



AI 프로그래밍 11

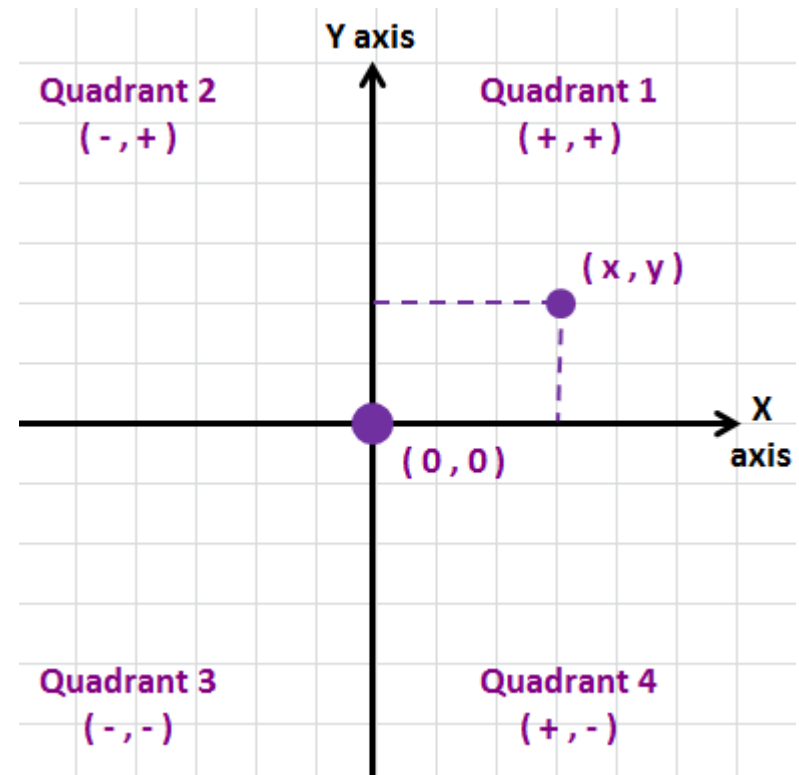
융합학과 권오영

oykwon@koreatech.ac.kr

터틀 그래픽

Turtle Graphics

- ❖ 거북이를 이동시키면서 그림을 그린다.
 - 거북이 꼬리에 펜이 달려서 거북이가 이동하면 흔적이 남는다.
- ❖ 아이들에게 프로그래밍을 소개하는데 사용
- ❖ 1967년 개발된 logo란 언어의 일부 (Wally Feurzeig, Seymour Papert, Cynthia Solomon)
- ❖ 파이썬 기본 모듈로 제공
- ❖ 명령어를 사용하여 그림을 그림
- ❖ 쉬운 명령어로 다양한 그래픽작업가능



