*HW4 과제 – 4/ 27*

**주제 :**

**바둑판과 다양한 선 출력하기**

**이름 : 김민석**

**학번 : 20195125**

**메일 : planetside2@naver.com**

**소속 : 소프트웨어융합**

**과목 : 컴퓨터사고와 문제해결**

**목차**

1. **첫번째 코드**
2. **실행화면 사진**
3. **내용 설명**
4. **두번째 코드**
5. **실행화면 사진**
6. **내용 설명**
7. **첫 번째 코드**

import turtle

line = turtle.Turtle()

for i in range(0, 151, 15):

line.penup()

line.goto(-100, 100 - i)

line.pendown()

line.goto(50, 100 - i)

for j in range(0, 151, 15):

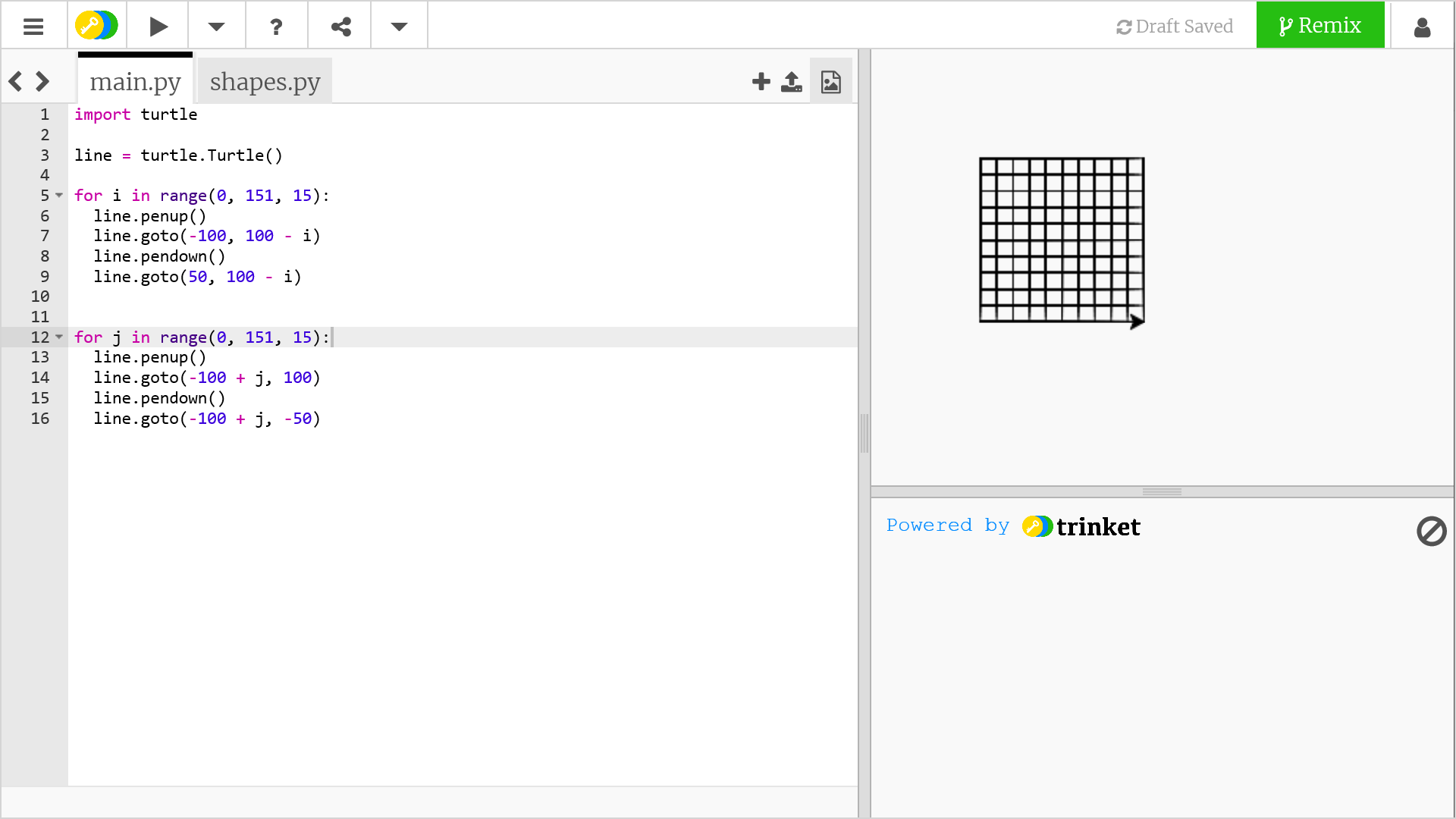
line.penup()

line.goto(-100 + j, 100)

line.pendown()

line.goto(-100 + j, -50)

