*HW3 과제 – 3/30*

**주제 :**

**조건에 맞는 명령어 작성하기**

**이름 : 김민석**

**학번 : 20195125**

**메일 : planetside2@naver.com**

**소속 : 소프트웨어융합**

**과목 : 컴퓨터사고와 문제해결**

**목차**

1. **첫번째 조건, 코드**
2. **실행화면 사진**
3. **내용 설명**
4. **두번째 조건, 코드**
5. **실행화면 사진**
6. **내용 설명**
7. **세번째 조건, 코드**
8. **실행화면 사진**
9. **내용 설명**

**첫번째 조건 : 다각형 선을 점으로 표현하는 함수 (DotPolygon) 를 만드세요.**

1. **코드 :**

**import turtle**

**work1 = turtle.Turtle()**

**def DotPolygon(side, length):**

**for w1 in range(side):**

**for i in range(length):**

**work1.penup()**

**work1.forward(2)**

**work1.pendown()**

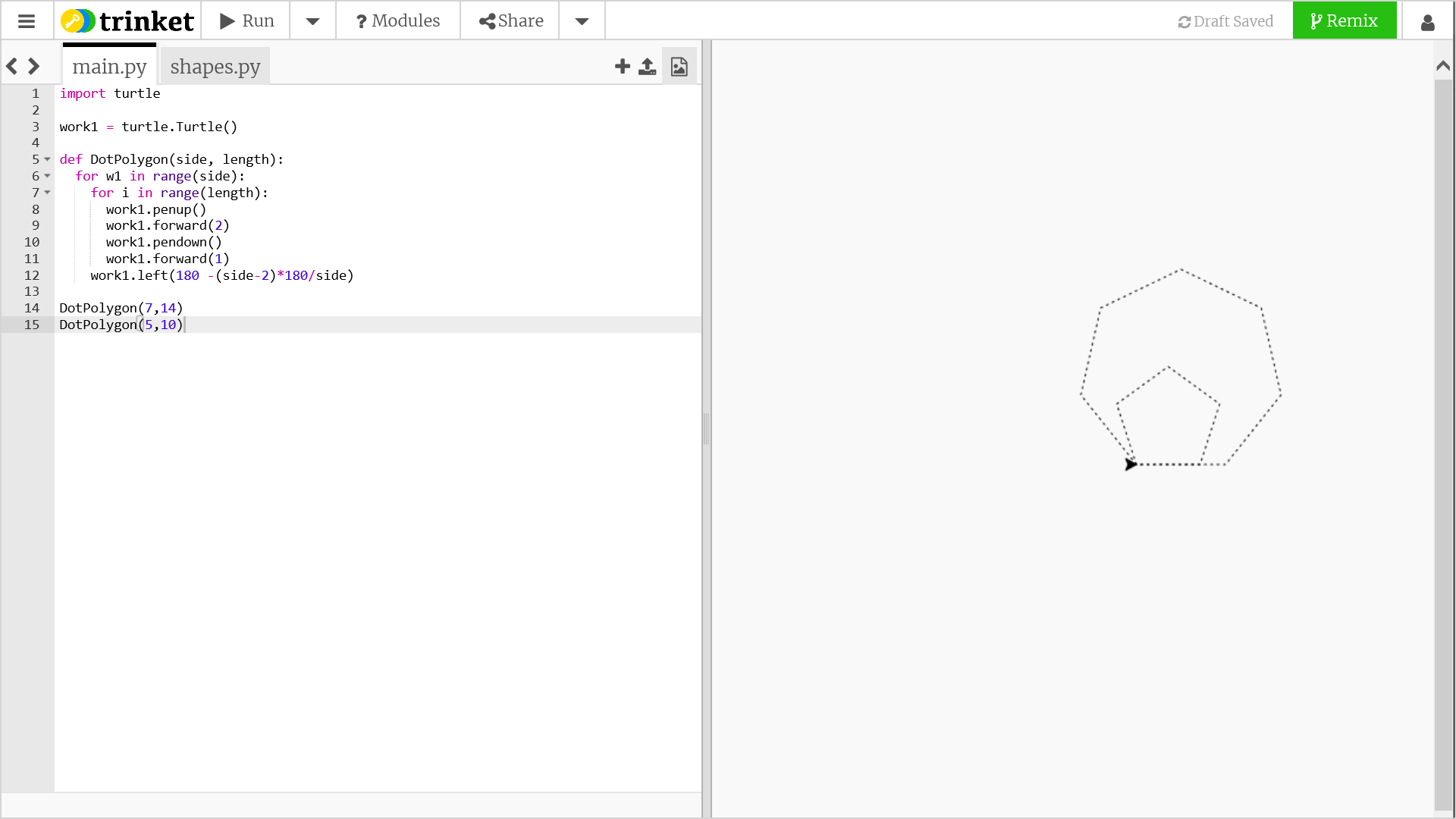
**work1.forward(1)**

**work1.left(180 -(side-2)\*180/side)**

**DotPolygon(7,14)**

**DotPolygon(5,10)**

**2.** 실행화면 사진 :



**3. 구현내용 설명**

1**. import turtle**로 **turtle** 도구를 가져오고, **work1 = turtle.Turtle()**로 **turtle**를 생성한다.

2. **def DotPolydon(side, length) ~ DotPolygon(7,14) DotPolygon(5,10)**으로 **side**와 **length** 를 각각 지정한다.

3. **for w1 in range(side)**로 정의된 변 수**(side)** 에 따라 도형을 그린다.

4. 도형을 그리는 동안 **work1.penup , work1.pendown** 으로 점선을 그릴 수 있도록 지정해주고 이것들을 **for i in range(length)** 로 **0부터** **def** 에 의해 정의된**length** 까지 점선을 그릴 수 있도록 해준다.

5.점선을 그릴 수 있게 한 후 한 변이 그려지면 도형의 각도에 맞게 꺾은 후 그릴 수 있도록 **work1.left(180 – (side-2)\*180/side)** 로 **각 변의 수(side)**에 맞게 점선으로 된 **DotPolygon**을 그린다.