Turtle Graphics

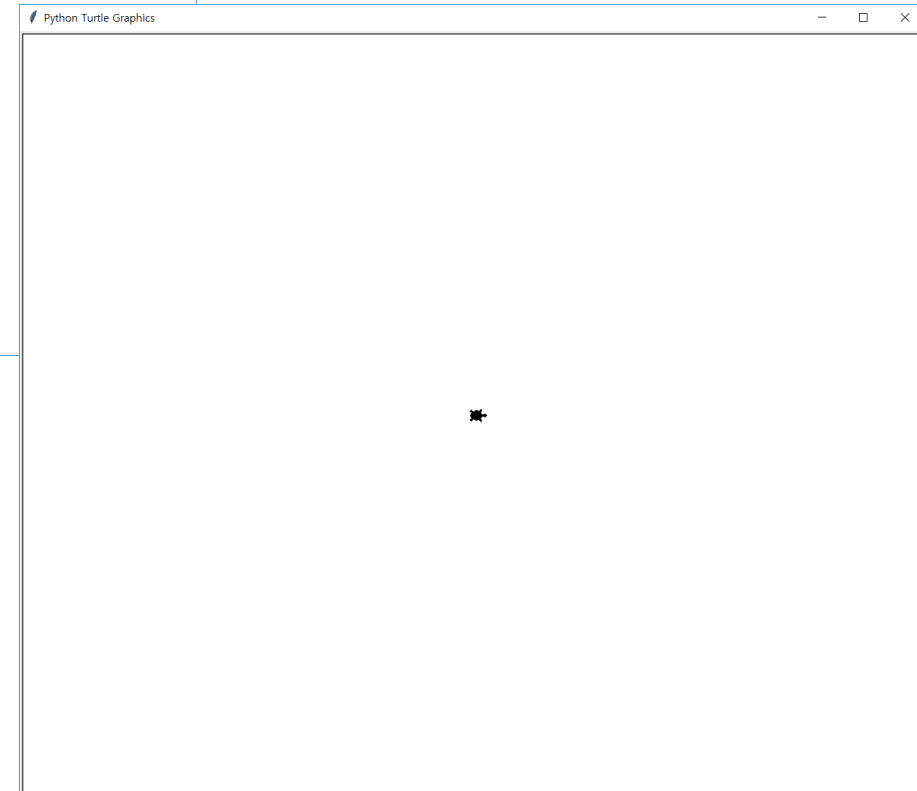
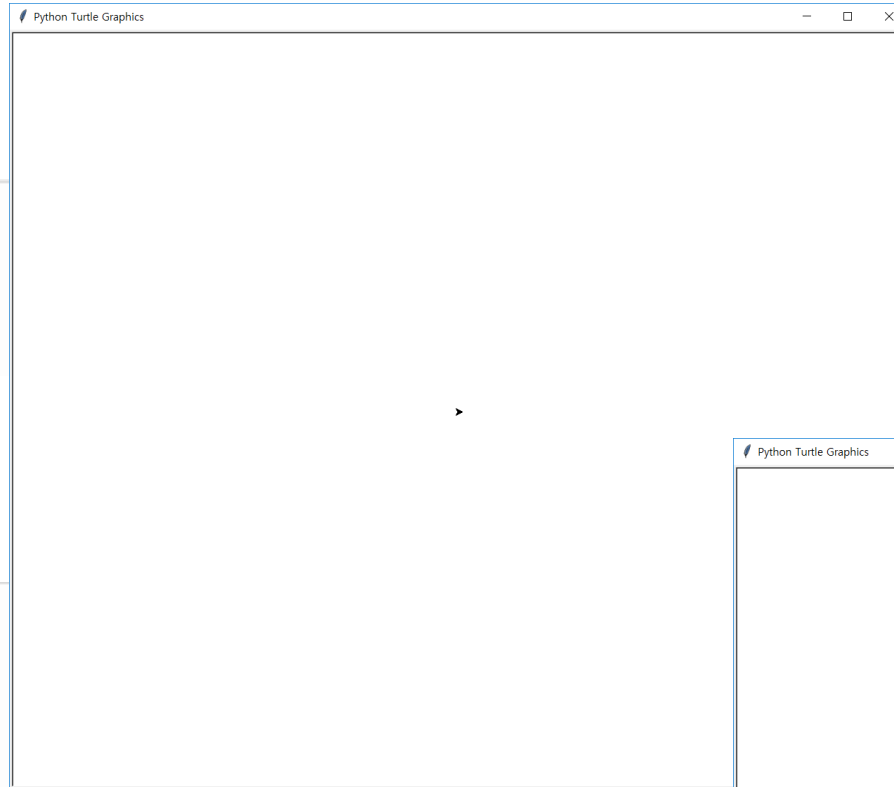
❖ 표준모드

- Heading이 동쪽으로 시작

```
1 import turtle
2
3 t = turtle.Turtle()
```

❖ 모양 변경

```
1 import turtle
2
3 t = turtle.Turtle()
4 t.shape('turtle')
5
6 # shape: arrow, turtle, circle,
7 #         square, triangle, classic
8
```



