

파이썬 거북이, 웅장한 국회의사당을 짓다! 🏛️🐢✨ 함수와 글로벌 변수의 비밀!

등장인물:

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

슬기 선생님: 여러분, 안녕하세요! 우리가 함수와 반복문 마법까지 배우면서 정말 많은 것을 할 수 있게 됐죠? 오늘은 그동안 배운 모든 기술을 총동원해서, 아주 웅장하고 멋진 건물, 바로 국회의사당을 그려볼 거예요! (선생님이 국회의사당 그림을 보여준다 - 사용자가 제공한 이미지 참고)

코딩이: 와! 저렇게 멋진 건물을 진짜 우리가 그릴 수 있어요? 돔 지붕도 있고, 창문도 엄청 많고, 계단 같은 받침대도 있는데요! paul 거북이가 오늘 완전 건축 총감독이네요!

슬기 선생님: (웃음) 맞아요! paul 거북이가 오늘 아주 큰 프로젝트를 맡았습니다. 코드를 보면 알겠지만, 건물은 여러 부분으로 나뉘어 그려져요. 맨 위에 동그란 돔 지붕, 그 아래 여러 층의 받침대와 건물 몸통, 그리고 아주 많은 창문들까지!

1. 국회의사당 골격 만들기!

```
clearscreen()
```

```
# 스크린 생성, 배경 지정
```

```
screen = Screen()
```

```
screen.setup(800, 600)
```

```
screen.bgcolor("black") # 밤하늘 배경!
```

```
#터틀 생성
```

```
paul = Turtle()
```

```
paul.speed(20) # 아주 빠르게!
```

```
# 원을 그려주세요. (돔 지붕)
```

```
paul.penup()
```

```
paul.goto(0, 20) # 돔의 중심을 살짝 위로
```

```
paul.pendown()
```

```
paul.color("deep sky blue") # 하늘색 돔
```

```
paul.begin_fill()
```

```
paul.circle(100) # 반지름 100
```

```
paul.end_fill()
```

```
# 맨 밑단 그리기
```

```

paul.penup()
paul.goto(-200, -200)
paul.pendown()
paul.color("slate gray") # 짙은 회색 받침
paul.begin_fill()
for i in range(2): # 사각형 그리기
    paul.forward(400)
    paul.left(90)
    paul.forward(20)
    paul.left(90)
paul.end_fill()

```

(이어서 밑에서 두번째 단, 건물 몸체, 위에서 두번째 단, 위에서 첫번째 단 그리는 코드가 순서대로 나와요!)

(각 부분은 알맞은 색과 크기로 사각형을 그려서 층층이 쌓아 올리는 방식입니다!)

밑에서 두번째 단 그리기

```

paul.penup()
paul.goto(-175, -180)
paul.pendown()
paul.color("dark khaki")
paul.begin_fill()
for i in range(2):
    paul.forward(350)
    paul.left(90)
    paul.forward(20)
    paul.left(90)
paul.end_fill()

```

#건물 몸체 그리기

```

paul.penup()
paul.goto(-150, -160)
paul.pendown()
paul.color("teal")
paul.begin_fill()
for i in range(2):
    paul.forward(300)
    paul.left(90)
    paul.forward(250)
    paul.left(90)
paul.end_fill()

```

```
# 위에서 두번째 단 그리기
paul.penup()
paul.goto(-175, 90)
paul.pendown()
paul.color("dark khaki")
paul.begin_fill()
for i in range(2):
    paul.forward(350)
    paul.left(90)
    paul.forward(20)
    paul.left(90)
paul.end_fill()
```

```
# 위에서 첫번째 단 그리기
paul.penup()
paul.goto(-150, 110)
paul.pendown()
paul.color("chocolate")
paul.begin_fill()
for i in range(2):
    paul.forward(300)
    paul.left(90)
    paul.forward(20)
    paul.left(90)
paul.end_fill()
```

지혜: 네! goto()로 위치를 옮겨가면서 circle()로 동을 그리고, for문으로 사각형들을 그려서 여러 층을 쌓아 올리는 방식이네요! 색깔도 다 다르게 지정하고, begin_fill()이랑 end_fill()로 채우기도 하고요. 여기까지는 우리가 배운 내용으로 충분히 이해할 수 있어요!

슬기 선생님: 아주 잘 따라오고 있네요! 이 건물에서 가장 눈에 띄는 것 중 하나가 바로 여러 개의 창문들이죠? 이 창문들을 그리기 위해 함수를 사용해 볼 거예요.

