

<[파이썬 그림 마법] 제1장. 안녕, 터틀! 그림을 그려보자!>

선생님: (색연필과 스케치북을 들고) 애들아, 안녕! 우리가 지금까지 코드로 글자도 만들고 계산도 했지? 오늘은 이 스케치북처럼 하얀 도화지에 우리가 명령하는 대로 그림을 그려주는 아주 특별한 친구, **터틀(Turtle, 거북이)**을 만나볼 거야! 🐢🐢

코딩새싹 🌱: 거북이가 그림을 그려요? 정말 신기한데요! 어떻게 하는 거예요?

선생님: ColabTurtlePlus라는 마법 도구 상자를 이용하면 된단다! 먼저, 이 도구 상자를 우리 작업 공간에 설치하는 주문부터 외워볼까? (Colab 환경에서만 처음 한 번 필요해!)

```
# !pip install ColabTurtlePlus # Colab 환경에서 이 도구 상자를 설치하는 마법 주문!
```

선생님: 이제 그림 그리는 데 필요한 모든 도구를 가져와야겠지?

```
from ColabTurtlePlus.Turtle import * # 터틀 그림 마법 도구를 모두 가져와!
```

<1. 우리의 캔버스, 스크린 준비하기!>

선생님: 그림을 그리려면 먼저 도화지가 필요하겠지? 터틀 그래픽에서는 이 도화지를 **스크린(Screen)**이라고 불러.

```
# clearscreen() # 혹시 이전에 그린 게 있다면 깨끗이 지우고 시작!
```

```
screen = Screen() # 스크린 도화지를 하나 만들고,  
screen_width = 800 # 가로 크기는 800,  
screen_height = 600 # 세로 크기는 600으로 정하자!  
screen.setup(screen_width, screen_height) # 도화지 크기 설정!  
screen.bgcolor("black") # 도화지 배경색은 검은색으로! 밤하늘 같지?
```

궁금이 💡: Screen()으로 도화지를 만들고, setup()으로 크기를, bgcolor()로 배경색을 정하는 거군요! 쉽네요!

<2. 오늘의 화가, 터틀 등장!>

선생님: 자, 이제 이 도화지에 그림을 그려줄 우리의 꼬마 화가, 터틀을 불러내자!

```
king = Turtle() # 'king'이라는 이름의 터틀을 만들자!  
king.shape("turtle") # 모양은 진짜 거북이 모양으로! (다른 모양도 있어!)
```

```
king.speed(10) # 거북이가 그림 그리는 속도! (1~10, 0은 최고 속도)
```

```
king.color("yellow") # 거북이가 그릴 선의 색깔은 노란색!
```

코딩새싹 🌱: 와! king이라는 이름의 거북이가 생겼어요! 이제 이 거북이한테 명령을 내리면 되는 건가요?

<3. 거북이의 기본 동작: 전진! 회전! 그리고 색칠!>

선생님: 맞아! king에게 “앞으로 가!”, “왼쪽으로 돌아!” 이렇게 명령을 내릴 수 있어. 한번 사각형을 그려볼까?
그리고 색도 채워보자!

```
# king.forward(100) # 앞으로 100만큼 가!
```

```
# king.left(90)      # 왼쪽으로 90도 돌아!
```

```
# king.forward(100)
```

```
# king.left(90)
```

```
# ... 이렇게 네 번 반복하면 사각형이 되겠지? for문을 쓰면 더 좋겠다!
```

```
king.begin_fill() # "지금부터 그리는 도형은 색칠할 준비!"
```

```
for i in range(4): # 네 번 반복하면서
```

```
    king.forward(100) # 앞으로 100만큼 가고
```

```
    king.left(90)     # 왼쪽으로 90도 돌기!
```

```
king.end_fill()    # "색칠 끝!"
```

궁금이 💡: begin_fill()과 end_fill() 사이에 그린 도형은 자동으로 king의 색깔인 노란색으로 칠해지는군요! for 반복문으로 사각형 그리니까 코드도 짧아졌어요!

<4. 하늘을 나는 거북이: 펜 들고 특정 위치로 순간이동!>

선생님: 우리 king 거북이는 그림을 그리면서 움직이지만, 때로는 선을 그리지 않고 다른 장소로 획~ 이동하고 싶을 때도 있겠지?

```
king.penup() # "거북아, 펜 들어!" (이제부터 움직여도 그림이 안 그려져)
```

```
king.goto(-100, 100) # (x좌표, y좌표)로 순간이동! (화면 중앙이 0,0 이야)
```

```
king.pendown() # "거북아, 펜 내려!" (이제부터 그림을 그린다!)
```

```
king.color("green") # 이번엔 초록색으로 원을 그려볼까?
```

```
king.begin_fill()
king.circle(100) # 반지름이 100인 원을 그려줘!
king.end_fill()
```

코딩새싹 🌱: penup()이랑 pendown() 정말 신기해요! goto()로 원하는 곳으로 바로 갈 수도 있고요! 화면 중앙이 (0,0)이면, goto(-100, 100)은 왼쪽 위로 간 거네요!

