

Lecture #2. 파이썬 기초 (2)

2D 게임 프로그래밍

이대현 교수



한국공학대학교
TECH UNIVERSITY OF KOREA

Turtle 모듈

- 거북이가 펜을 가지고, 화면 위를 다니면서 그림을 그림.
- 전진, 후진, 회전, 원 그리기 등 다양하게 움직이면서 그림을 그릴 수 있음.



펜을 물고 있는 거북이


모듈의 사용 문법

모듈을 사용하기 위해 수입(import)함.

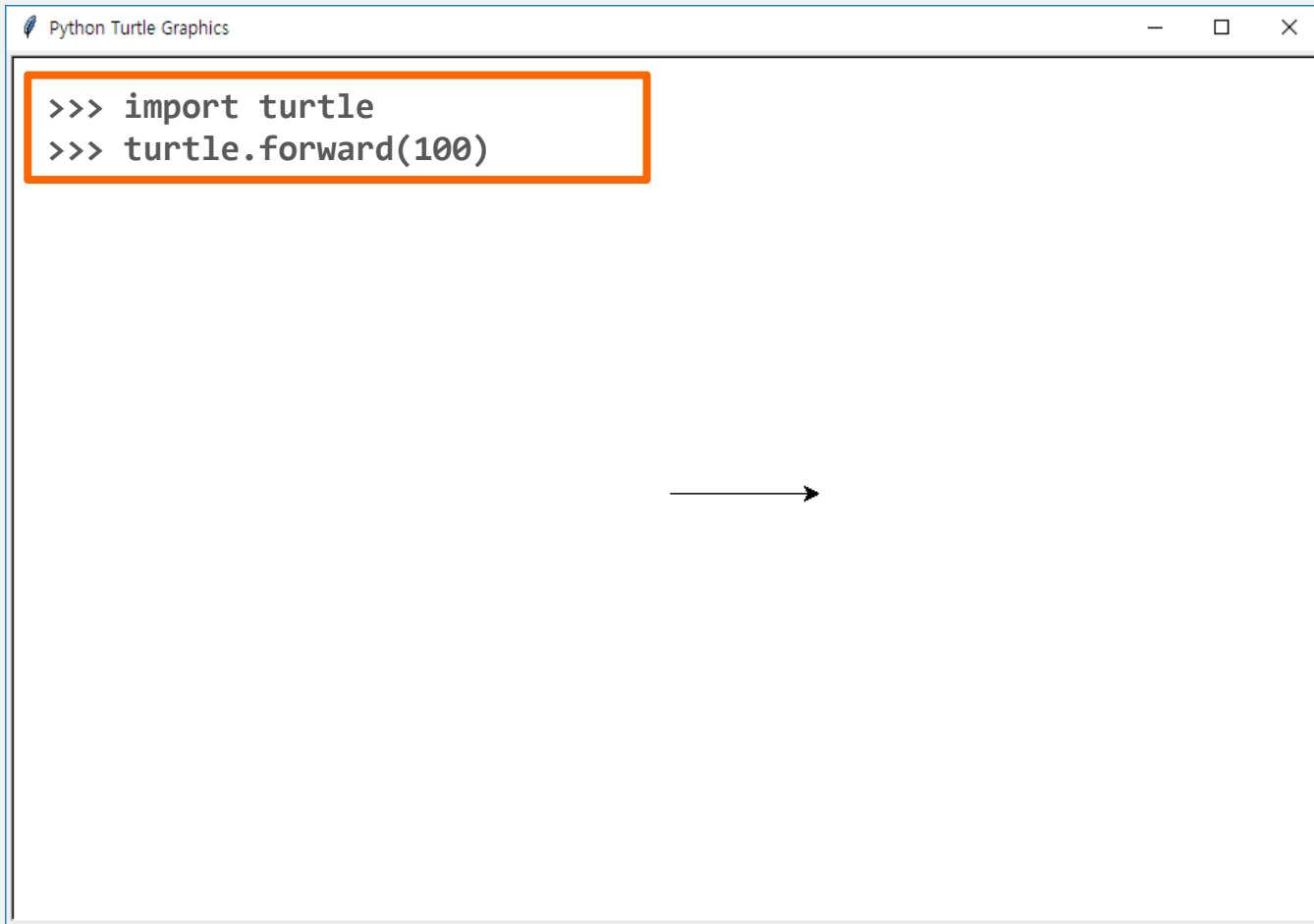


