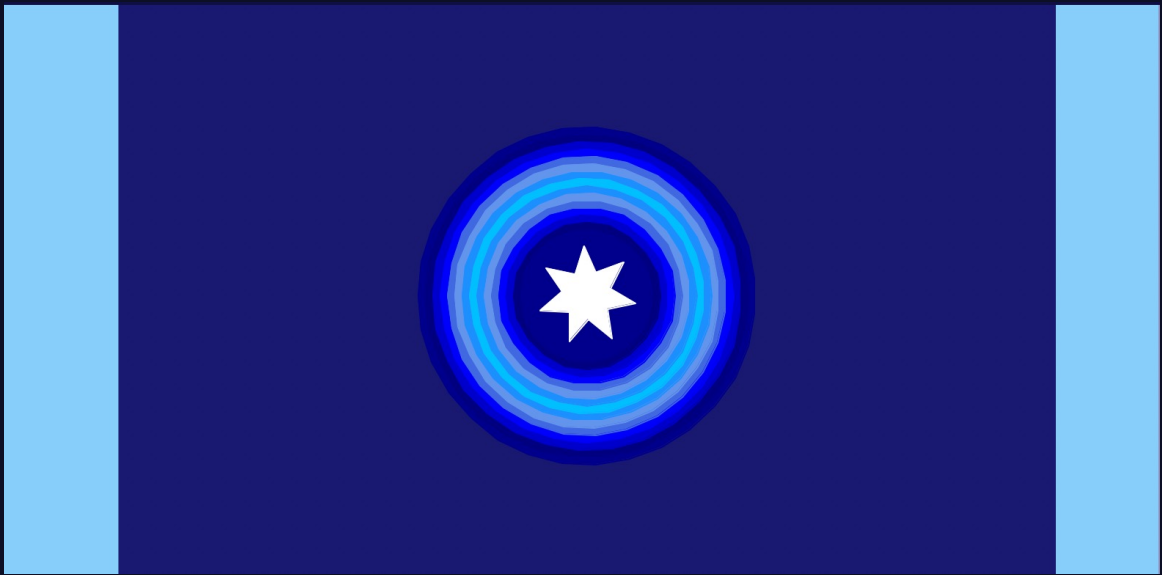


Python. Turtle Project



20244360 권민서

Term1. 기초 프로그래밍 03분반

목차

1. 나라별 국기 p.2 ~ p.12

(선택 계기, 문제점, 해결 방안)

2. 중요 코드. p.13 ~ p.17

3. 실행 화면 p.18 ~ p.19

4. 결론 및 후기 p.20

미국

- 미국 국기 선택 계기

아직 파이썬 turtle이 익숙하지 않은 상태에서 연습하면서 하기에 적합한 국기라고 생각했다.

- 문제점 및 해결방안

1) 국기의 비율을 규격에 맞게 서로 맞추는 것

원인 처음에 반복문을 사용할 생각을 못했기 때문에 코드도 복잡해지고 각 그림과의 비율과 위치도 맞지 않는 일이 일어났다.

해결방안 하나의 직사각형을 기준으로 크기와 비율을 전체적으로 다시 세팅하였다.
'for' 반복문을 이용하여 반복되는 문장을 함수로 표현하였다.

2) 계산기로 계산한 국기의 비율과 위치가 보여지는 것과 다른 점

원인 소수점까지 작성을 하지 않았다.
계산기를 사용하는 과정에서 판단의 오류가 있었다.
turtle이 그리기 시작한 지점으로 계산하지 않았다.
계산하지 못하는 각도와 삼각함수 값이 있었다.

해결방안 핸드폰에서 각도를 구할 수 있는 편집프로그램을 실행하였다.
공학 계산기를 이용하여 cot값과 sec값을 이용하여 길이를 구했다.
시작지점을 다시 보고 시작 지점을 변경 or 시작 지점에 맞게 코드를 변경하였다.