1. **실행화면 사진**



1. **구현내용 설명**
2. **import turtle** 로 그림을 그릴 **turtle** 상자를 가져온다.
3. 선을 이용해 그릴 것이므로 **line = turtle.Turtle()** 로 **line** 에 **Turtle** 이라는 도구를 불러온다.
4. **10X10 바둑판**을 만들려면, 가로와 세로선이 둘 다 **11개씩** 필요하므로 **for i(j) in range(0, 151, 15)** 로 정사각형 바둑판이 그려질 수 있도록 반복시킨다.
5. **for 문 안에 line.goto(x좌표, y좌표)** 를 넣어서 **가로와 세로**를 그리게 하는데, 이때 **for문 안**에서 **첫** **goto** 함수에서 이동하는 선이 보이므로 안보이게 **line.penup()** 을 이용하여 가로선과 세로선만 보이게 해준다. 그리고나서 다음부터 다시 **line.pendown()**로 선이 제대로 보일 수 있게 해준다.
6. **가로**는 x 가 **-100 에서 50 까지** 이동할 수 있도록, 그리고 **세로**는 y 가 **100에서 -50 까지** 이동할 수 있도록 **goto**함수를 짜서 바둑판을 그릴 수 있게 한다.
7. **두 번째 코드**
8. **Main.py**

import turtle

import Dot

line = turtle.Turtle()

def Python(m\_length, m\_cnt = 10):

line.forward(m\_length)

Dot.MyGoto(line, 0, -20)

Dot.dot(line, m\_length, m\_cnt)

Dot.MyGoto(line, 0, -40)

Dot.dot\_line(line, m\_length, m\_cnt)

Dot.MyGoto(line, 0, -60)

Dot.chain\_line(line, m\_length)

Python(150)

1. **Dot.py**

def MyGoto(line, x\_xx, y\_yy):

line.penup()

line.goto(x\_xx, y\_yy)

line.pendown()

def dot(line, x\_length, x\_cnt):

p\_cnt = int(x\_length/x\_cnt)

for i in range(p\_cnt):

line.dot(3)

line.penup()

line.forward(x\_cnt)

line.pendown()

def dot\_line(line, x\_length, x\_cnt):

p\_cnt = int(x\_length/x\_cnt)

for i in range(p\_cnt):

line.forward(6)

line.penup()

line.forward(4)

line.pendown()

def chain\_line(line, x\_length):

p\_cnt = int(x\_length/30)

for i in range(p\_cnt):

line.pendown()

for j in range(3):

line.forward(x\_length/p\_cnt/6)

line.penup()

line.forward(x\_length/p\_cnt/6)

