼는데, 이번에도 괄호()가 붙었네요!

슬기 선생님: 정확해요! baby는 우리가 만든 거북이에게 붙여준 이름이에요. 다른 이름으로 해도 된답니다. Turtle()은 “아까 우리가 가져온 Turtle 설계도대로 진짜 거북이 한 마리를 만들어줘!” 라는 뜻이에요. 여기서도 괄호()는 “만들어줘!” 하고 실행하는 버튼 역할을 하는 거죠. 이제 baby라는 이름의 우리 거북이가 화면 어딘가에 나타났을 거예요!

코딩이: 와! 진짜 거북이가 생겼어요! 선생님, 근데 거북이가 가만히 있는데요? 움직이게 하고 싶어요!

슬기 선생님: 좋아요! 우리 baby 거북이에게 명령을 내려볼까요? 앞으로 가라고 해봅시다!

```
baby.forward(400)
```

지혜: baby 다음에 점(.)이 또 나왔어요! 그리고 forward는 앞으로 가라는 뜻인가요? 괄호 안의 400은 뭐예요?

슬기 선생님: 네, 아주 잘 봤어요! baby.forward(400)은 ” baby 야, 앞으로 400만큼 가!” 라는 뜻이에요. 여기서 점(.)은 ” baby 거북이가 가지고 있는 능력 중에서 ” 라는 뜻이에요. baby 거북이는 forward라는 앞으로 가는 능력을 가지고 있는 거죠. forward는 앞으로 이동하라는 명령어이고, 괄호 안의 숫자 400은 얼마나 많이 갈지 알려주는 거예요. 400만큼 앞으로 쪽 가는 거죠!

(거북이가 앞으로 400만큼 이동한다)

코딩이: 우와! 거북이가 앞으로 갔어요! 신기하다! 그럼 뒤로도 갈 수 있어요?

슬기 선생님: 물론이지! 뒤로 가는 방법은 두 가지가 있어요. 하나는 forward에 음수 값을 넣어주는 거예요.

```
baby.forward(-200)
```

지혜: 아! forward인데 숫자가 -200이니깐 반대 방향으로 200만큼 가는 거군요!

슬기 선생님: 맞아요! 또는 back이라는 명령어를 사용할 수도 있답니다.

```
baby.back(200)
```

코딩이: 오! `baby.back(200)` 이라고 하니까 뒤로 200만큼 갔어요! `forward(-200)` 이랑 똑같네요!

슬기 선생님: 그렇죠! 이제 우리 거북이가 앞뒤로 자유롭게 움직일 수 있게 됐어요. 이번에는 방향을 바꿔볼까요? 오른쪽으로 돌아보게 할게요.

```
baby.right(90)
```

지혜: `baby.right(90)` 은 " baby 야, 오른쪽으로 90도 만큼 돌아!" 라는 뜻이겠죠?

슬기 선생님: 정답! `right` 는 오른쪽으로 회전하라는 명령어이고, `90` 은 90도 만큼 돌라는 뜻이에요. 직각으로 딱 꺾는 거죠!

(거북이가 오른쪽으로 90도 회전한다)

코딩이: 와! 거북이가 고개를 돌렸어요! 이제 앞으로 가면 다른 방향으로 가겠네요!

슬기 선생님: 그렇죠! 한번 앞으로 가게 해볼까요?

```
baby.forward(100)
```

(거북이가 회전한 방향으로 100만큼 이동한다)

지혜: 와! 진짜 옆으로 갔어요! 선생님, 그럼 왼쪽으로 도는 건 `left()` 인가요?

슬기 선생님: 맞아요, 지혜 학생! `left()` 명령어도 있습니다. 오늘 우리는 거북이를 만들고, 앞으로 가게 하고, 뒤로 가게 하고, 방향도 바꿔봤어요. 어때요, 재미있었나요?

코딩이: 네! 너무 신기해요! 제가 시키는 대로 거북이가 움직이니까 제가 마법사가 된 것 같아요!

슬기 선생님: (웃음) 맞아요, 프로그래밍은 마치 마법과 같답니다! 앞으로 이 거북이로 여러 가지 도형도 그리고, 더 복잡한 그림도 그려볼 거예요. 기대해도 좋아요!

지혜 & 코딩이: 네! 다음 시간이 기다려져요!