영국

- 영국 국기 선택 계기

다른 나라들 중에 영국 국기가 작게 포함되는 국기들이 많으니까 영국 국기를 그려서 활용할 생각이었다.

- 문제점 및 해결방안

1) 빨간 X자 대각선의 복잡성

원인 빨간 X자 대각선이 직선으로 연속해서 그릴 수 있는 그림이 아니었기에 일일이 코드를 작성해야 했다.

국기 끝에서 대각선이 끝날 때의 각도와 서로 간의 비율을 알기 힘들었다.

해결방안 대각선 간의 비율을 측정하고 그에 맞게 위치 설정까지 미리 하였다.
공학용 계산기로 삼각함수를 계산하여 길이를 구하고 비율에 맞게 조정했다.

2) 각도 설정의 오류

원인 소수점까지 작성해도 정확도의 한계 존재.
살짝만 각을 틀어지게 입력해도 국기의 틀을 벗어나게 되는 점.

해결방안 turtle 실행 후 계속해서 코드를 수정하며 값을 조정함.
코드를 깔끔하게 실행하기 위해

튀르키예

- 튀르키예 국기 선택 계기

달 모양 그림이 들어간 국가가 꽤 있었기에 함수로 묶어서 사용하고 싶어서 선택했다.

- 문제점 및 해결방안

- 1) 달 모양 비율 맞추기

원인 튀르키예의 달 모양 비율을 생각하지 못하고 임의로 설정한 것.

해결방안 달 모양에서 제일 두꺼운 중심부분의 길이를 비율에 맞게 계산한 후에 달 모양을 만들기 위한 원의 반지름을 설정함.

엔티가 바부다

- 엔티가 바부다 국기 선택 계기

삼각형을 함수 정의하고 반복해서 사용할 수 있었기 때문에 선택하였다.

- 문제점 및 해결방안

1) 해 모양의 각도의 오류

원인 180도 회전과 360도 회전을 자주 헷갈려 예상과 다른 값을 각도로 지정한 점.
알맞은 각도를 입력했음에도 소수점 자리로 넘어가는 값으로 인해 해의 뾰족한 부분에 빈공간이 발생하는 점.
for반복문을 사용하였기에 조금의 각도차이로도 시작과 끝이 연결되지 않는 점.

해결방안 실행 후 오류인 부분을 계산하여 정확한 값이 아니더라도 조금씩 조정하면서 알맞은 모양을 맞추었다.

2) 삼각형 크기의 오류

원인 삼각형이 총 세가지 색깔로 이루어져 있는데 각각의 비율을 잘못 설정하여 각 삼각형 간의 빈 공간 발생.

해결방안 실제 구한 값의 길이보다 살짝 크게 길이를 설정한 후 실행 코드를 순차적으로 실행 시킴으로써 빈 공간이 없고 각각의 비율과 크기는 맞도록 조정하였다.

한국

- 한국 국기 선택 계기

필수 포함 국가이기도 하였지만 우리나라이기 때문에 선택하였다.

- 문제점 및 해결방안

1) 건곤감리 4괘의 위치와 간격 설정

원인 처음에는 건곤감리 4괘에서 각각의 위치 설정을 할 수 있는 함수를 정의하고 그 안에 반복문을 사용할 생각으로 코드를 생각함.
건곤감리 4괘 위치 시작 지점 설정을 잘못했기 때문에 코드도 복잡해지고 각 그림과의 비율과 위치도 맞지 않는 일이 일어났다.

해결방안 하나의 직사각형을 기준으로 함수는 지정하고 나머지는 일일이 위치를 바꿔가며 작성하였다.
‘for’ 반복문을 이용하여 반복되는 사각형을 간단하게 작성하였다.

2) 태극 문양에서 작은 원의 색칠 오류

원인 코드를 다 작성한 후 실행해본 것. (미리 실행하면서 코드 작성을 했어야 한다고 생각)
작은 원으로 태극 문양을 표현하려 하였지만, 그 과정에서 원 그리는 시작 각도를 틀어 버림.