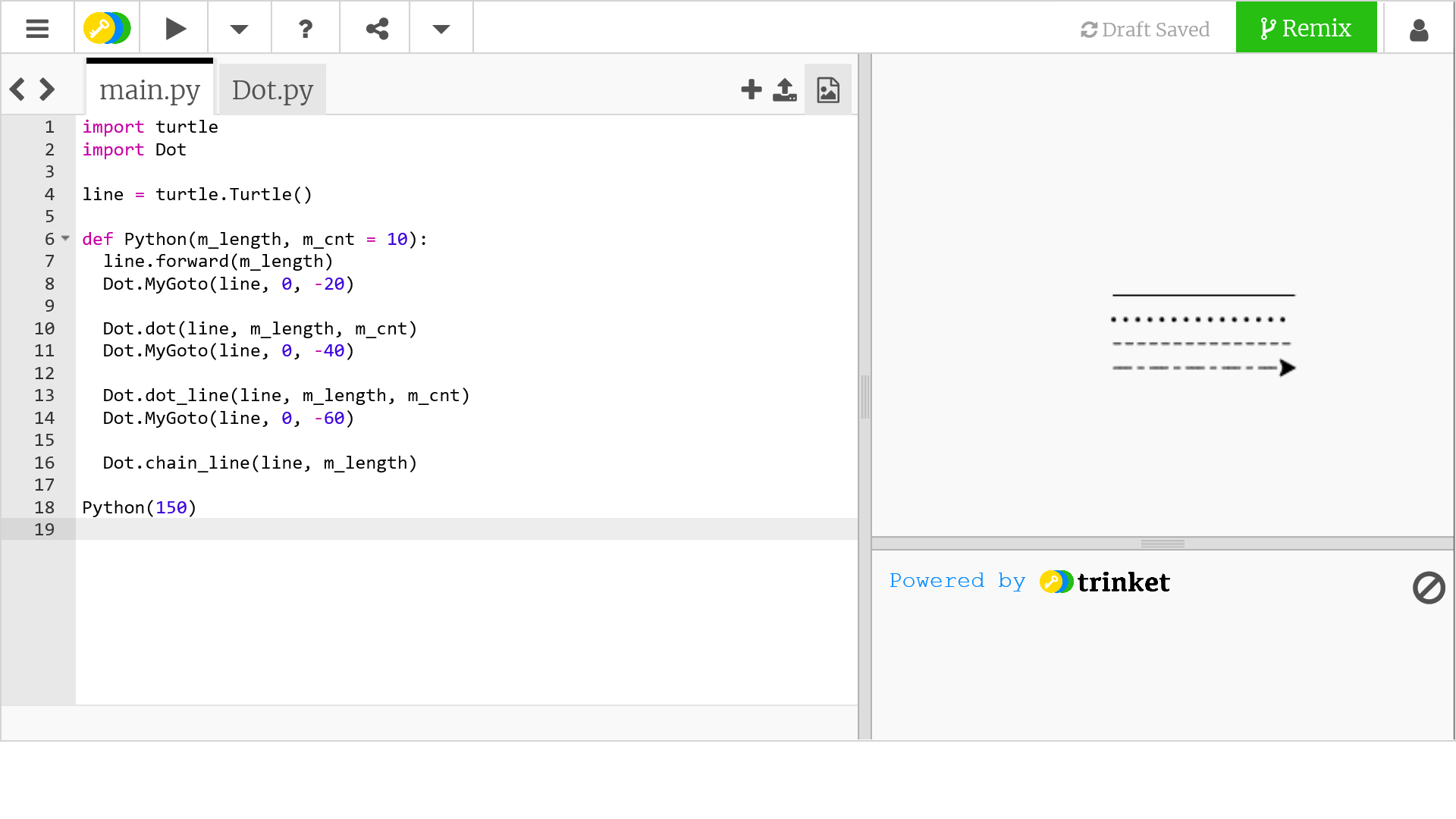
line.pendown()

