

파이썬 거북이, for문으로 그리는 점점 작아지는 원들과 반짝이는 별! 🐢✨🌀

등장인물:

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

슬기 선생님: 여러분, 안녕하세요! for 반복 마법사와 circle() 함수로 멋진 원을 그려봤죠? 오늘은 for 반복문을 더욱 다양하게 활용해서, 크기가 점점 작아지는 여러 개의 원을 그려보고, 밤하늘의 별처럼 반짝이는 별 모양도 그려볼 거예요!

코딩이: 와! 점점 작아지는 원이요? 마술피리 같아요! 별은 어떻게 그려요? 뽕뽕뽕뽕한 모양인데, 거북이가 잘 그릴 수 있을까요?

슬기 선생님: (웃음) 그럼요! 우리 david 거북이는 이제 for 반복문의 달인이 다 되었거든요. 먼저, 점점 작아지는 원들을 그려볼까요? 배경색은 회색으로 하고, 원은 초록색으로 채워볼게요.

1. for문과 함께! 점점 작아지는 동글이 행진!

```
clearscreen()

screen = Screen()
screen.bgcolor("gray") # 배경은 차분한 회색!

david = Turtle()
david.speed(10)
david.pensize(5)
david.color("white", "green") # 선은 하얀색, 채우기는 초록색!

david.penup()
david.goto(-200, -100) # 왼쪽 아래에서 시작해 볼까요?
david.pendown()

# for i in range(10): # 이 코드는 10개의 원을 그려요! (만약 4개만 그리려면 range(4)로!)
for i in range(4): # 설명을 위해 4개만 그려볼게요!
    david.begin_fill()
    david.circle(150 - i * 10) # i가 커질수록 원이 작아져요!
    david.end_fill()
    david.penup()
    david.forward(70 + i * 10) # i가 커질수록 다음 원까지 더 멀리 이동!
    david.pendown()
```

지혜: 선생님! `for i in range(4):`로 네 번 반복하네요! (원래 코드에서는 `range(10)`이었지만, 설명을 위해 4로 줄였어요!) 그리고 `david.circle(150 - i * 10)` 이 부분이 신기해요! `i` 값에 따라 원의 크기가 달라지는 거죠?

슬기 선생님: 맞아요, 지혜 학생! `i`는 `for` 반복문이 돌 때마다 0, 1, 2, 3 이렇게 값이 변하잖아요?

- 첫 번째 바퀴(`i`가 0일 때): `david.circle(150 - 0 * 10)`이니까 반지름 150인 원을 그려요.
- 두 번째 바퀴(`i`가 1일 때): `david.circle(150 - 1 * 10)`이니까 반지름 140인 원을 그리고요.
- 세 번째 바퀴(`i`가 2일 때): `david.circle(150 - 2 * 10)`이니까 반지름 130!
- 네 번째 바퀴(`i`가 3일 때): `david.circle(150 - 3 * 10)`이니까 반지름 120인 원을 그리게 되죠!
이렇게 `i`가 커질수록 10씩 빼주니까 원의 크기가 점점 작아지는 마법이 일어나는 거예요!

코딩이: 와! 진짜 원이 점점 작아지면서 옆으로 이동해요! `david.forward(70 + i * 10)` 이것도 `i` 때문에 움직이는 거리가 달라지는 거죠? 처음엔 70만큼 갔다가, 다음엔 80, 그 다음엔 90, 100만큼 가네요!

슬기 선생님: 정확해요, 코딩이 학생! 원이 작아지는데, 다음 원을 그릴 위치까지의 거리는 점점 늘어나도록 해서 원들이 겹치지 않고 나란히 보이도록 한 거랍니다. `begin_fill()`과 `end_fill()`로 각 원을 초록색으로 예쁘게 채웠고요. `penup()`과 `pendown()`으로 원과 원 사이를 이동할 때는 선이 그려지지 않도록 하는 것도 잊지 않았죠!

지혜: `for` 반복문 안에서 `i` 값을 이렇게 다양하게 활용하니까 정말 동적인 그림이 만들어지네요!

슬기 선생님: 그렇죠? 자, 그럼 이제 두 번째 도전! 반짝반짝 별을 그려볼 시간이에요!