**두번째 조건 : Turtle 의 circle 과 같은 함 수 MyCircle() 을 완성하세요.**

**- 펜의 색은 파란색을 사용하세요.**

**- 반지름 값을 입력 값으로 사용할 것**

1. **코드 :**

**import turtle**

**import math**

**aaa = turtle.Turtle()**

**aaa.pencolor('red')**

**aaa.circle(100, 360)**

**circle = turtle.Turtle()**

**circle.pencolor('blue')**

**r = int(input("원하시는 반지름의 길이를 입력해주세요."))**

**def MyCircle(x\_radius, x\_angle, pi = 3.1416):**

**length = 2\*pi\*x\_radius/x\_angle**

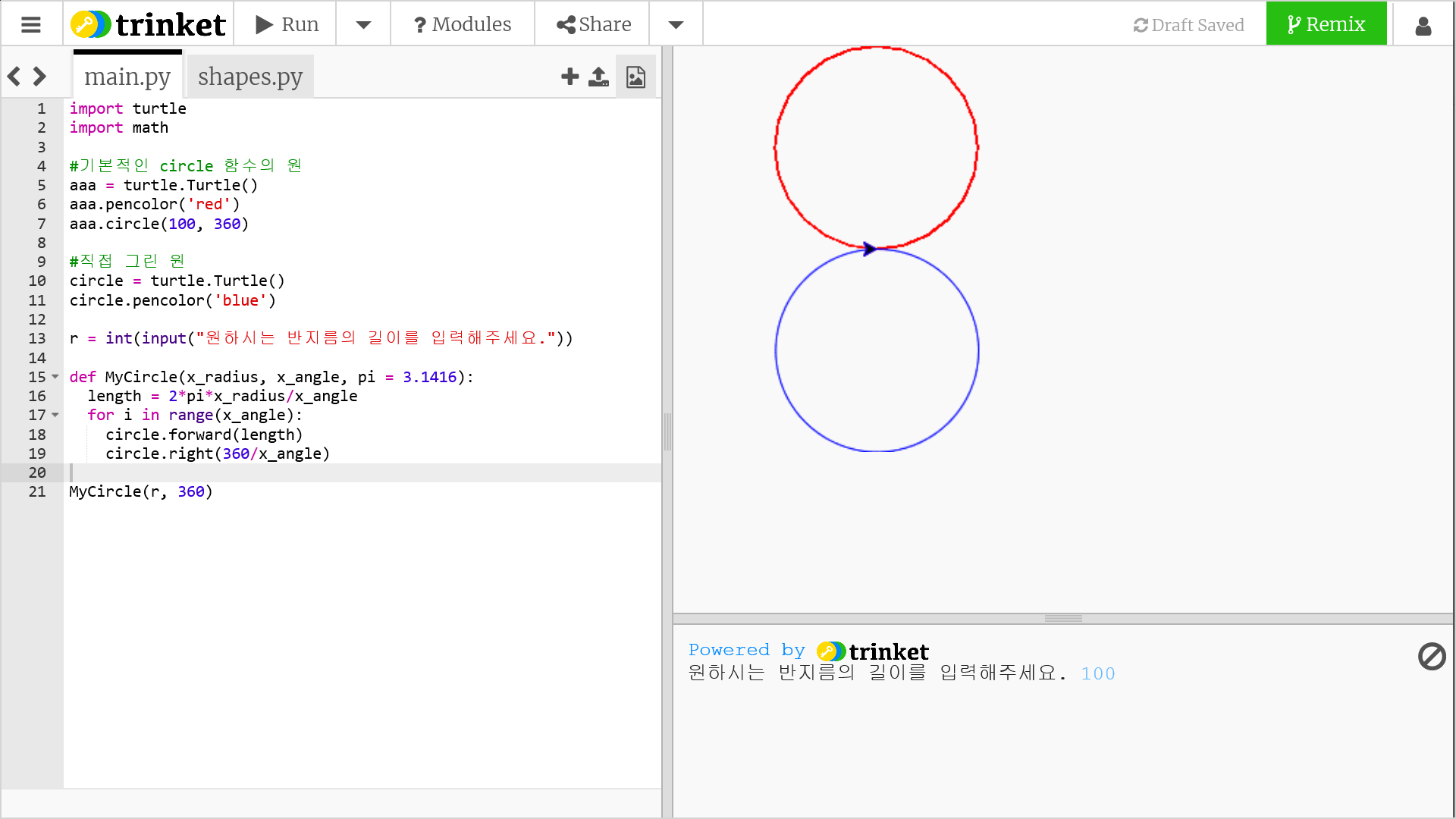
**for i in range(x\_angle):**

**circle.forward(length)**

**circle.right(360/x\_angle)**

**MyCircle(r, 360)**

1. **실행화면 사진 :**



1. **구현내용 설명**

1. 먼저 **turtle 과 math** 도구를 불러옵니다.