2. 창문 그리기 함수 window()와 글로벌 변수의 비밀!

슬기 선생님: 창문은 모양이 다 똑같으니까 함수로 만들어두면 편리하겠죠? 그런데 창문은 여러 줄에 걸쳐서, 각 줄마다 여러 개씩 그려야 해요. 그래서 창문의 위치를 기억하고 다음 창문 위치를 계산하는 게 중요하답니다.

$x = -125$ # 첫 번째 창문이 그려질 가로(x) 위치의 시작점

$y = -100$ # 첫 번째 창문 줄이 그려질 세로(y) 위치의 시작점

#window를 함수로 짜기

```
def window():
    global x # "함수 안에서 쓰는 x는 바깥에 있는 x랑 똑같은 거예요!" 라고 선언!
    paul.penup()
    paul.goto(x, y) # 현재 x, y 위치로 가서 창문 그릴 준비
    paul.pendown()
    paul.color("pale green") # 연두색 창문
    paul.begin_fill()
    for i in range(4): # 정사각형 창문 그리기
        paul.forward(40)
        paul.right(90)
    paul.end_fill()

    x = x + 70 # 다음 창문을 그리기 위해 x 위치를 오른쪽으로 70만큼 이동!
```

코딩이: window() 함수 안에서 global x 라는 처음 보는 명령어가 나왔어요! global이 뭐예요? 그리고 함수 안에서 x = x + 70 이렇게 x 값을 바꾸네요!

슬기 선생님: 아주 중요한 부분을 발견했어요, 코딩이 학생! 보통 함수 안에서 만들어지거나 사용되는 변수는 그 함수 안에서만 의미가 있어요. 함수 바깥에 있는 똑같은 이름의 변수와는 다른 존재죠. 이것 '지역 변수(Local Variable)'라고 해요. 그런데 window() 함수는 창문을 하나 그리고 나서 다음 창문을 그릴 x 위치를 계산해서 바꿔줘야 하잖아요? 만약 global x 라는 선언 없이 그냥 함수 안에서 x = x + 70을 하면, 함수는 "아, window 함수 안에서만 쓸 새로운 x를 만들어서 70을 더해야겠다!" 라고 생각해요. 그러면 함수 바깥에 우리가 맨 처음에 정해둔 x = -125 값은 전혀 바뀌지 않겠죠. 그럼 어떻게 될까요?

지혜: 어... 그럼 window() 함수를 여러 번 불러도 x 값이 계속 -125에서 시작하니까... 창문이 전부 다 똑같은 가로 위치에 겹쳐서 그려지겠는데요?

슬기 선생님: 정확해요! 그래서 global x 라고 써주는 것은, "window 함수 안에서 내가 사용하는 x는 함수 안에서 새로 만든 지역 변수가 아니라, 저기 함수 바깥에 있는 전역 변수(Global Variable) x를 직접 가져다 쓰는 거야! 그러니까 내가 여기서 x 값을 바꾸면 바깥에 있는 x 값도 진짜로 바뀌는 거야!" 라고 파이썬에게 알려주는 거랍니다. 그래서 창문 하나 그리고 x가 70만큼 오른쪽으로 이동하고, 다음 창문은 그 이동된 x 위치에서 시작할 수 있는 거죠. y 값은 이 window 함수 안에서 직접 바뀌지는 않고 읽기만 하기 때문에 global y 선언이 필수는 아니었지만, 만약 window 함수가 y 값도 바꿨다면 global y도 필요했을 거예요.

코딩이: 아하! global은 함수 안이랑 함수 바깥이랑 똑같은 변수 이름표를 쓰게 해주는 마법 주문 같은 거네요!

슬기 선생님: 맞아요! 하지만 global 변수를 너무 많이 사용하면 코드가 복잡해지고 어디서 값이 바뀌는지 추적하기 어려워질 수도 있어요. 그래서 함수가 외부의 변수를 직접 바꾸는 것보다는, **함수에 필요한 정보를 파라미터(매개변수)로 전달하고, 함수가 계산한 결과를 return으로 돌려받는 방식**을 더 많이 사용한답니다. 예를 들어 window 함수를 이렇게 바꿀 수도 있어요.

```
# # global 없이 파라미터와 return을 사용하는 window 함수 (예시)
# def window_alternative(current_x, current_y):
```

```

# paul.penup()
# paul.goto(current_x, current_y)
# paul.pendown()
# paul.color("pale green")
# paul.begin_fill()
# for i in range(4):
#     paul.forward(40)
#     paul.right(90)
# paul.end_fill()
# return current_x + 70 # 다음 x 위치를 계산해서 돌려줌

