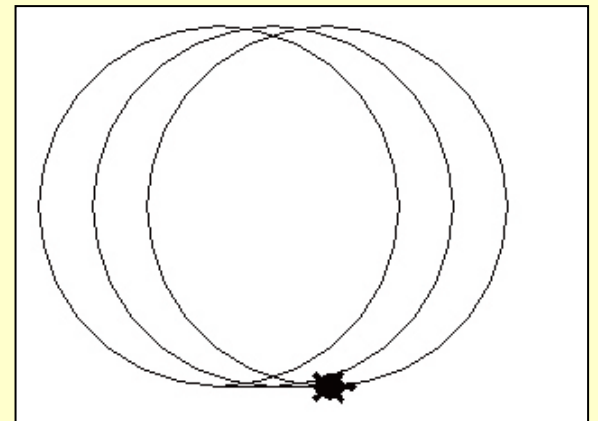


Lab: 변수는 어디에 유용할까?

- 다음과 같이 터틀 그래픽을 사용하여 반지름이 100픽셀인 3개의 원을 그리는 프로그램이 있다고 하자.

```
import turtle
t = turtle.Turtle()
t.shape("turtle")

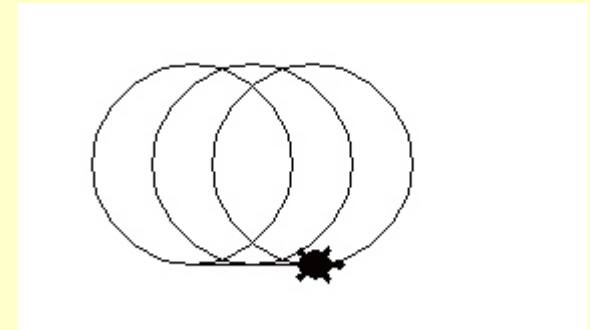
radius = 100
t.circle(radius) # 반지름이 100인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 100인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 100인 원이 그려 진다.
```



Sol.

```
import turtle
t = turtle.Turtle()
t.shape("turtle")

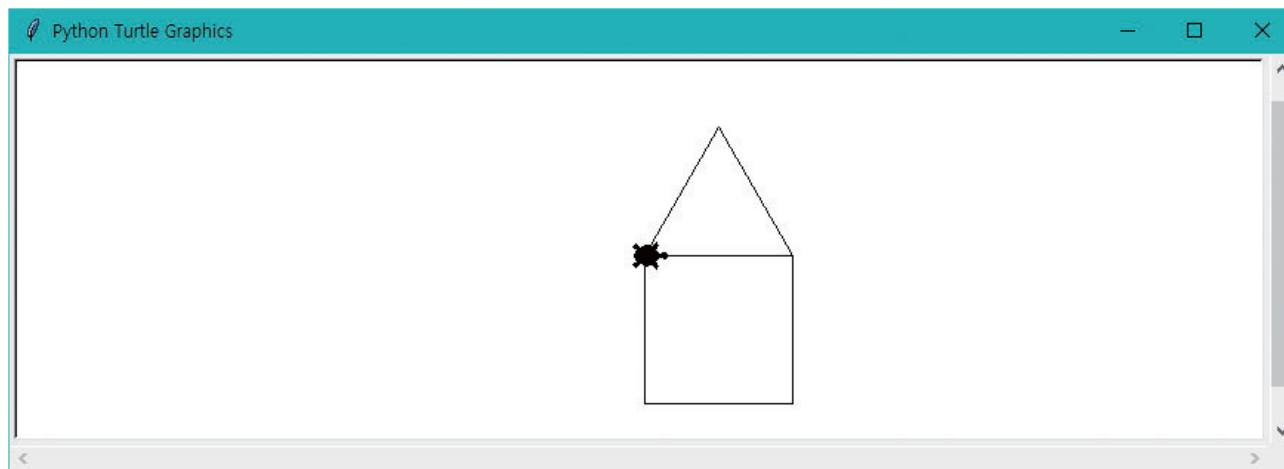
radius = 50
t.circle(radius) # 반지름이 50인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 50인 원이 그려 진다.
t.fd(30)
t.circle(radius) # 반지름이 50인 원이 그려 진다.
```



Lab: 집그리기

- 우리는 사용자로부터 집의 크기를 입력받아서 크기에 맞는 집을 그려보자.

집의 크기는 얼마로 할까요? 100

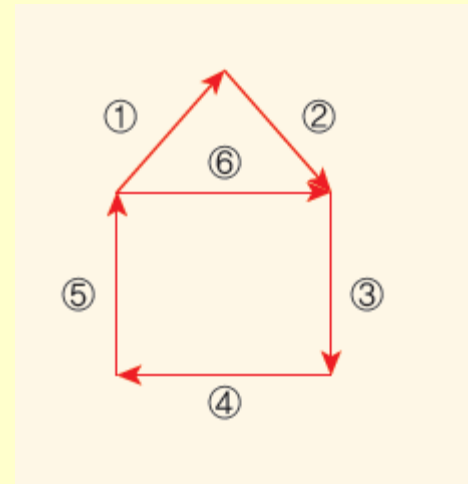


Solution

```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")
```

사용자로부터 집의 크기를 받아서 size라는 변수에 저장한다.
size = int(input("집의 크기는 얼마로 할까요? "))

```
t.left(60)  
t.forward(size) # ①  
t.right(120)  
t.forward(size) # ②  
t.right(30)  
t.forward(size) # ③  
t.right(90)  
t.forward(size) # ④  
t.right(90)  
t.forward(size) # ⑤  
t.right(90)  
t.forward(size) # ⑥  
  
turtle.done()
```



도전 문제



도전문제

집을 좀 더 정교하게 그려보자. 창문과 현관도 만들어 보자. 또 다음과 같이 속을 채워서 그려보자.

```
t.color("blue")  
t.begin_fill()  
...  
t.end_fill()
```