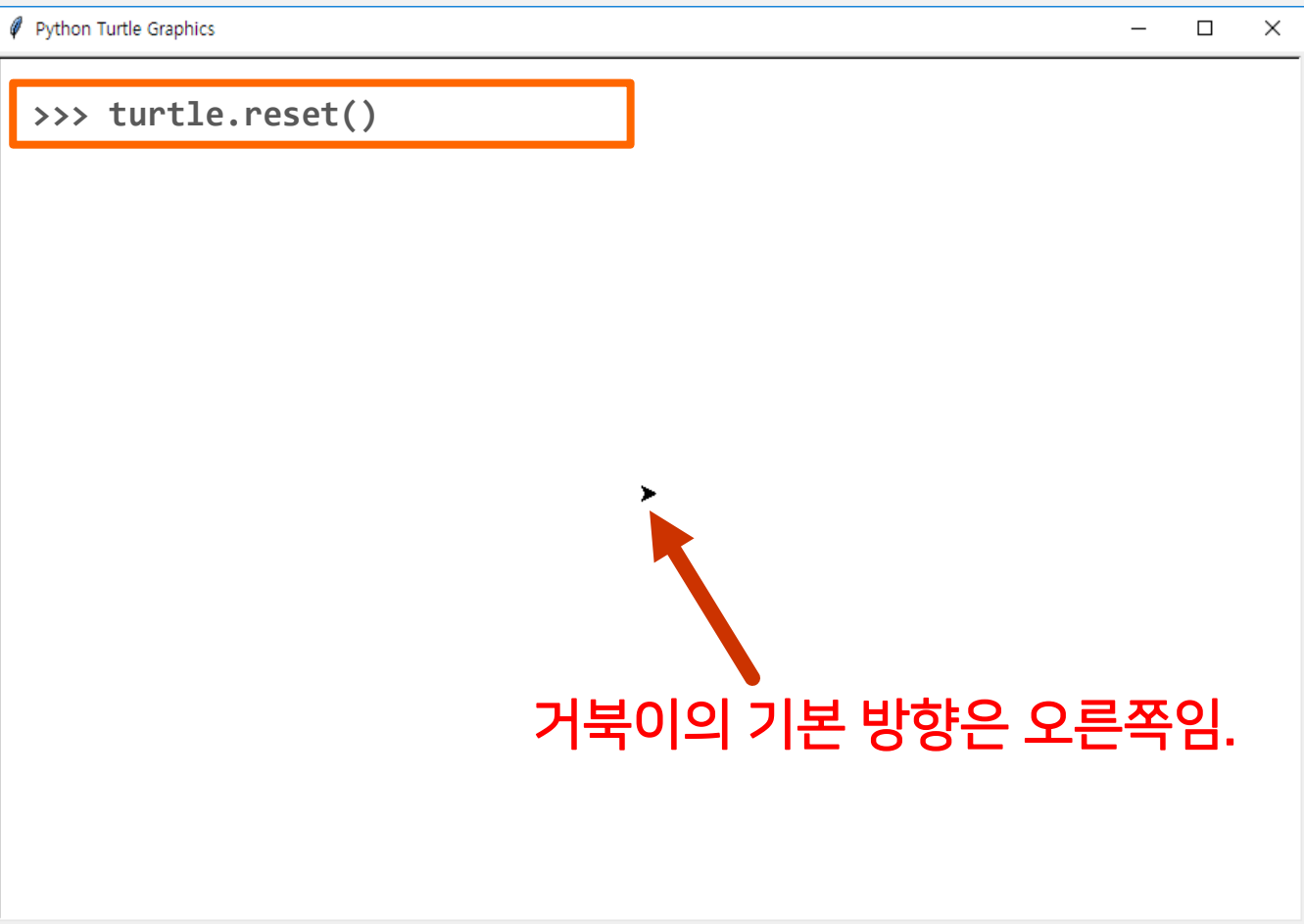
```
import turtle
```

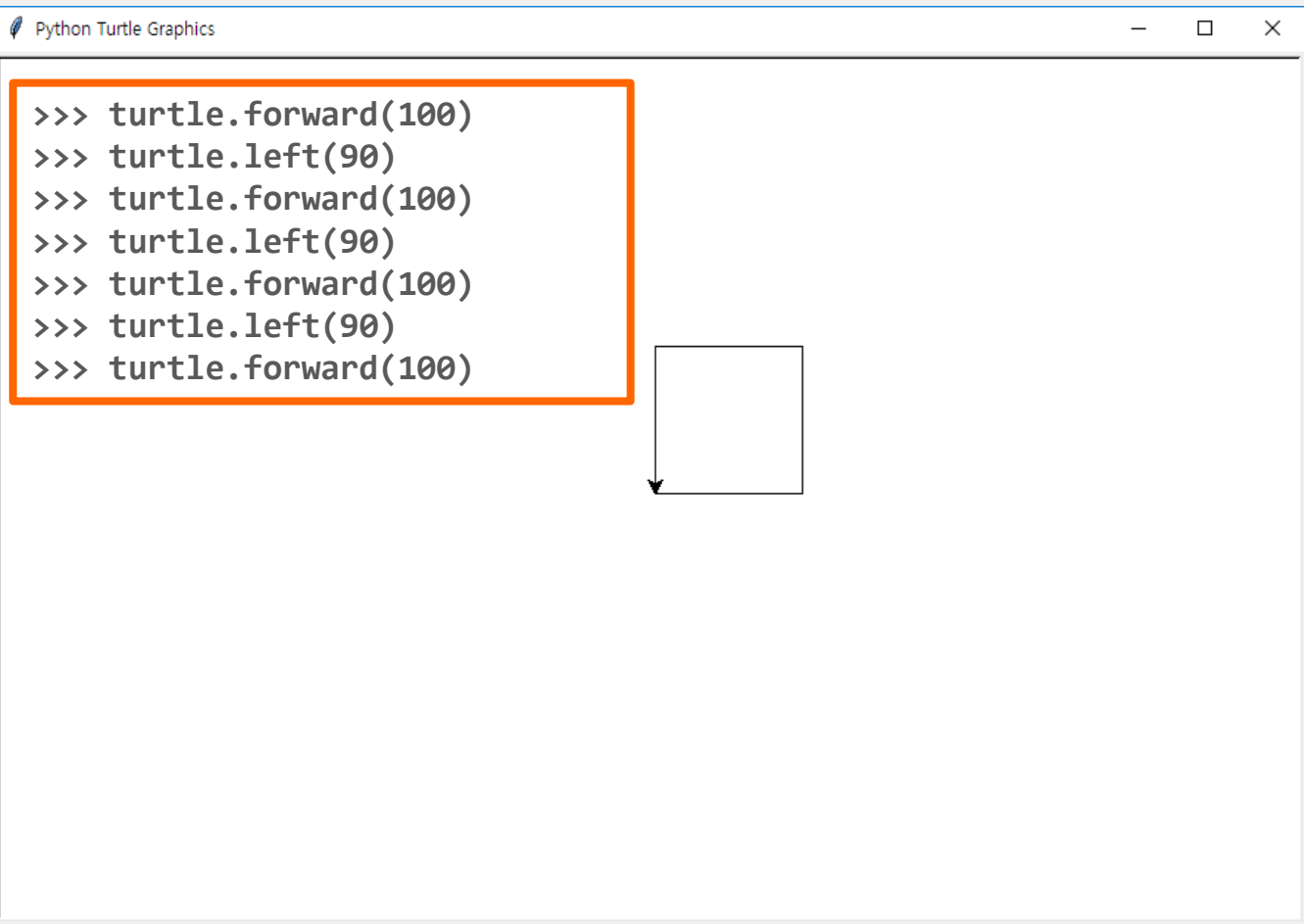