Turtle Graphics

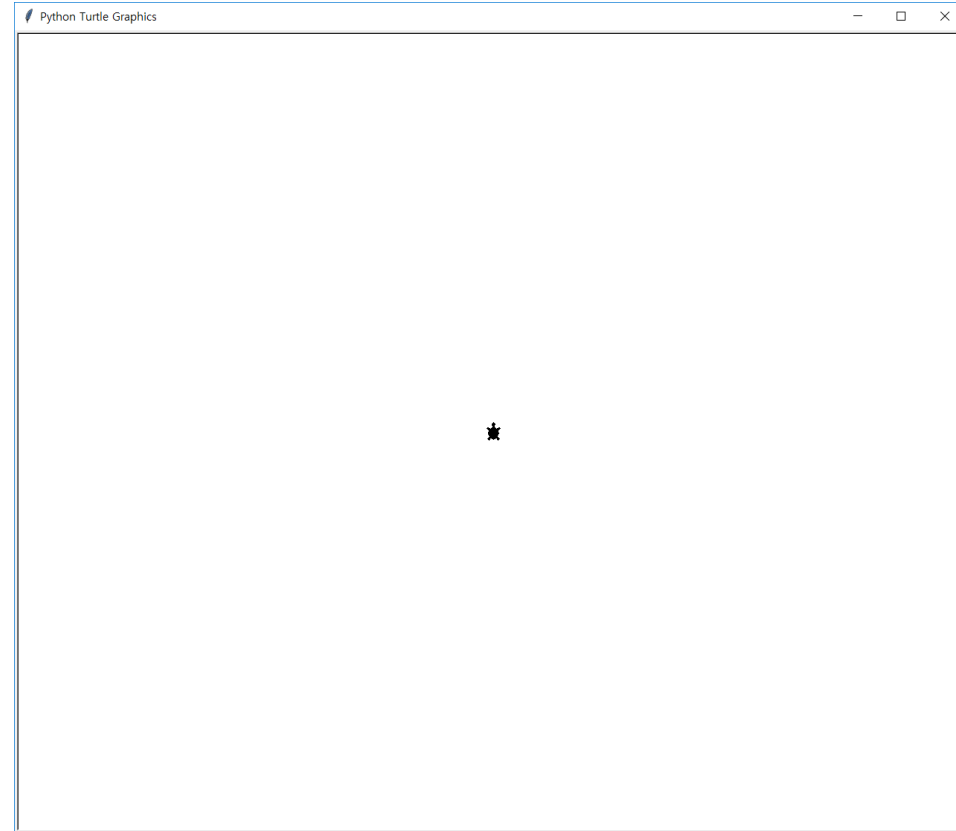
❖ 로고(logo) mode

- Heading이 북쪽으로 시작

```
1 import turtle
2
3 turtle.mode('logo')
4
5 t = turtle.Turtle()
6 t.shape('turtle')
7
```

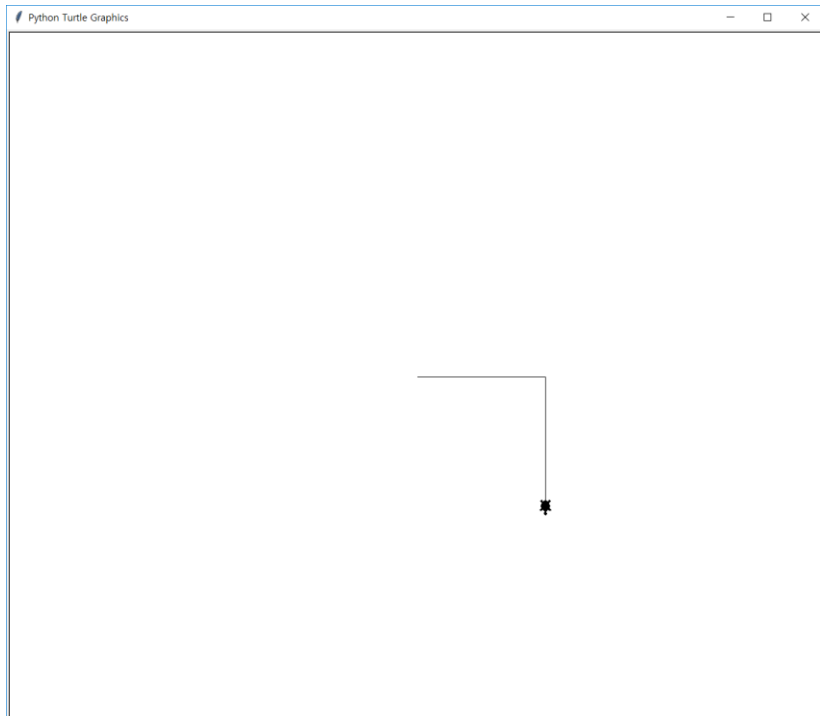
❖ 헤드의 방향설정 t.setheading(angle)