해결방안 원을 그리는 시작 각도에 맞게 반지름과 위치를 다시 재설정.
코드 수정 과정에서 수정 후 바로바로 실행 화면 확인으로 효율을 높이려 노력함.

그리스

- 그리스 국기 선택 계기

직사각형 그리는 함수가 정의되어 있기 때문에 이를 활용할 수 있는 국기를 찾던 중 그리스의 국기가 직사각형으로만 그릴 수 있어서 선택하였다.

- 문제점 및 해결방안

- 1) 십자가의 크기와 작은 정사각형 크기의 불일치

원인	분명히 정확한 값과 정사각형 변의 길이와 십자가의 길이를 같게 설정했음에도 십자가의 크기가 정사각형 크기보다 커지는 현상이 발생함.
----	---

해결방안	십자가의 크기와 정사각형 변의 길이를 같게 설정하지 않고 실행 후 둘의 크기가 일치하도록 미세하게 값을 조정하였다.
------	--

쿠바

- 쿠바 국기 선택 계기

별 모양과 직사각형 모양 삼각형 모양을 정의하는 함수를 작성해 놓았기 때문에
함수를 반복하여 사용 후 간단하게 코드를 작성 할 수 있다고 생각하여 선택하였다.

- 문제점 및 해결방안

- 1) 삼각형의 각도 계산 오류

원인	삼각형에서 오른쪽으로 각도를 전환할 때 생각 오류로 값을 지나치게 크게 입력함.
----	--

해결방안	각도를 재설정 후 실행하였다. 색칠된 부분에서 빈공간이 발생하는 오류가 생겨 변의 크기도 같이 조정하였다.
------	--

캐나다

- 캐나다 국기 선택 계기

개인적으로 국기가 굉장히 예쁘다고 생각하기 때문에 선택하였다.

- 문제점 및 해결방안

1) 코드의 복잡성

원인	처음에 절반을 기준으로 왼쪽 부분을 그리는 코드 작성 후 오른쪽은 데칼코마니처럼 실행 시켜 간단한 코드를 작성하려 하였지만, 그에 맞는 코드를 찾지 못했다.
----	---

해결방안	코드를 복사 붙여넣기 하여 't.right'를 't.left'로 바꾸는 등 앞에서 사용한 코드를 이용하여 일일이 코드를 작성해야 했다.
------	---

오스트레일리아

- 오스트레일리아 국기 선택 계기

전에 작성했던 영국 국기가 작게 들어가기 때문에 이를 이용할 목적으로 선택하였다.

- 문제점 및 해결방안

1) 영국 국기의 비율을 규격에 맞게 서로 맞추는 것

원인 처음에 작성된 영국 국기의 전체적인 크기를 한번에 줄이는 코드가 존재할 것이라 생각하였다. 하지만 그런 것은 존재하지 않았다.

해결방안 오스트레일리아 국기 크기의 $\frac{1}{4}$ 크기로 길이와 위치 값을 비율에 맞게 하나하나 조정하였다.

2) 7개의 꼭짓점을 가진 별

원인 함수로 한꺼번에 작성 가능하게 하고 싶었기에 길이와 위치를 따로 입력할 수 있는 함수로 작성 하였다. 이 과정에서 각도 오차로 인해 꼭짓점이 6개인 별이 완성되거나 별의 시작과 끝이 맞지 않는 등 다양한 오류가 발생하였다.

해결방안 코드 수정과 실행을 반복하며 각도를 조정해 나갔다.
반복 함수를 실행해야 했기에 그에 맞는 길이와 위치를 비율에 맞게 구하여 대입하였다.

북마케도니아

- 북마케도니아 국기 선택 계기

시도해보고 싶은 국기였다. 이 국기를 함수로 간단하게 표현해보고 싶었다.