```
turtle.forward(100)
```

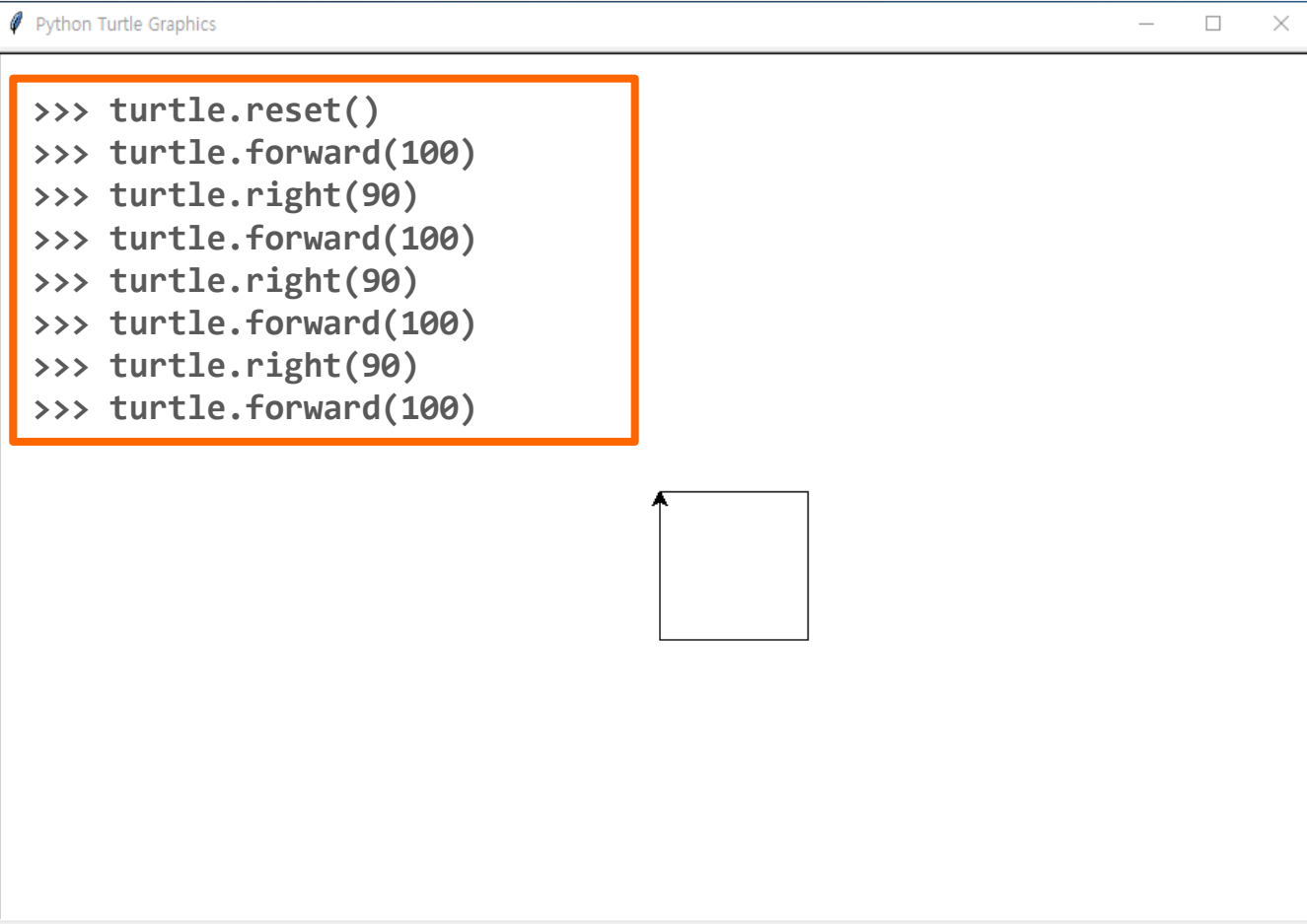


turtle 이 갖고 있는 기능(함수, function)
을 이용하여, 그림을 그린다.