표준모드		로고모드	
0	동	0	북
90	북	90	동
180	서	180	남
270	남	270	서

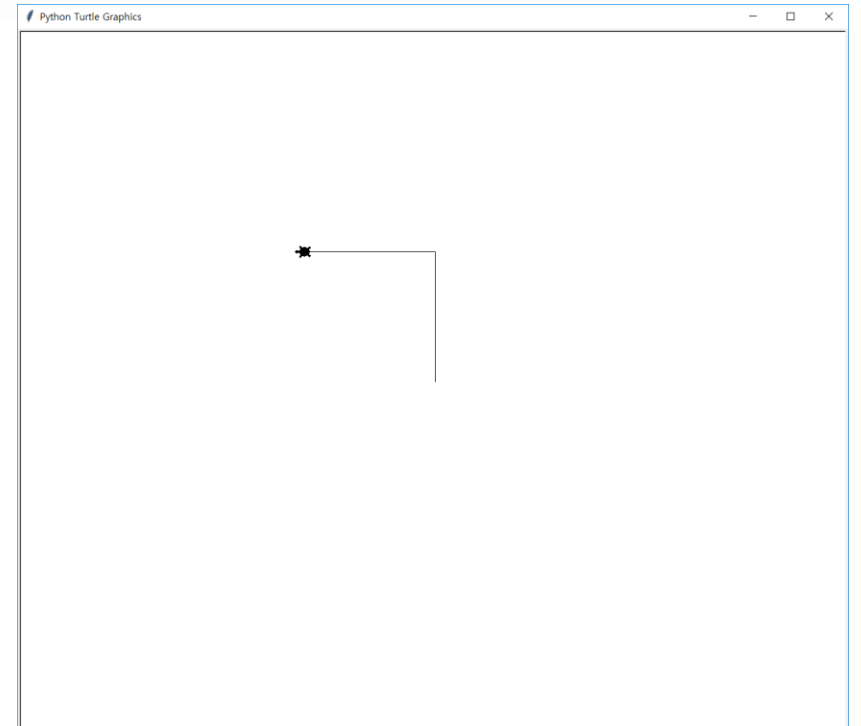


Turtle Graphics

```
1 import turtle
2
3 t = turtle.Turtle()
4 t.shape('turtle')
5
6 t.forward(200)    # 동쪽으로 200 이동
7 t.setheading(270) # 남쪽으로 전환
8 t.forward(200)
9
```



```
1 import turtle
2
3 turtle.mode('logo')
4
5 t = turtle.Turtle()
6 t.shape('turtle')
7
8 t.forward(200)    # 북쪽으로 200 이동
9 t.setheading(270) # 서쪽으로 전환
10 t.forward(200)
11
```



Turtle Graphics

❖ 초기화 명령어

- `reset()` : 화면을 지우고, 터틀을 가운데로 위치
- `clear()` : 화면을 지움

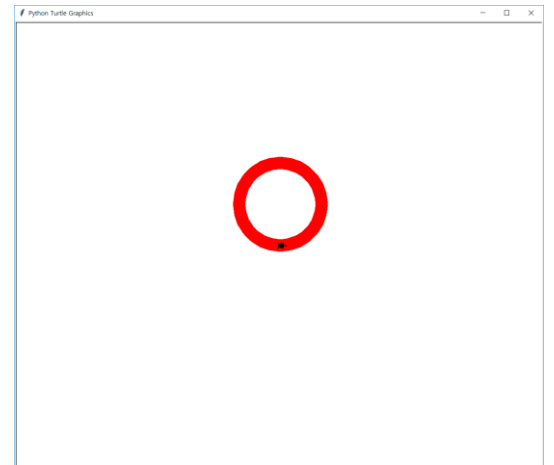
❖ 설정 명령어

- `speed(int)`: 이동 속도 설정 (10: 빠름, 6: 보통, 3: 느림)
- `penup()` : 펜들기
- `pendown()` : 펜 내리기 ghkaud 거북이를 이동시키면서 그림을 그린다.
- `pensize()` : 펜의 두께
- `pencolor(color_string)` : 펜 색 지정

```
1 import turtle
2
3 t = turtle.Turtle()
4 t.shape('turtle')
5 t.speed(6)
6 t.pensize(30)
7 t.pencolor('red')
8
9 t.circle(100)|
```

