**1번 문제**

**import turtle**

**import random**

**t=turtle.Turtle()**

**def colorshape():**

**c=["red","green","blue","yellow","purple","pink"] #색깔들**

**pc=[] #펜색깔이 들어갈 리스트변수**

**n=12 #n각형**

**for j in range(int(n/2)): #리스트 초기화**

**pc.append("black")**

**if(n%2==0): #마주보는 변이 있는 경우**

**i=0**

**while i<n/2 :**

**pc[i]=c[random.randrange(0,len(c))] #무작위로 c에있는 색깔중에 하나를 넣음**

**if (pc[i-1]==pc[i])|(pc[int(n/2)-1]==pc[0]): #이전 변과 색이 같거나 첫번째와 마지막이**

**i=i-1 #색이 같을 때 다시 넣어줌**

**i+=1**

**for i in range(n): #n각형 그리기**

**if i<n/2: #절반 이하 그릴 때**

**t.pencolor(pc[i])**

**else : #절반 이후 그릴 때**

**t.pencolor(pc[i-int(n/2)]) #절반을 그리면 색깔을 처음부터 다시 반복**

**t.forward(50)**

**t.left(360/n)**

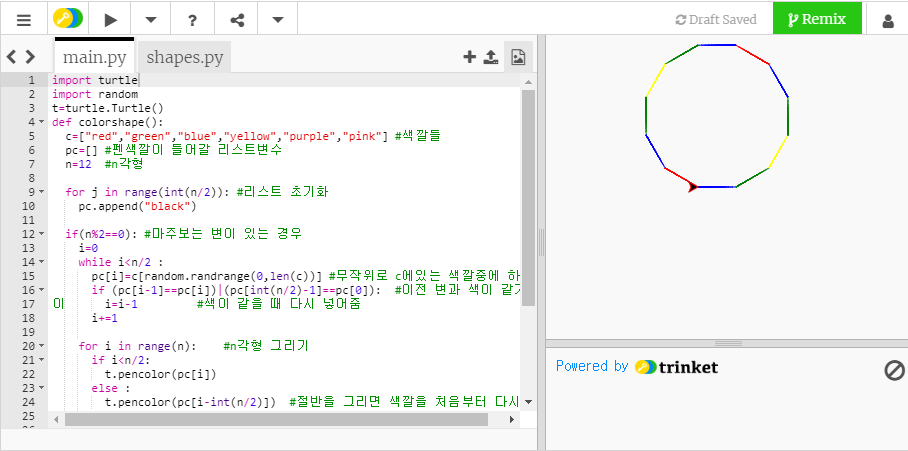
**else : #마주보는 변이 없는경우**

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

**t.forward(50)**

**t.left(360/n)**

**colorshape()**

****

**여러 색정보가 있는 리스트를 만들고 그 리스트에서 무작위로 색을 가져와 선의 색이 될 리스트에 넣습니다. 이때 인접한 색이 연속해서 출력되지 않도록, 무작위로 나온 색이 바로 전의 색과 같거나 첫번째 색과 마지막 색이 같은 경우 다시 무작위로 넣어줍니다.**

**N이 짝수일 때만 마주보는 면이 생기므로 N이 홀수일때는 흑백으로 출력합니다.**

**N이 짝수일 때, 절반 이후부터의 면은 첫번째 면과 마주보고 그 다음면은 두번째면, 그 다음은 세번째면, 이렇게 진행이 되므로 n각형의 반인 n/2까지의 선의 색 리스트를 만들고 그 리스트를 두 번 반복해서 출력함으로써 마주보는 면의 색이 같게 출력됩니다**

**2번문제**

import math

dic ={} #딕셔너리 생성

for i in range(0,361,30): #딕셔너리에 deg, radian, sin 값을 저장

rad = math.radians(i)

sinval=math.sin(rad)

dic[i]=[rad,sinval]

for i,j in dic.items(): #딕셔너리에 있는 값들 출력

print('[deg]{0:>4} [rad]{1:>7.4} [sin()]{2:>7.4}'\

.format(i,j[0],j[1]))

deg=int(input('degree?')) #deg를 입력받음

for i,j in dic.items(): #for문을 이용한 검색

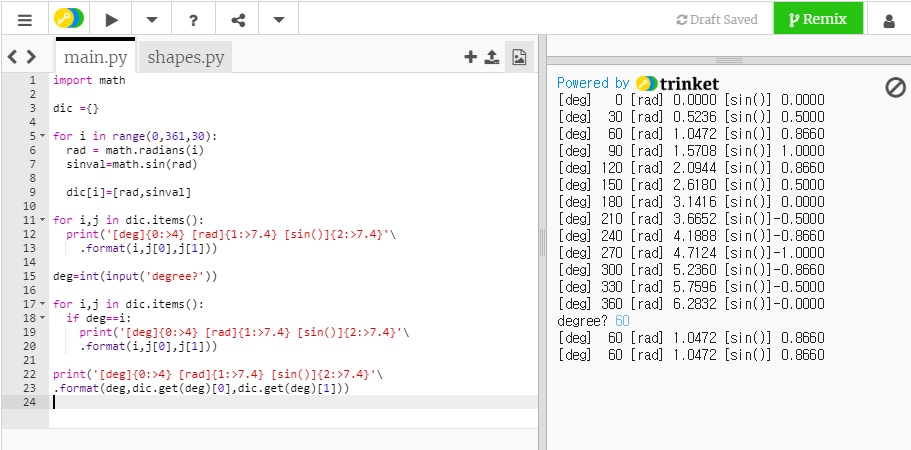
if deg==i:

print('[deg]{0:>4} [rad]{1:>7.4} [sin()]{2:>7.4}'\

.format(i,j[0],j[1]))

print('[deg]{0:>4} [rad]{1:>7.4} [sin()]{2:>7.4}'\ #get()함수를 이용한 검색

.format(deg,dic.get(deg)[0],dic.get(deg)[1]))



For문으로 딕셔너리에 deg를 30간격으로 0~360까지 deg를 key, radian과 sin을 value로 값들을 저장합니다.

For문을 이용한 검색은 딕셔너리에 처음부터 끝까지 반복하면서, 입력 받은 deg와 딕셔너리에 key값이 같으면 그 key에 맞는 value를 출력합니다

Get()함수를 이용한 검색은 함수에 입력 받은 deg값을 넣고, deg가 key인 value를 받아와서 출력합니다.