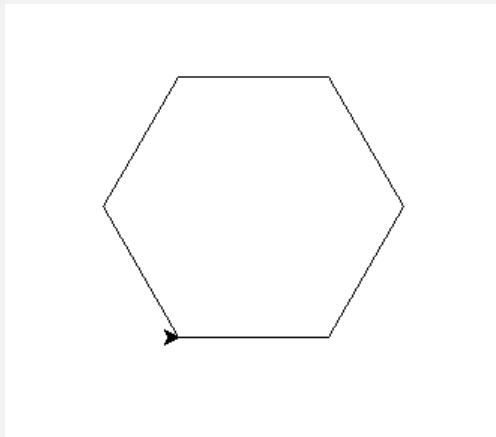


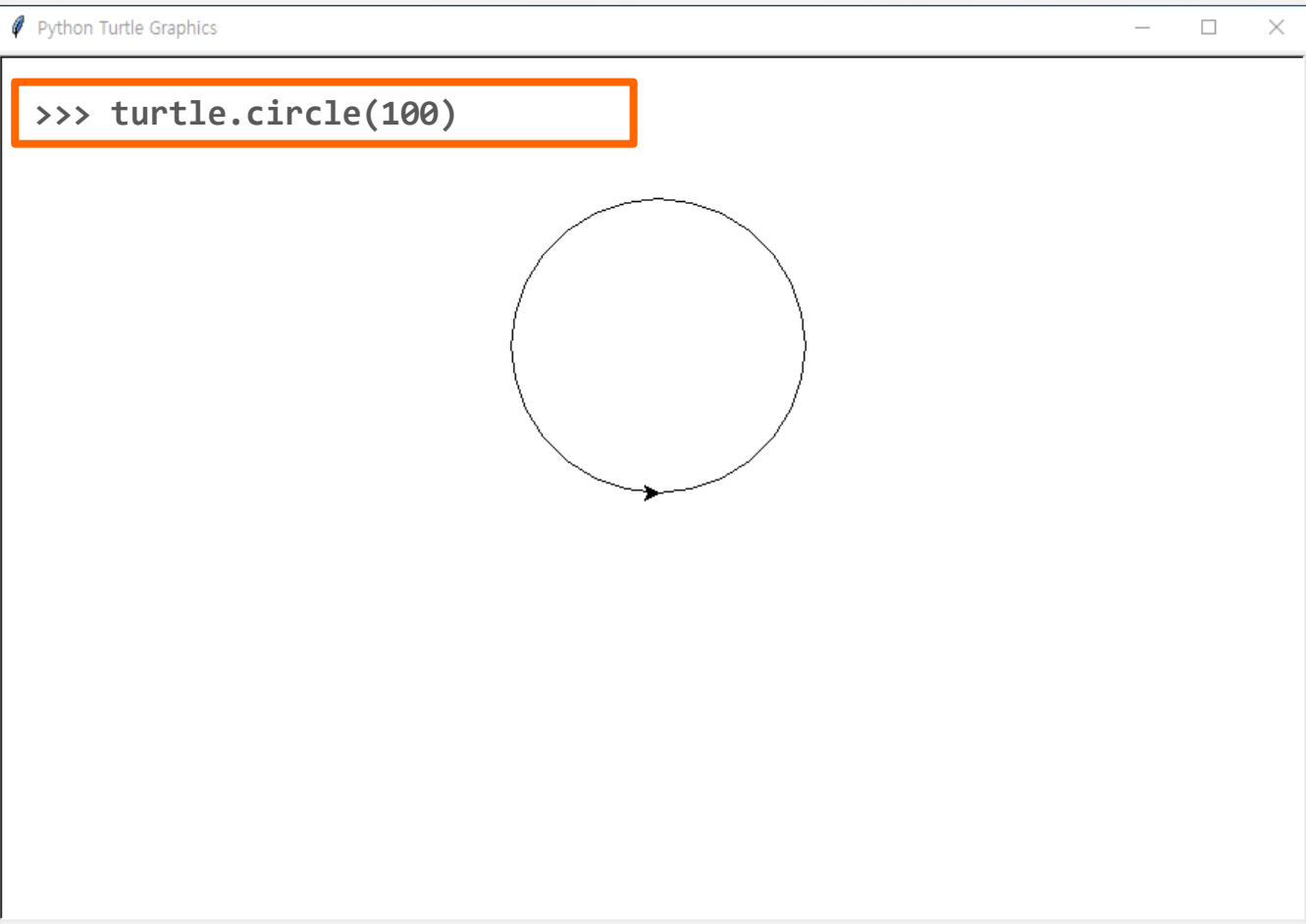


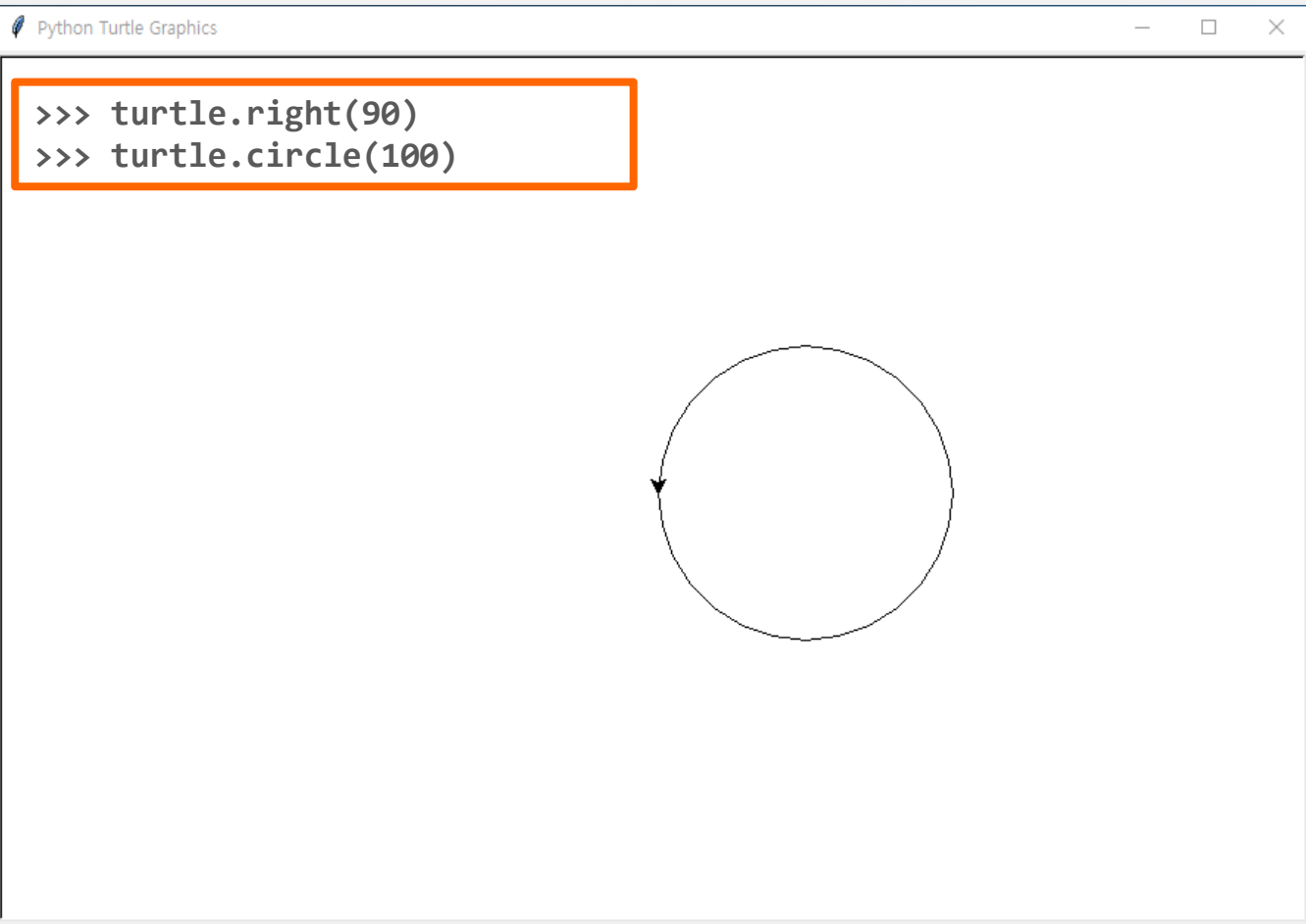
```
>>> turtle.forward(100)
>>> turtle.left(120)
>>> turtle.forward(100)
>>> turtle.left(120)