❖ 원 그리기

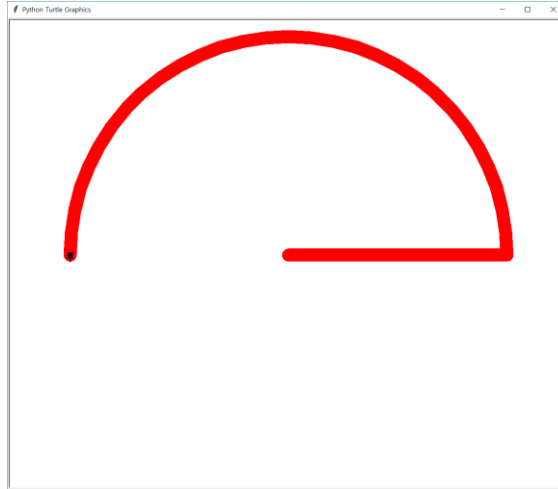
- `circle(radius)` : 반지름 `radius` 인 원 그리기
- `circle(radius, angle)` : 반지름 `radius` 인 원을 `angle` 만큼 그리기 (180: 반원)



Turtle Graphics

❖ 이동

- `forward(distance)` : *distance*만큼 앞으로 이동
- `backward(distance)` : *distance*만큼 뒤로 이동
- `right(angle)` : *angle* 각도 만큼 우회전
- `left(angle)` : *angle* 각도 만큼 좌회전
- `home()` : (0,0)으로 이동
- `goto(x,y)` : (*x*, *y*)로 이동
- `setheading(degrees)` 거북이의 방향 설정
(표준모드, 로고모드 구분)



```
1 import turtle
2
3 t = turtle.Turtle()
4 t.shape('turtle')
5 t.speed(6)
6 t.pensize(30)
7 t.pencolor('red')
8
9 t.goto(500,0)
10 t.setheading(90)|
11 t.circle(500,180)
```

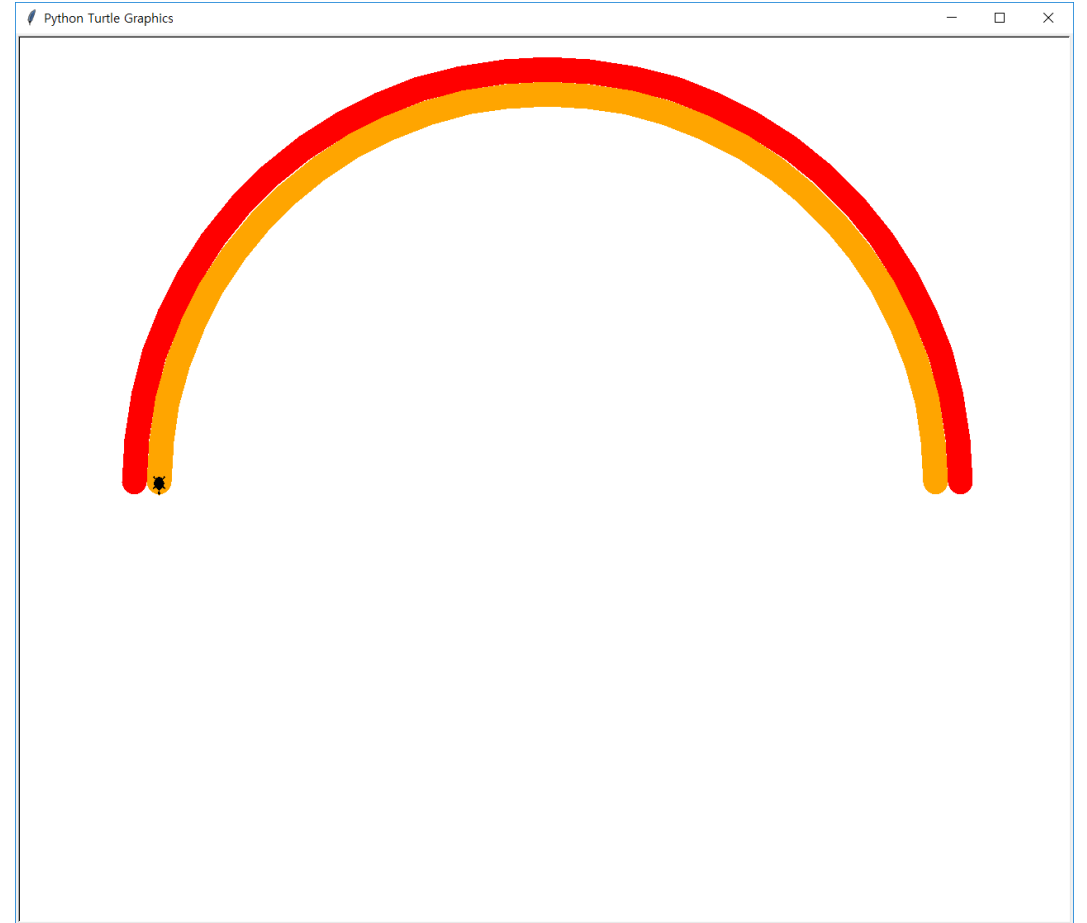
실습 :

