

파이썬 거북이, paul 건축가의 멋진 집짓기 프로젝트! 🏠🌳☀️🌧️🐢

등장인물:

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

슬기 선생님: 여러분, 안녕하세요! 오늘은 우리 paul 거북이가 아주 멋진 집짓기 프로젝트를 시작한다고 해요! 파란 하늘과 노란 태양 아래, 예쁜 구름도 둥실 떠 있고, 아늑한 집에는 굴뚝과 지붕, 창문과 현관문까지 모두 갖춰져 있습니다. (선생님이 완성된 집 그림을 보여준다) 이 모든 것을 우리가 배운 파이썬 명령과 함수로 차근차근 만들어 볼 거예요!

코딩이: 와! 진짜 멋진 집이에요! 저 푸른 초원 위에 그림 같은 집을 짓는 거네요! paul 거북이, 오늘 할 일이 엄청 많겠는데요?

슬기 선생님: (웃음) 맞아요! 하지만 paul은 아주 유능한 건축가니까 걱정 없어요. 자, 먼저 작업 환경부터 만들어 볼까요?

1. 작업 환경 설정: 스크린과 거북이 준비!

```
clearscreen()
```

```
# 스크린 생성, 배경 지정
```

```
screen = Screen()
```

```
screen_width = 800 # 화면 넓이도 변수로!
```

```
screen_height = 600 # 화면 높이도 변수로!
```

```
screen.setup(screen_width, screen_height) # 변수를 사용해 화면 크기 설정!
```

```
screen.bgcolor("green") # 넓은 초록색 마당!
```

```
# 거북이, 속력은 15로
```

```
paul = Turtle()
```

```
paul.speed(15) # paul 거북이, 오늘은 조금 더 빠르게 움직여요!
```

지혜: 네! screen 객체를 만들어서 넓이랑 높이를 변수로 정하고, 배경색은 초록색으로 했어요. paul 거북이도 준비됐고, 속도도 꽤 빠르게 설정했네요!

슬기 선생님: 훌륭해요! 이제 하늘부터 그려볼까요?

2. 파란 하늘과 노란 태양 그리기

```
# 하늘
```

```
paul.penup()
```

```
# 하늘은 화면 왼쪽 아래에서 조금 위(-screen_height/6)부터 시작해서 위로 짝 차게 그려요.
```

```
paul.goto(-screen_width/2, -screen_height/6)
```

```

paul.pendown()
paul.color("deep sky blue") # 아주 맑은 하늘색!
paul.begin_fill()
for i in range(2): # 사각형 그리기
    paul.forward(screen_width) # 하늘의 넓이는 화면 전체 넓이!
    paul.left(90)
    paul.forward(screen_height * 2/3) # 하늘의 높이는 화면 높이의 2/3 정도!
    paul.left(90)
paul.end_fill()

```

태양

```

paul.penup()
paul.goto(-320, 225) # 하늘 왼쪽 위에 태양 위치 잡기!
paul.pendown()
paul.color("yellow")
paul.begin_fill()
paul.circle(30) # 반지름 30인 동그란 태양!
paul.end_fill()

```

코딩이: 와! goto()로 위치를 잡고 커다란 하늘색 사각형을 그려서 하늘을 만들었어요! screen_width랑 screen_height 변수를 쓰니까 하늘 크기도 화면에 맞춰서 딱 좋네요! 그 위에 노란색 동그라미로 태양도 반짝!

슬기 선생님: 그렇죠? 이제 하늘에 몽실몽실 구름을 그려 넣을 건데, 구름은 모양이 비슷하니까 함수로 만들면 좋겠죠?

3. 뭉게뭉게 cloud(x, y) 함수: 원하는 곳에 구름 소환!

구름 그리기. 반지름 25인 원 4개를 겹쳐서...

```

def cloud(x_pos, y_pos): # 구름을 그릴 위치 정보 (x_pos, y_pos)가 필요해!
    paul.penup()          # 일단 펜 들고
    paul.goto(x_pos, y_pos) # 알려진 x_pos, y_pos 위치로 이동!
    paul.pendown()
    paul.color("white")
    paul.begin_fill()
    paul.circle(25)        # 반지름 25짜리 동그라미 하나 (구름 조각)
    paul.end_fill()

```

이제 cloud 함수를 불러서 구름을 여러 개 만들어 볼까요?

```

cloud(200, 200) # (200, 200) 위치에 구름 조각 하나!
cloud(220, 240) # (220, 240) 위치에 또 하나!
cloud(230, 215) # 이런 식으로 위치를 바꿔가며
cloud(180, 225) # 여러 번 호출하면 뭉게구름 완성!