- 문제점 및 해결방안

1) 작성된 코드를 대칭할 수 없다는 것 (캐나다 국기 코드 작성시와 같은 이유)

원인 간단하게 작성하고 싶은 마음에 오히려 더 복잡하게 코드를 작성하였다.

해결방안 함수로 정의하는 것보다 하나하나 쓰는게 더 간단할 수도 있다는 점을 깨닫고 국기의 $\frac{1}{4}$ 을 그리는 코드를 작성한 후 't.left' 를 't.right' 로 바꾸는 등 변환하여 재사용 하였다.

알제리

- 알제리 국기 선택 계기

터키에서 달 모양을 사용하였기에 이를 이용할 수 있다고 생각하여 선택하였다.

- 문제점 및 해결방안

- 1) 달 모양 안에 뒤의 나누어진 국기를 표현해야 한다는 점

원인

애초에 달 모양을 제작할 때 원 위에 배경색의 작은 원을 덧그리며 만들었기 때문에 뒤의 나누어진 부분을 표현할 수 없었다.

해결방안

많은 방법을 시도하던 중 활꼴을 표현하여 덧그리는 것이 제일 간단하다 판단하였다. 따라서 작은 원의 호 크기에 맞는 활꼴 제작 코드를 작성 후 실행하여 표현하였다.



중요 코드

‘If 조건문 사용’

- ✓ if문을 사용하여 반복문을 실행할 때 짝수번째와 홀수번째에 실행되는 코드를 달리 하였다.
- ✓ ‘start_y - i * stripe_height’ 를 시작 y좌표로 함으로써 반복문이 진행될수록 y좌표값이 점점 줄어들도록 설정하였다.

```
# Draw stripes
for i in range(13):
    if i % 2 == 0:
        color = "#B22234" # Red
    else:
        color = "white"
    draw_rectangle(color, start_x, start_y - i * stripe_height, width: 494, stripe_height)

# Draw union
draw_rectangle( color: "#3C3B6E", start_x, start_y, width: 197.6, height: 140.01)

# Draw stars
t.color("white")
for row in range(9):
    for col in range(6 if row % 2 == 0 else 5):
        x = start_x + 14 + col * star_space + (0 if row % 2 == 0 else star_space / 2)
        y = start_y - 14 - row * star_row_space
        draw_star(x, y)
```

중요 코드

'직사각형 그리기' 함수로 정의

- ✓ def를 활용하여 draw_rectangle 함수를 정의하여 사용하였다.

```
def draw_white_cross():
    # 중앙의 하얀 십자가
    draw_rectangle("#FFFFFF", -width/2, -50, width, 100)
    draw_rectangle("#FFFFFF", -50, -height/2, 100, height)

1 usage
def draw_red_cross():
    # 중앙의 빨간 십자가
    draw_rectangle("#CF142B", -width/2, -30, width, 60)
    draw_rectangle("#CF142B", -30, -height/2, 60, height)

1 usage
def draw_bg():
    # 파란 배경
    draw_rectangle("#00247D", -width/2, -height/2, width, height)
```

중요 코드

'별 그리기' 함수로 정의

- ✓ def를 활용하여 draw_star 함수를 정의하여 사용하였다.

```
def draw_star(start_x, start_y, length):  
    t.penup()  
    t.goto(start_x, start_y)  
    t.pendown()  
    t.begin_fill()  
    t.setheading(72)  
  
    t.color("#FFFFFF")  
  
    for _ in range(7):  
        t.forward(length)  
        t.right(138)  
        t.forward(length)  
        t.left(87)  
  
    t.end_fill()
```

```
def draw_star():  
    t.color("#FFFFFF")  
    t.penup()  
    t.goto(x: 180, -10)  
    t.pendown()  
    t.begin_fill()  
    t.setheading(254.5)  
    for _ in range(5):  
        t.forward(8)  
        t.left(144)  
        t.forward(8)  
        t.right(72)  
    t.end_fill()
```