사각형 그리기,
사각형 색 칠하기 (`begin_fill()`, `end_fill()`)
오각형 그리기,
`t.color('#FF69B4')` # 핫핑크(hotpink)

HTML Color Picker: https://www.w3schools.com/colors/colors_picker.asp

Turtle Graphics

```
1 import turtle
2
3 t = turtle.Turtle()
4 t.shape('turtle')
5 t.speed(6)
6 t.pensize(30)
7
8 t.pencolor('red')
9 t.penup()
10 t.goto(500,0)
11 t.setheading(90)
12 t.pendown()
13 t.circle(500,180)
14
15 t.pencolor('orange')
16 t.penup()
17 t.goto(470,0)
18 t.setheading(90)
19 t.pendown()
20 t.circle(470,180)|
```

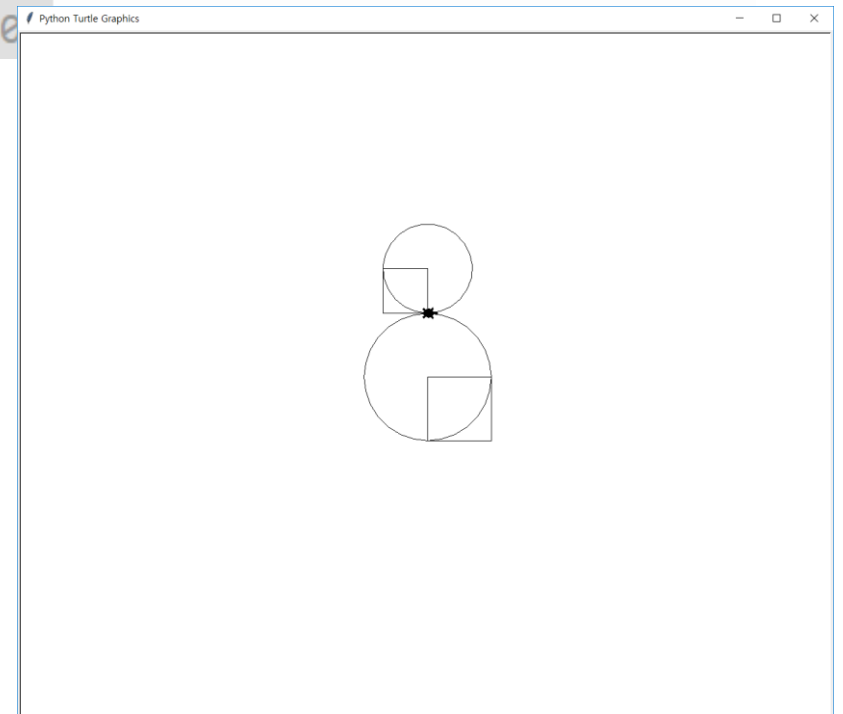


Turtle Graphics

- ❖ 사각형 그리기
- ❖ 눈사람 그리기

```
1 import turtle
2
3 def rectangle(t, x, y, side):
4     t.penup()
5     t.goto(x, y)
6     t.pendown()
7     t.setheading(0)
8
9     for i in range(4):
10        t.forward(side)
11        t.right(90)
12
13
14 def circle(t, x, y, radius):
15     t.penup()
16     t.goto(x, y)
17     t.pendown()
18     t.setheading(0)
19     t.circle(radius)
20
```

```
20
21 t = turtle.Turtle()
22 t.shape('turtle')
23 t.speed(6)
24
25 rectangle(t, 0, 0, 100)
26 rectangle(t, -70, 170, 70)
27
28 circle(t, 0, -100, 100)
29 circle(t, 0, 100, 70)
30
```



문제1. 무지개를 그려보자.