```

지혜: cloud(x_pos, y_pos) 함수는 x_pos랑 y_pos라는 두 개의 “정보 쪽지(파라미터)”를 받네요! 이 정보 쪽지에 적힌 숫자들이 바로 구름을 그릴 x좌표랑 y좌표가 되는 거군요!

슬기 선생님: 맞아요! cloud 함수를 정의할 때 def cloud(x_pos, y_pos): 이렇게 “나는 두 개의 위치 정보가 필요해!”라고 약속했어요. 그래서 우리가 함수를 부를(호출할) 때, 예를 들어 cloud(200, 200) 이렇게 하면, x_pos에는 200이, y_pos에도 200이라는 실제 값이 전달되는 거예요. 그럼 paul 거북이는 (200, 200) 위치로 가서 하얀 동그라미를 하나 그리겠죠.

코딩이: 아하! 그럼 cloud(220, 240) 이렇게 부르면, 이번엔 x_pos에 220, y_pos에 240이 들어가서 다른 위치에도 동그라미를 그리는 거네요! 이렇게 함수에 다른 “정보(인자)”를 주면서 여러 번 부르니까 뭉게구름처럼 보여요! 만약 cloud() 이렇게 괄호 안에 아무것도 안 넣고 부르면 어떻게 돼요?

슬기 선생님: 아주 좋은 질문이에요! cloud 함수는 x_pos랑 y_pos 두 가지 정보를 꼭 받기로 약속했잖아요? 그런데 아무것도 안 주면 “어? 구름을 어디다 그려야 할지 위치 정보를 안 줬잖아! 일을 못 하겠어!” 하면서 파이썬이 오류를 낼 거예요. 함수를 부를 때는 함수가 필요로 하는 정보(인자)를 약속대로 잘 챙겨줘야 한답니다!

4. 집의 기본 구조: 몸통, 굴뚝, 지붕 그리기

(선생님은 집 몸통, 굴뚝, 지붕 그리는 코드를 보여주며, 각 부분이 goto로 위치를 잡고, color, begin_fill, for문(사각형/삼각형)을 사용해 그려지는 과정을 간략히 설명한다. 이 부분은 이전 학습 내용의 복습에 가깝다.)

집 몸체

```
paul.penup() # 펜 올리고 위치 이동 후 펜 내리고 그리기!
paul.goto(-100, -screen_height/6) # 하늘 바로 아래쪽부터 집 몸체 시작
paul.pendown()
paul.color("brown", "orange") # 선은 갈색, 채우기는 주황색
paul.pensize(3)
paul.begin_fill()
for i in range(4): # 정사각형 몸체
    paul.forward(170)
    paul.left(90)
paul.end_fill()
```

굴뚝

```
paul.penup()
paul.goto(20, 130) # 집 몸체 위쪽 오른쪽에 굴뚝 위치
paul.pendown()
paul.color("brown", "lime green") # 선은 갈색, 채우기는 연두색
paul.begin_fill()
for i in range(2): # 길쭉한 사각형 굴뚝
    paul.forward(20)
    paul.left(90)
    paul.forward(100)
```

```

    paul.left(90)
paul.end_fill()

# 지붕(정삼각형)
paul.penup()
paul.goto(-127, 70) # 집 몸체 꼭대기에서 살짝 조정된 위치에서 지붕 시작
paul.pendown()
paul.color("brown", "green") # 갈색 선에 초록색 지붕! (색상 코드 수정 제안)
paul.begin_fill()
for i in range(3): # 정삼각형은 세 번 반복
    paul.forward(224) # 지붕 한 변의 길이
    paul.left(120) # 정삼각형의 외각은 120도!
paul.end_fill()

```

지혜: 집 몸통이랑 굴뚝은 사각형이고, 지붕은 정삼각형이네요! for문이랑 각도 조절하는 거 이제 익숙해요! goto로 위치 잡는 것도 중요하고요.

슬기 선생님: 맞아요! 이제 집에서 가장 중요한 창문을 달아줄 차례예요. 창문도 모양이 같으니까 함수로 만들어볼게요.