중요 코드

'삼각형 그리기' 함수로 정의

- ✓ def를 활용하여 draw_triangle 함수를 정의하여 사용하였다.

```
def draw_triangle():  
    t.penup()  
    t.goto(-width/2, height/2)  
    t.setheading(-30)  
  
    t.color("#CF142B")  
  
    t.begin_fill()  
    t.forward(300)  
    t.right(120)  
    t.forward(300)  
    t.right(120)  
    t.forward(300)  
    t.end_fill()  
    t.setheading(0)
```

중요 코드

'for 반복문' 사용

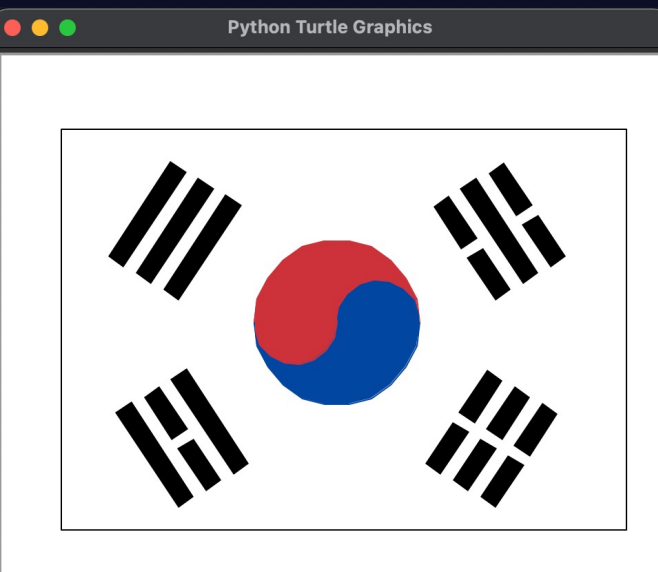
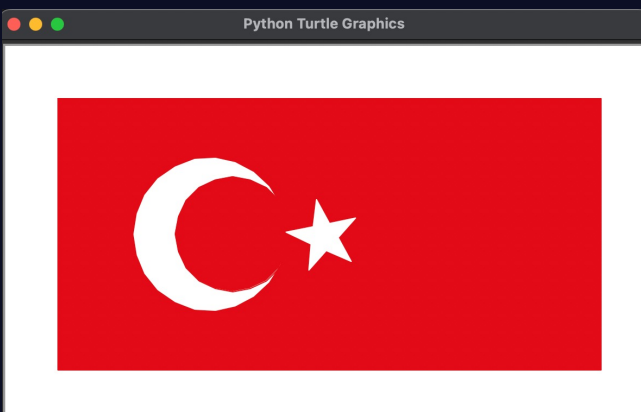
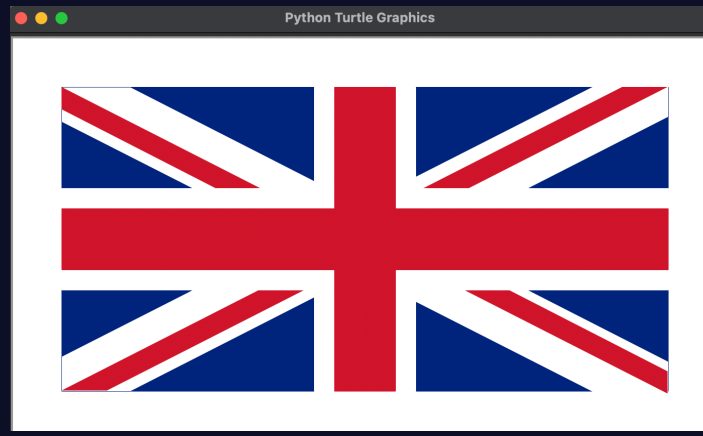
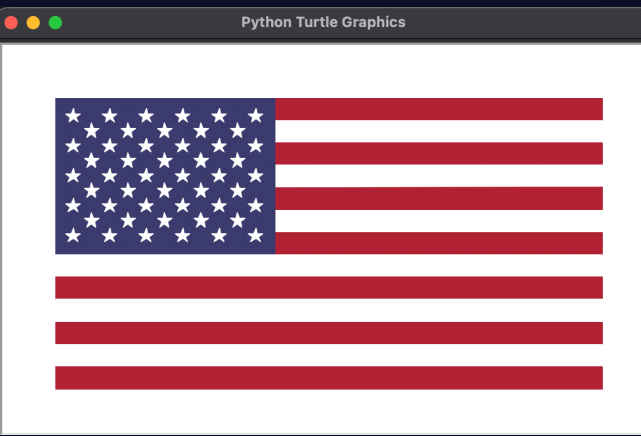
- ✓ for 반복문을 활용하여 반복되는 코드를 보다 간결하게 작성하였다.

```
for _ in range(2):  
    t.forward(width)  
    t.left(90)  
    t.forward(height)  
    t.left(90)  
t.end_fill()
```