```

```

# # 이렇게 호출하겠죠?
# # x = -125
# # y = -100
# # x = window_alternative(x, y) # x 값을 업데이트
# # x = window_alternative(x, y) # 또 x 값을 업데이트

```

지혜: 와! window_alternative(current_x, current_y) 이렇게 하면 함수가 현재 x, y 값을 받아서 창문을 그리고, 다음 창문이 그려질 새로운 x 값을 return으로 알려주네요! 그럼 함수를 부르는 쪽에서 x = window_alternative(x, y) 이렇게 x 값을 직접 업데이트해주면 global 안 써도 되겠어요! 이게 더 깔끔한 것 같아요!

슬기 선생님: 맞아요, 지혜 학생! 이렇게 파라미터와 return을 사용하는 것이 코드를 이해하고 관리하기에 더 좋은 방법일 때가 많답니다. 오늘은 global을 사용한 코드를 이해하는 데 집중하고, 다음에는 파라미터를 활용하는 방법도 더 연습해 볼게요!

3. 이중 for문으로 창문들 배열하기!

슬기 선생님: 자, 이제 이 window() 함수를 이중 for문으로 호출해서 창문들을 여러 줄, 여러 칸으로 그려볼 거예요.

```

# (앞부분 x = -125, y = -100 와 window() 함수 정의는 그대로 두고...)
for j in range(3): # 바깥쪽 for문: 창문 줄을 3줄 그릴 거예요. (j는 0, 1, 2)
    for i in range(4): # 안쪽 for문: 각 줄마다 창문을 4개씩 그릴 거예요. (i는 0, 1, 2, 3)
        window() # 창문 하나 그리기! (이때 window 안에서 x가 70씩 증가해요)

x = -125 # 한 줄 다 그렸으면, 다음 줄은 다시 왼쪽(-125)부터 시작해야죠! x 위치 초기화!
y = y + 80 # 다음 줄은 현재 y 위치보다 80만큼 위로 올라가서 그려요!

```

코딩이: 와! 이중 for문이다! 바깥쪽 j 루프가 3번 도니까 창문이 3줄 그려지고, 안쪽 i 루프가 4번 도니까 각 줄마다 창문이 4개씩 그려지는 거네요! window() 함수가 불릴 때마다 함수 안에서 x가 70씩 늘어나니까 옆으로 창문이 쭉 그려지고... 한 줄(안쪽 i 루프)이 다 끝나면 x = -125로 다시 왼쪽으로 돌아가고, y = y + 80으로 한 칸 위로 올라가서 다음 줄 창문을 그리는 거군요!

슬기 선생님: 정확히 이해했어요! 바깥쪽 j 루프는 창문의 **세로 줄(행)**을 바꾸는 역할을 하고, 안쪽 i 루프는 각 줄에서 창문의 **가로 칸(열)**을 따라 그리게 만들죠. window() 함수 안에서 x가 바뀌면서 가로로 창문이 그려지고, 안쪽 루프가 끝나면 x를 초기화하고 y를 바꿔서 다음 줄로 넘어가는 거예요. 이렇게 해서 건물 몸체에 창문들이 격자무늬로 예쁘게 배열되는 거랍니다!

(paul 거북이가 돛과 여러 층, 그리고 규칙적으로 배열된 창문들을 가진 국회의사당 건물을 완성한다)

지혜: 우와! 정말 멋진 국회의사당이 완성됐어요! global 변수를 쓰는 이유도 알겠고, 그걸 안 쓰고 함수에 정보를 주고받는 방법도 있다는 걸 알게 됐어요! for문으로 x, y 위치를 바꿔가면서 그림을 그리는 것도 정말 신기하고요!

슬기 선생님: 아주 잘했어요, 여러분! 오늘 우리는 꽤 복잡해 보이는 그림도 함수와 반복문, 그리고 global 변수(또는 파라미터 전달 방식)를 잘 활용하면 체계적으로 그릴 수 있다는 것을 배웠어요. 특히 함수 안팎에서 변수가 어떻게 작동하는지 이해하는 것은 아주 중요하답니다!

코딩이 & 지혜: 네! 선생님! 오늘 정말 많은 것을 배우고 멋진 그림도 완성해서 너무 뿌듯해요!