2. 그리고 나서 **aaa = turtle.Turtle()**로 **turtle**를 생성하고, **aaa.pencolor(‘red’)** 로 색을 바꾸고 **math 도구** 안에 있는 **aaa.circle(반지름, 각도)**를 이용해서 **빨간색 원 aaa** 를 그려줍니다.

3. **이번엔 .circle** **의 도움을 받지 않은 원**을 그리기 위해 **circle = turtle.Turtle()**로 **turtle을 생성**합니다.

4. **circle.pencolor(‘blue’)**로 새로 그릴 원의 색깔을 파랑색으로 하고, 사용자가 입력한 반지름 r **을 int(input(“~ “))**로 받습니다.

5. **def MyCircle(x\_radius, x\_angle, pi = 3.1416) ~MyCircle(r, 360)**으로 **x\_radius** 는 **입력받은 반지름, x\_angle**은 **360** 그리고 **pi** 는 **3.1416**으로 정의해줍니다.

6. 앞으로 전진하고 우측으로 꺾기 위해서 **length = 2\*pi\*x\_radius** 를 써서 전진 거리 **length** 를 지정합니다.

7. 그리고 나서 **for 문을 이용해서 0부터 x\_angle 동안 circle.forward(length), circle.right(360/x\_angle)**로 .circle 의 도움을 받지 않은 원을 그립니다.