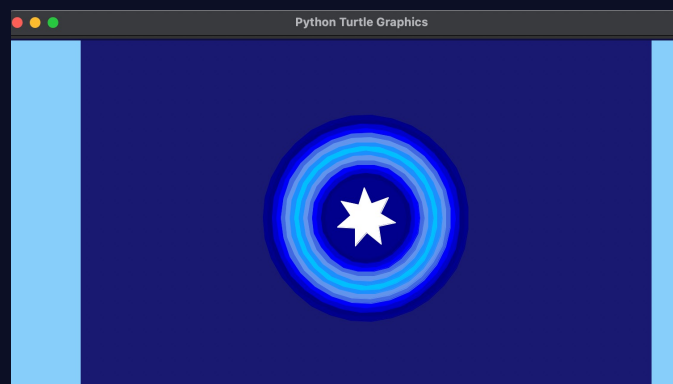
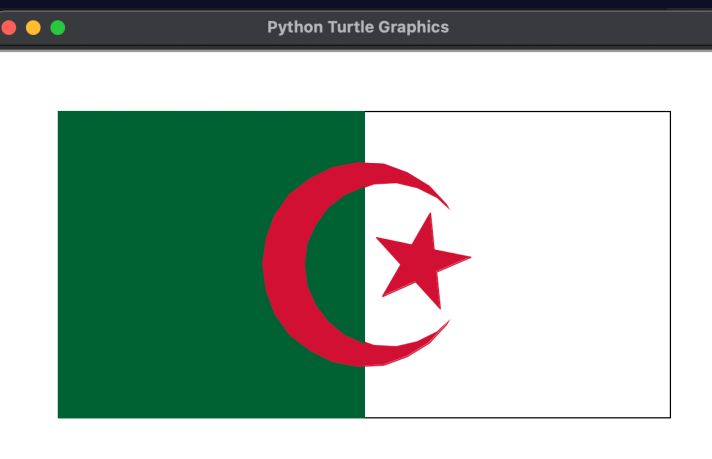
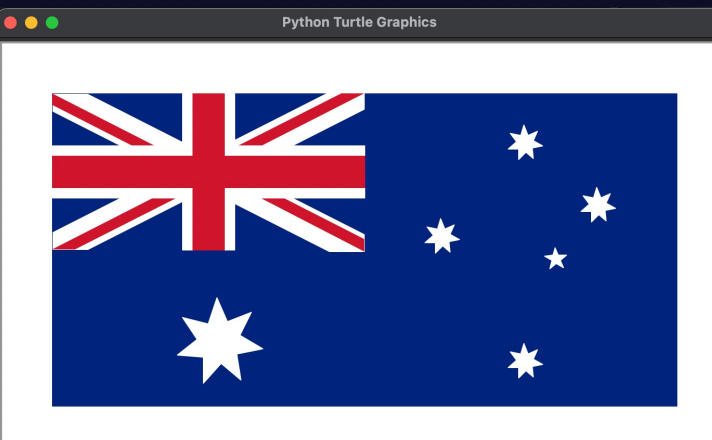
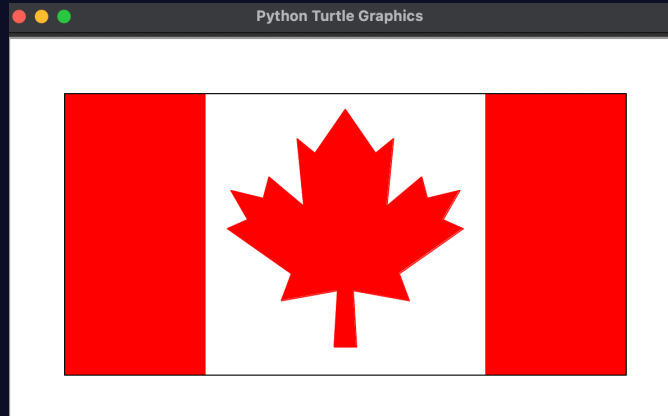
```
for _ in range(5):  
    t.forward(30)  
    t.left(144)  
    t.forward(30)  
    t.right(72)  
t.end_fill()
```

```
for row in range(9):  
    for col in range(6 if row % 2 == 0 else 5):  
        x = start_x + 14 + col * star_space + (0 if row % 2 == 0 else star_space / 2)  
        y = start_y - 14 - row * star_row_space  
        draw_star(x, y)
```