5. 창문과 창틀 그리는 frame(x, y) 함수: 정확한 위치에 창문 설치!

```

# 창틀(50픽셀 정사각형)
# def frame(x_coord, y_coord): # 창틀을 그릴 위치 정보 (x_coord, y_coord)가 필요해요!
#     paul.penup() # (원래 코드에서는 t 거북이를 사용했는데, paul로 통일해서 설명할게요!)
#     paul.goto(x_coord, y_coord)
#     paul.pendown()
#     paul.color("black", "white") # 검은색 테두리에 하얀색 창문
#     paul.begin_fill()
#     for i in range(4): # 정사각형 창문 바깥틀
#         paul.forward(50)
#         paul.left(90)
#     paul.end_fill()

# # 창문 가운데 십자(+) 모양 그리기
# paul.penup()
# paul.goto(x_coord + 25, y_coord) # 창문 가로 중앙으로 이동
# paul.pendown()
# paul.color("black") # 창살은 검은색으로
# paul.left(90) # 위를 향하도록 방향 전환 후 선긋기
# paul.forward(50) # 창문 세로선

```

```
# paul.penup()
# paul.goto(x_coord, y_coord + 25) # 창문 세로 중앙으로 이동
# paul.pendown()
# paul.right(90) # 원래 방향으로 돌린 후 오른쪽을 향하도록 하고 선긋기
# paul.forward(50) # 창문 가로선
```

```
# # frame 함수 호출해서 창문 두 개 달기!
# frame(-80, 0) # (-80, 0) 위치에 창문 하나!
# frame(0, 0) # (0, 0) 위치에 또 창문 하나!
```

(선생님이 frame 함수 코드에서 t를 paul로 수정하고, 창살 그리는 부분의 color와 방향 전환 설명을 추가하며 안내한다.)

슬기 선생님: frame(x_coord, y_coord) 함수도 cloud 함수처럼 창틀을 그릴 위치 정보 x_coord와 y_coord를 “정보 쪽지(파라미터)”로 받아요. 이 정보대로 paul 거북이가 이동해서 하얀색으로 채워진 정사각형 창문을 그리고, 그 위에 검은색으로 십자 모양 창살까지 그려준답니다! frame(-80, 0) 이렇게 부르면, x_coord에는 -80이, y_coord에는 0이 전달돼서 그 위치에 창문이 생기는 거죠. frame(0,0)도 마찬가지고요!

코딩이: 함수 덕분에 똑같은 창문을 여러 개 그릴 때 코드가 엄청 짧아졌어요! frame 함수에 숫자만 바꿔서 넣어주면 되니까요!

6. 현관문과 손잡이: 디테일로 완성도 UP!

(선생님은 현관문과 손잡이 그리는 코드를 보여주며, 사각형과 작은 원으로 구성됨을 설명한다.)

```
# 현관문
paul.penup()
paul.goto(-40, -97) # 집 몸체 앞쪽 적절한 위치에 현관문 시작점
paul.pendown()
paul.color("black", "red") # 검은 테두리, 빨간 문
paul.begin_fill()
for i in range(2): # 직사각형 문
    paul.forward(50) # 문의 너비 (수정 제안: 높이가 더 길어야 하므로 forward 순서 변경 또는 값 조정)
    paul.right(90)
    paul.forward(80) # 문의 높이
    paul.right(90)
paul.end_fill()

#손잡이
paul.penup()
paul.goto(-25, -57) # 문 오른쪽에 손잡이 위치
paul.pendown()
paul.color("black")
```

```
paul.begin_fill()  
paul.circle(5) # 반지름 5의 작은 원 손잡이  
paul.end_fill()
```

```
paul.hideturtle() # 마지막엔 거북이 숨기기! 뽕!
```

지혜: 현관문은 빨간색 직사각형이고, 손잡이는 작은 검은색 동그라미네요! hideturtle()로 마지막에 거북이를 숨기니까 그림만 깔끔하게 보여서 좋아요!

슬기 선생님: 네, 이렇게 해서 paul 건축가의 멋진 집짓기 프로젝트가 완성되었어요! 오늘 우리는 특히 cloud() 함수와 frame() 함수처럼, **함수에 필요한 정보(위치 좌표 x, y)를 파라미터로 정확히 전달해주면** 거북이가 그 정보에 맞춰서 그림을 그려준다는 것을 확실히 배웠어요. 함수를 잘 활용하면 복잡한 그림도 훨씬 쉽고 체계적으로 그릴 수 있답니다!

코딩이 & 지혜: 네! 선생님! 오늘 정말 멋진 집을 함께 지어서 너무 재미있었어요! 함수에 정보를 전달하는 방법, 이제 확실히 알겠어요!