<5. 다 함께 그리는 대작! 멋진 건물 만들기!>

선생님: 자, 이제 배운 기술들을 총동원해서 여러 층으로 된 멋진 건물을 함께 그려보자! (선생님은 화면을 깨끗이 지우고 새 도화지를 준비한다)

```
clearscreen() # 이전 그림은 모두 지우고!
```

```
screen = Screen()
screen.setup(800, 600)
screen.bgcolor("black")
```

```
baby = Turtle() # 이번엔 'baby'라는 이름의 터틀!
baby.speed(20) # 속도는 조금 빠르게!
baby.color("deep sky blue") # 아기 거북이의 첫 색깔!
```

```
# (0, 20)으로 이동해서, 반지름 100인 원 그리기 (건물의 둥근 지붕이라고 상상해볼까?)
```

```
baby.penup()
baby.goto(0, 20)
baby.pendown()
baby.begin_fill()
baby.circle(100)
baby.end_fill()
```

```
# --- 건물의 여러 층을 차례대로 그려보자! ---
```

```
# 각 층을 그릴 때마다 baby.penup(), baby.goto(), baby.pendown(), baby.color(), baby.begin_fill(), for문으
```

```

# 맨 밑단 그리기

baby.penup()

baby.goto(-200, -200)

baby.pendown()

baby.color("indigo")

baby.begin_fill()

for i in range(2): # 사각형 그리는 반복문

    baby.forward(400)

    baby.left(90)

    baby.forward(20)

    baby.left(90)

baby.end_fill()


# (이와 유사한 방식으로 중간층, 몸체, 상단층들을 차례대로 그립니다)

# 밑에서 두번째 단 그리기 (색: chocolate, 위치와 크기 조절)

baby.penup(); baby.goto(-175, -180); baby.pendown(); baby.color("chocolate"); baby.begin_fill()

for i in range(2): baby.forward(350); baby.left(90); baby.forward(20); baby.left(90)

baby.end_fill()


# 건물 몸체 (색: olive, 위치와 크기 조절)

baby.penup(); baby.goto(-150, -160); baby.pendown(); baby.color("olive"); baby.begin_fill()

for i in range(2): baby.forward(300); baby.left(90); baby.forward(250); baby.left(90)

baby.end_fill()


# 상단 두번째 단 (색: chocolate, 위치와 크기 조절)

baby.penup(); baby.goto(-175, 90); baby.pendown(); baby.color("chocolate"); baby.begin_fill()

for i in range(2): baby.forward(350); baby.left(90); baby.forward(20); baby.left(90)

baby.end_fill()

```

```
# 최상단 (색: pale goldenrod, 위치와 크기 조절)

baby.penup(); baby.goto(-150, 110); baby.pendown(); baby.color("pale goldenrod"); baby.begin_fill()

for i in range(2): baby.forward(300); baby.left(90); baby.forward(20); baby.left(90)

baby.end_fill()
```

궁금이 💡: 우와! goto()로 위치를 잘 잡고, for문으로 사각형을 차곡차곡 쌓으니까 정말 건물이 만들어지고 있어요! 같은 패턴이 반복되는데, 이것 좀 더 편하게 할 방법은 없을까요?

<6. 똑똑하게 반복하기: 함수로 창문 그리기!>

선생님: 바로 그거야, 궁금아! 똑같은 모양을 여러 번 그려야 한다면, 그리는 방법을 함수로 만들어두면 정말 편하겠지? 건물에 창문을 한번 달아볼까?

```
# 창문 그리는 함수 만들기!

x = -125 # 창문 시작 x좌표 (이 값을 함수 안에서 바꿔야 하니, 전역 변수로!)

y = 30 # 창문 시작 y좌표
```

```
baby.color("yellow") # 창문은 노란색!
```

```
def draw_window(turtle_artist): # 그림 그릴 터틀을 재료로 받자!

    global x # 함수 안에서 바깥 세상의 x 값을 바꿀 거라고 알려주기!

    turtle_artist.penup()

    turtle_artist.goto(x, y)

    turtle_artist.pendown()

    turtle_artist.begin_fill()

    for i in range(4): # 정사각형 창문

        turtle_artist.forward(40)

        turtle_artist.left(90)

    turtle_artist.end_fill()
```

```
x = x + 70 # 다음 창문은 오른쪽으로 70만큼 이동해서 그리도록 x좌표 변경!
```

```

# 이제 함수를 호출해서 창문을 그려보자!

# 한 줄에 창문 4개씩, 총 3줄을 그려볼까?

for row_num in range(3): # 3줄 반복

    for window_num in range(4): # 한 줄에 창문 4개씩 반복

        draw_window(baby) # baby 터틀에게 창문 그리라고 시키기!


# 한 줄 다 그렸으면, 다음 줄을 위해 x좌표는 처음으로 돌리고, y좌표는 낮춰야겠지?

x = -125

y = y - 85 # y좌표를 아래로 이동 (숫자가 작아지면 아래로)

```

코딩새싹 🌱: draw_window 함수를 만드니까 코드가 훨씬 깔끔해졌어요! for문으로 함수를 여러 번 부르니까 창문이 순식간에 생기네요! global x는 뭐예요?

선생님: 함수 안에서 바깥 세상에 만들어진 x 변수의 값을 직접 바꾸고 싶을 때, “내가 지금부터 사용하는 x는 바깥 세상의 그 x입니다!” 라고 알려주는 특별한 선언이야. (사실, global은 꼭 필요할 때만 신중하게 써야 하는 마법이란다. 더 좋은 방법들도 있지만, 지금은 이 코드를 이해하는 데 집중해보자!)

궁금이 💡: 중첩 for문으로 행과 열을 만들어서 창문을 배치하니까 정말 멋진데요! x는 가로 위치, y는 세로 위치를 조절하는 거군요!

선생님: (흐뭇하게) 아주 잘했어! 오늘 우리는 터틀 그래픽의 기본부터 시작해서, 함수와 반복문을 이용해 복잡한 그림까지 그려봤어. 코드가 그림이 되는 마법, 정말 신기하고 재미있지 않니? 이제 여러분도 상상하는 모든 것을 터틀로 그려볼 수 있을 거야!