실행 화면



실행 화면



결론 및 후기

작성하고자 하는 함수가 있어도 마음대로 되지는 않는다.

- 함수로 정의하여 간단하게 만들하고자 하여도 그게 오히려 더 힘들 수도 있다는 것을 알게 되었다.
- 열심히 코드를 짜고 실행하여도 예상과 다르게 실행되는 코드들을 보고 무엇이 잘못되었는지 찾아보며 코드 하나하나의 의미를 다시 한번 생각할 수 있었다.
- 반복문과 조건문을 효율성있게 사용하겠다는 의도를 가지고 국기를 선택하였지만 생각대로 되지 않아서 결국 새롭게 코드를 짜고 있는 나 자신을 보며 파이썬에 대한 지식의 부족함을 깨닫게 되었다

Turtle에 대해 더 많이 배우고 알게 되었다.

- Turtle을 이렇게까지 오랫동안 붙잡고 있는 것이 처음인데, 하면서 생각대로 되지 않아 속상했기도 했고 시간은 흐르는데 완성된 것은 없는 나의 결과물에 실망도 했지만 이번 프로젝트를 기회로 삼고 더 배울 수 있어서 유익했다.
- 모르는 코드도 많고 표현하고자 하는 그림을 어떻게 구현하는지 몰라서 힘들었지만 이번 프로젝트를 통해 책도 찾아보고 인터넷도 찾아보고 동영상 강의도 찾아보며 전보다 확실히 turtle에 대해 더 많이 알게 됐다는 것을 느꼈다.

파이썬 터틀에서 내가 무엇이 부족한지 알게 되었다.

- Turtle을 사용하는 프로젝트를 진행하며 부족한 부분들을 직접적으로 느낄 수 있었다.
- Turtle 프로젝트를 진행하며 점점 재미를 느끼는 나 자신을 보게 되었다. 물론 당연히 처음 프로젝트를 시작할 때에는 막막하고 코드 하나 짜는 것도 힘들었는데 점점 배워가며 더 욕심이 생겼고 시간 가는 줄 모르고 코드를 붙잡고 있는 나 자신을 발견하게 되는 시간이었다.
- 힘들지 않았다면 거짓말이지만 이번 프로젝트를 통해 많이 배우고 재미도 느낀 사람으로서 앞으로 이런 프로젝트 기회가 더 많이 있었으면 좋겠다고 생각했다.