

<[파이썬 그림 마법] 제3장. 반짝반짝 별이 빛나는 밤하늘 그리기!>

선생님: (밤하늘 그림이 그려진 스케치북을 보여주며) 애들아, 안녕! 밤하늘을 본 적 있니? 수많은 별들이 반짝이고, 때로는 달이 아름답게 떠 있는 모습, 정말 황홀하지? 오늘은 우리 터틀 친구 'paul'과 함께 이 멋진 밤하늘을 직접 그려볼 거야!

코딩새싹 🌱: 와! 컴퓨터 화면에 별이랑 달을 그리는 거예요? 정말 기대돼요!

선생님: 먼저, 그림을 그릴 도화지(스크린)와 우리 화가 터틀 'paul'을 준비해야겠지? 밤하늘이니까 배경은 까만색으로 하자!

<1. 밤의 도화지 준비하기 (스크린과 터틀 설정 복습!)>

```
# !pip install ColabTurtlePlus # Colab 환경에서는 처음에 한 번만!
```

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

```
import random # 별의 위치나 크기, 색을 무작위로 정하기 위해!
```

```
clearscreen() # 이전 그림이 있다면 깨끗하게!
```

```
screen = Screen()
```

```
screen_width = 800
```

```
screen_height = 500
```

```
screen.setup(screen_width, screen_height)
```

```
screen.bgcolor("black") # 배경은 까만 밤하늘!
```

```
paul = Turtle() # 우리 화가 터틀, 'paul' 등장!
```

```
paul.speed(20) # 그림 그리는 속도는 빠르게! (0이 최고 속도, 10도 빨라요)
```

```
# paul.hideturtle() # 그림 그리는 동안 거북이를 숨기고 싶다면 이 주문을! (선택)
```

궁금이 💡: import random은 왜 하는 거예요? 별들이 다 똑같은 위치에 똑같은 모양으로 뜨면 재미없으니까, 무작위로 만들려고 하는 거죠?

선생님: 정확해! random 마법사를 이용하면 별들의 위치, 크기, 색깔을 제멋대로 정해서 진짜 밤하늘처럼 자연스럽게 만들 수 있단다.

<2. 반짝이는 별을 그리는 마법 주문, star() 함수 만들기!>

선생님: 별 하나를 그리는 것도 꽤 여러 단계가 필요해. 만약 별을 30개 그린다면 똑같은 코드를 30번이나 써야겠지? 이럴 때 뭐가 필요하다고 했더라?

코딩새싹 🌱: 함수요! 별 그리는 방법을 함수로 만들어두면 이름만 불러서 쓸 수 있잖아요!

선생님: 맞아! star(크기, 색깔) 이렇게 호출하면 원하는 크기와 색깔의 별을 똑딱 그려주는 함수를 만들어보자. 별은 뾰족뾰족한 모양이니까, 각도를 잘 맞춰서 그려야 해!

```
def star(size, color): # 별의 크기와 색깔을 재료로 받는다!

    paul.color(color) # 먼저 펜 색깔을 정하고,

    # 별의 바깥 뾰족한 부분 그리기

    paul.begin_fill() # 색칠 준비!

    for i in range(5): # 별은 보통 오각형 모양이니까 5번 반복!

        paul.forward(10 * size) # size 만큼 앞으로 쪽~ (size가 1이면 10만큼)

        paul.right(144)      # 오른쪽으로 144도 회전! (이게 별 모양의 비밀 각도!)

    paul.end_fill() # 색칠 끝!

    # 별 가운데 부분에 작은 오각형 장식 추가 (선택 사항)

    paul.forward(3.8 * size) # 살짝 안으로 이동해서

    paul.begin_fill()

    for i in range(5):

        paul.forward(2.4 * size)

        paul.right(360 / 5) # 정오각형의 외각은 360/5 = 72도!

    paul.end_fill()
```

궁금이 💡: paul.right(144) 저 각도가 별을 만드는 핵심이군요! 그리고 별 가운데에 작은 오각형을 또 그리는 것도 재미있어요! size 값을 바꾸면 별 크기가 달라지겠네요!