2. `for`문으로 그리는 반짝반짝 별님!

```
clearscreen()
```

```
david = Turtle()
```

```
david.speed(10)
```

```
david.color("red") # 별은 강렬한 빨간색으로! (선과 채우기 모두 빨강)
```

```
david.begin_fill() # "지금부터 그릴 별 전체를 채울 거야!"
```

```
for i in range(5): # 별은 꼭짓점이 5개니까 5번 반복!
```

```
    david.forward(200) # 별의 한 변을 꼭 그리고!
```

```
    david.right(144) # 오른쪽으로 144도 휙! (이게 별을 만드는 마법의 각도!)
```

```
david.end_fill() # "별 다 그렸으니 빨간색으로 채워줘! 짠!"
```

코딩이: 와! 진짜 별이 그려졌어요! `david.forward(200)`으로 앞으로 갔다가 `david.right(144)`로 꺾는 걸 5번 반복하니까 뾰족뾰족 별 모양이 됐어요! 144도는 어떻게 찾은 거예요? 마법의 각도인가요?

슬기 선생님: (웃음) 맞아요, 코딩이 학생! 별을 그릴 때 144도는 정말 마법의 각도 같은 거예요. 이 각도로 5번 회전하면서 선을 그으면 정확히 처음 시작점으로 돌아오면서 예쁜 별 모양이 만들어진답니다. `for i in range(5)`로 5번 반복하는 이유는 별의 꼭짓점이 5개이기 때문이고요!

지혜: 선생님, 여기서 `begin_fill()`이랑 `end_fill()` 위치가 중요할 것 같아요! 만약에 `end_fill()`을 `for`

반복문 안에 넣으면 어떻게 돼요? 가끔 친구들이 그렇게 실수하는 걸 봤어요.

슬기 선생님: 지혜 학생, 정말 중요한 질문을 해줬어요! 많은 친구들이 헛갈려 하는 부분입니다. 만약 `end_fill()`을 `for` 반복문 안에 넣으면, 즉 이렇게 코드를 짜면요:

```
# 잘못된 예시: end_fill()이 for문 안에 있는 경우
# david.begin_fill() # 이것도 반복문 안에 같이 들어가면 더 이상해져요!
# for i in range(5):
#     david.begin_fill() # 만약 이것도 안에 있다면 매번 새로 시작
#     david.forward(200)
#     david.right(144)
#     david.end_fill() # <- 여기가 문제!
```

이렇게 되면, `david.forward(200)`하고 `david.right(144)`로 별의 한쪽 선분과 꺾임만 그린 상태에서 `end_fill()`을 만나게 돼요. 그럼 컴퓨터는 “어? 아직 별 모양이 다 안 만들어졌는데 색을 채우라고?” 하면서, 그 순간까지 그려진 아주 작은 삼각형이나 이상한 부분만 채우려고 할 거예요. 그래서 별 전체가 예쁘게 채워지는 게 아니라, 조각조각 이상하게 채워지거나 아예 제대로 안 채워질 수 있습니다.

코딩이: 앗! 그럼 `begin_fill()`도 `for`문 밖에 있어야 해요?

슬기 선생님: 네, 맞아요! 우리가 **하나의 완성된 별 전체**를 채우고 싶은 거죠? 그렇다면,

1. 별을 그리기 **시작하기 직전에** `david.begin_fill()`을 딱 한 번 호출해서 “이제부터 그리는 도형 전체를 채울 준비!”라고 알려주고,
2. `for` 반복문으로 별의 5개 꼭짓점을 **모두 다 그리고 나서**, 별 모양이 완성된 후에
3. `david.end_fill()`을 딱 한 번 호출해서 “자, 이제 이 완성된 별 안쪽을 채워줘!” 라고 명령해야 하는 거예요.

그래서 `begin_fill()`은 `for`문 **앞에**, `end_fill()`은 `for`문이 **끝난 뒤에** 위치하는 것이랍니다!

지혜: 아하! `begin_fill()`과 `end_fill()`은 항상 짝궁처럼, 채우고 싶은 도형 전체를 감싸줘야 하는 거군요! `for`문으로 그 도형을 그릴 때는 `for`문 바깥에서 시작하고 끝내야 하고요!

슬기 선생님: 정확합니다! 오늘 우리는 `for` 반복문의 변수 `i`를 이용해서 그림의 크기나 위치를 바꾸는 방법도 배우고, 별을 그리는 원리와 함께 `begin_fill()`, `end_fill()`의 올바른 사용법까지 확실하게 익혔어요. 이제 여러분은 `for` 반복문과 거북이로 더욱 다양하고 멋진 작품들을 만들 수 있을 거예요!

코딩이 & 지혜: 네! 선생님! 오늘 배운 걸로 더 멋진 그림들을 도전해보아야겠어요! 너무 재미있어요!