❖ 문제 1. 무지개를 그려보자.

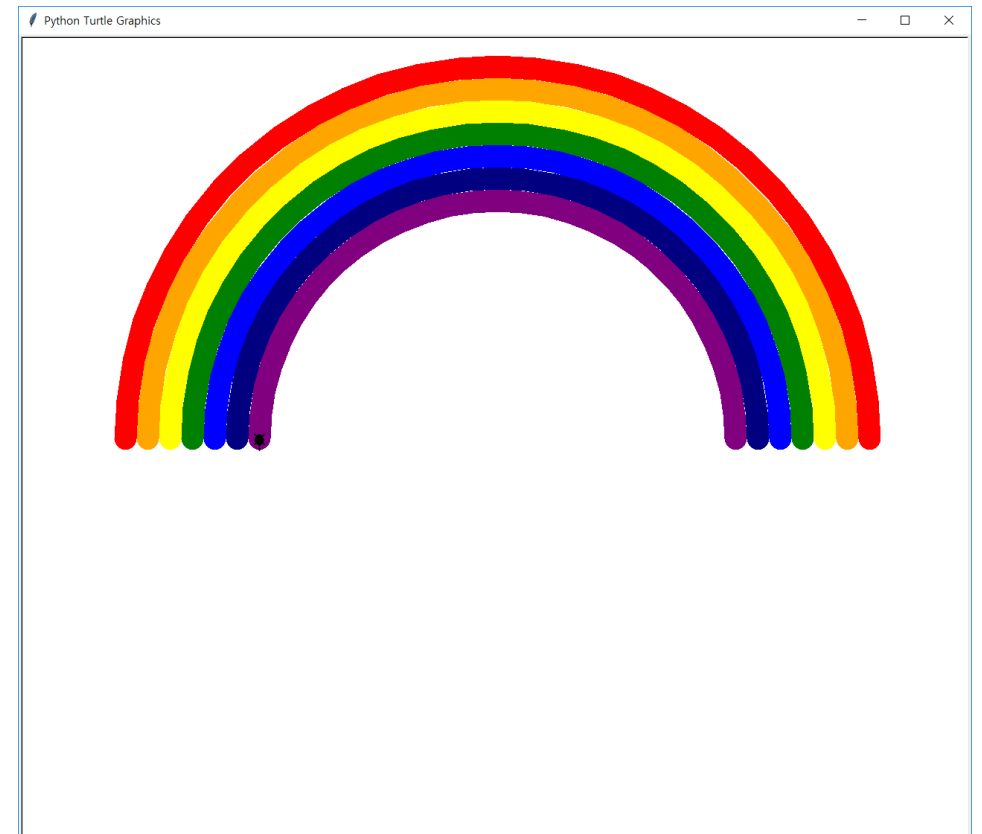
9쪽의 코드를 보면 붉은색 반원과 주황색 반원을 그리는 부분이 반복됩니다.

goto 와 circle에 들어가는 위치값을 변화시켜주면 다음과 같은 무지개를 그릴 수 있습니다.

순차로만 그려도 되지만, 반복과 함수를 사용하여 코드를 작성할 수 있습니다.

여러분이 아는 python지식을 다 이용해서 무지개를 그려보세요.

(color : red, orange, yellow, green, blue, navy, purple)



문제2. 나의집 그리기

- ❖ 집은 지붕(roof), 창문(window), 문(door), 벽(wall)으로 구분해 볼 수 있습니다. 각각을 함수로 만들어서 아래와 같은 형태의 집 그림을 그려보세요.

(함수를 사용하기 어려우면 순차 처리로 그려보세요.)