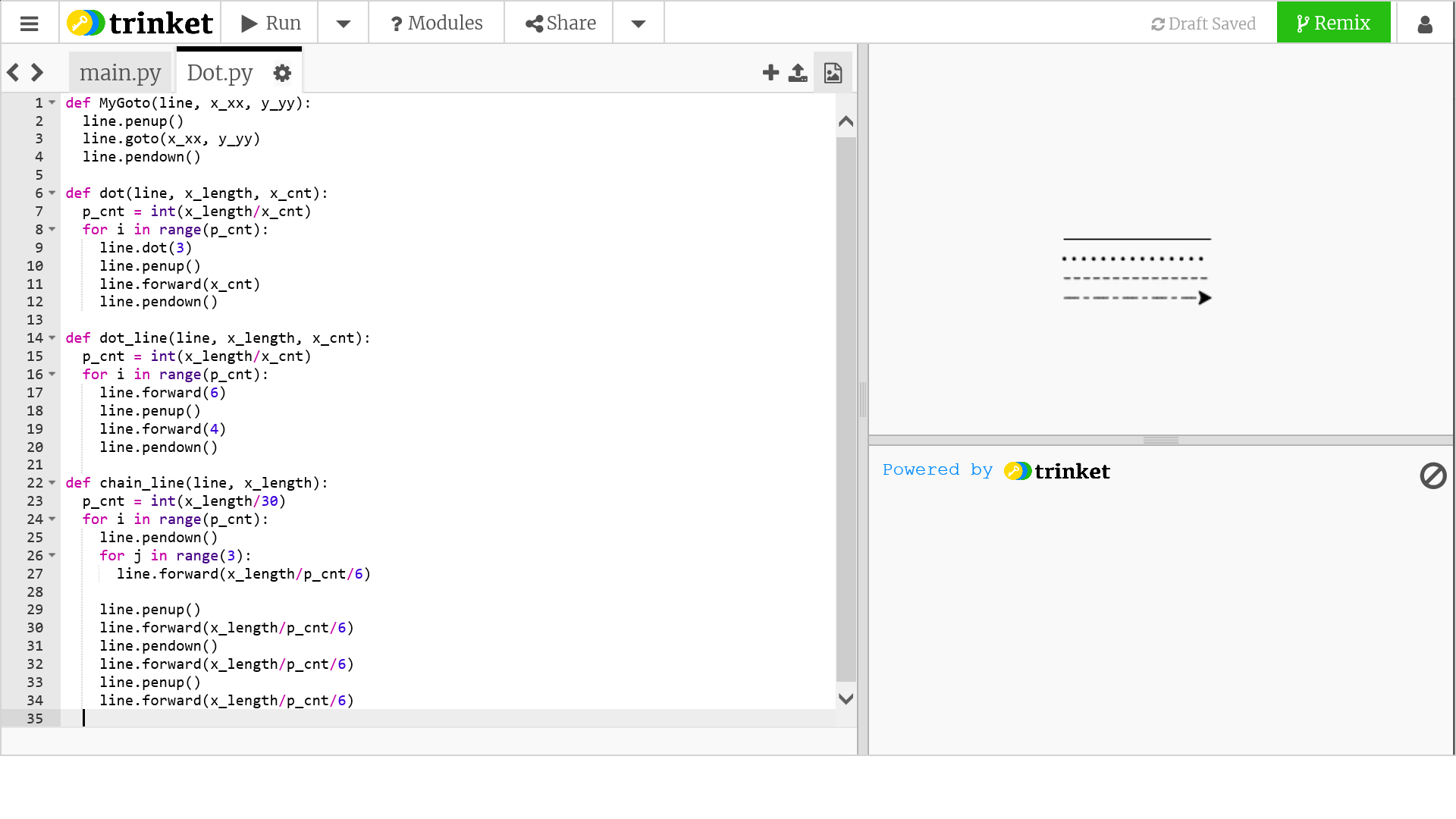
line.forward(x\_length/p\_cnt/6)

line.penup()

line.forward(x\_length/p\_cnt/6)

1. **실행화면 사진**





1. **구현내용 설명**

1. **import turtle**로 turtle 상자를 불러온다.

2. **line**을 선언해서 Turtle도구를 꺼내고, **메인 함수(m)**의 전체적인 길이와 점 개수를

**def Python(m\_length, m\_cnt = 10):**

**~ Python(150)**로 잡는다.

3. 이제 맨 처음 실선은 **line.forward(m\_length)**로 그린 후 **Dot.py를 생성해 앞으로 그릴 선들을 분리한다.**

4. 선을 간격이 생기게 그려야 하는데 **Dot.py과 main.py에서 변수들을 서로 인식**해야 하므로 Dot.py에 함수를 짤 때

**def (함수 이름)(line, ~)로 변수 line을 선언하고 시작한다.**

5. **Dot.py**에서 **def MyGoto(line, x\_xx, y\_yy):** 라는 **MyGoto 함수**를 만들어 그 안에 다음 줄로 갈 수 있게 **goto 함수를 짠다.**

6. **짠 MyGoto함수**를 **main.py에서 import Dot**을 **사용**해서 **Dot.py에서 짠 함수들을 쓸 수 있게 한다.**

7. **Dot.MyGoto(line, ~)**로 **Dot.py** 에 있는 함수 **MyGoto**를 **line**이라는 변수를 이용해서 사용한다.

8. 이와 같은 방법으로 **Dot.py**에 **dot(점), dot\_line(점선), chain\_line(쇄선)**에 대한 함수들을 짠 후 Dot.(함수 이름)((변수), ~)로 제대로 선들이 그려질 수 있도록 만든다.