<3. 밤하늘을 수놓는 별들: 무작위로 여러 별 그리기!>

선생님: 자, 이제 우리 star 함수 마법 주문으로 밤하늘을 가득 채워볼까? for 반복문과 random 마법사를 함께 사용해서 다양한 별들을 그려보자!

```

for i in range(30): # 별을 30개 그려보자!

    # 별이 나타날 무작위 위치 (x, y 좌표) 정하기

    # 화면 가로가 800이니까 x는 -400 ~ 400, 세로가 500이니까 y는 -250 ~ 250 사이로!

    x = random.randint(-400, 400)

    y = random.randint(-250, 250)


    # 별의 크기도 무작위로! (1, 2, 3 중에 하나. 작은 별이 더 많도록 1을 여러 번 넣어줌)

    star_size = random.choice([1, 1, 1, 1, 1, 1, 2, 2, 2, 3])

    # 별의 색깔도 무작위로! (흰색, 노란색, 파란색 중에 하나)

    star_color = random.choice(["white", "white", "yellow", "yellow", "yellow", "blue"])


    paul.penup() # 펜을 들고 별 그릴 위치로 순간이동!

    paul.goto(x, y)

    # paul.pendown() # star 함수 안에서 펜을 내리므로, 여기서 꼭 안 해도 될 수 있어. (함수 구현에 따라 다름)


    star(star_size, star_color) # 드디어 별 그리기 함수 호출!

```

코딩새싹 🌱: random.randint()로 x, y 좌표를 마구잡이로 정하고, random.choice()로 크기랑 색깔도 마음대로 고르니까 정말 별들이 제멋대로 반짝이는 것 같아요!

<4. 밤하늘의 주인공, 달 그리기! (동그라미 겹치기 마법!)>

선생님: 반짝이는 별들만 있으면 조금 아쉽지? 아름다운 달도 그려 넣어보자! 초승달이나 그믐달 모양은 어떻게 그릴 수 있을까? 동그라미 두 개를 겹쳐 그리는 마법을 쓰면 된단다!

```

# 달을 그릴 위치로 paul을 이동시키자!

paul.penup()

paul.goto(-300, 100) # 화면 왼쪽 위에 그려볼까?

paul.pendown()


# 1. 먼저 하얀색으로 짝 찬 보름달을 그린다!

paul.color("white")

```

```
paul.begin_fill()
paul.circle(50) # 반지름 50짜리 원
paul.end_fill()
```

2. 그 위에, 살짝 옆으로 옮겨서 밤하늘색(검은색) 동그라미를 겹쳐 그린다!

```
paul.penup()
paul.goto(-270, 120) # 하얀 달보다 살짝 오른쪽 위로 이동
paul.pendown()
paul.color("black") # 밤하늘 색과 같은 색!
paul.begin_fill()
paul.circle(50) # 똑같은 크기의 원
paul.end_fill()
```

```
paul.hideturtle() # 달까지 다 그렸으니 우리 화가 paul은 이제 쉬도록 숨겨주자!
```

궁금이 💡: 아하! 하얀 동그라미 위에 검은 동그라미를 살짝 겹쳐서 그리니까, 가려지지 않은 하얀 부분만 보여서 초승달 모양이 되는 거군요! 정말 기발한 아이디어인데요!

선생님: (밤하늘 그림을 가리키며) 자, 이렇게 해서 우리만의 반짝이는 밤하늘이 완성되었어! 함수로 별을 만들고, random으로 다양성을 주고, 동그라미 두 개로 달까지 표현했지. 코드가 만들어낸 예술 작품 같지 않니?

코딩새싹 🌱: 네, 선생님! 제가 직접 그린 밤하늘이라니, 너무 뿌듯해요! 이제 터틀로 뭐든지 그릴 수 있을 것 같아요!

선생님: (활짝 웃으며) 물론이지! 오늘 배운 기술들을 응용하면 더 복잡하고 아름다운 그림도 얼마든지 그릴 수 있단다. 우주선을 그려 넣을 수도 있고, 반짝이는 효과를 더 줄 수도 있겠지! 여러분의 상상력을 마음껏 펼쳐보렴! 오늘 정말 멋진 밤하늘을 만들어냈어! 🌟🌙