**세번째 조건 :** **radians 과 degrees 함수와 같은 기능을 하는 MyRadians 과 MyDegrees 함수를 만들어보자.**

1. **코드 :**

**import math**

**def PI(pi):**

**for deg in range(0, 361, 30):**

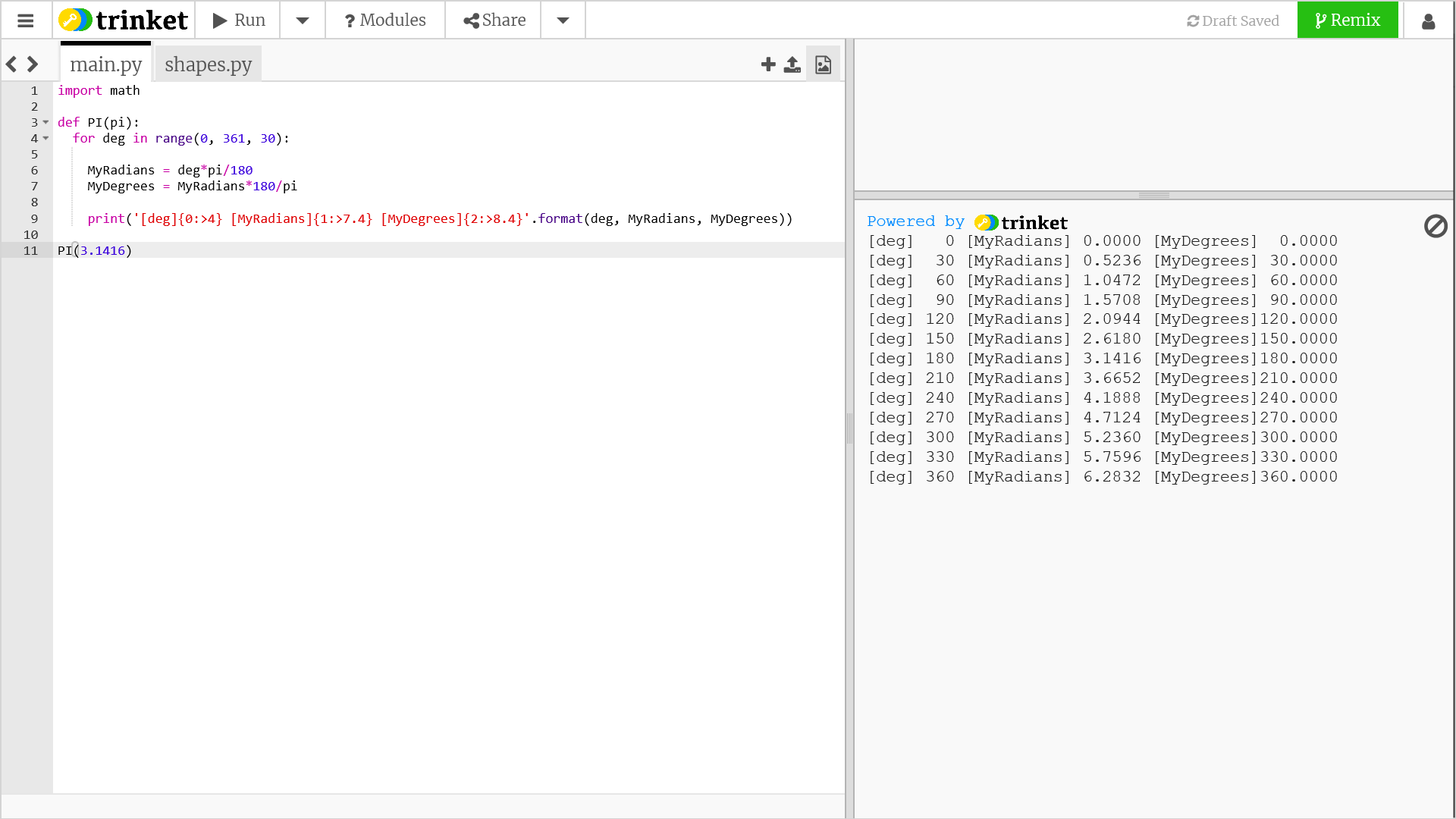
**MyRadians = deg\*pi/180**

**MyDegrees = MyRadians\*180/pi**

**print('[deg]{0:>4} [MyRadians]{1:>7.4} [MyDegrees]{2:>8.4}'.format(deg, MyRadians, MyDegrees))**

**PI(3.1416)**

1. **실행화면 사진 :**



1. **구현내용 설명 :**

1. **import math** 로 **math** 도구를 불러온다.

2. **def PI(pi) ~ PI(3.1416)으로 pi** 를 **3.1416**으로 정의한다.

3. **for deg in range(0, 361, 30)** 으로 **deg 가 0부터 361까지 30씩 증가** 시킨다.

4. 그리고 **MyRadians = deg\*pi/180, MyDegrees = MyRadians\*180/pi**로 변수에 값을 대입 한다.

5. **for** 문에 의해 **deg(각도)**가 변하는 동안 **MyRadians**와 **MyDegrees** 도 같이 변하게 된다.

6. 이제 이 값들을 보기 위해 **print(‘[deg]{0:>4} [MyRadians]{1:>7.4} [MyDegrees]{2:>8.4}’.format(deg, MyRadians, MyDegrees))** 를 써서 변하는 **deg** 에 따른 **MyRadians, MyDegrees**값을 볼 수 있도록 해준다.