>>> turtle.forward(100)
```



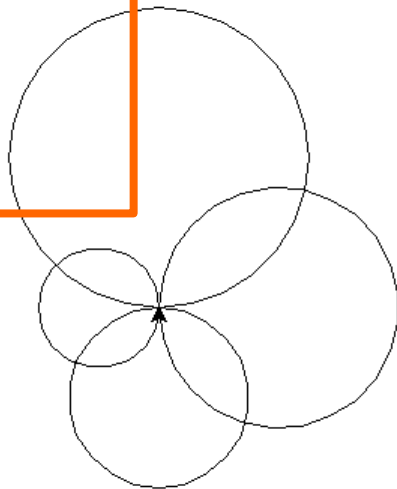
퀴즈 #1: 정육각형을 그려보자!



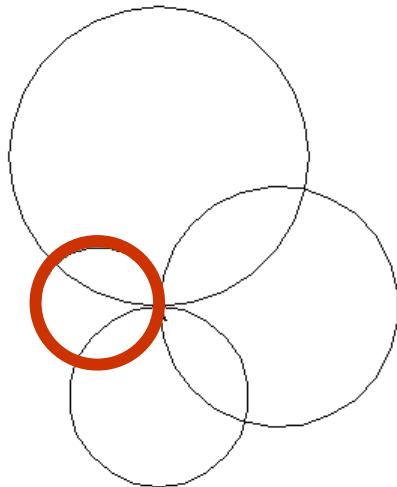




```
>>> turtle.circle(100)
>>> turtle.right(90)
>>> turtle.circle(80)
>>> turtle.right(90)
>>> turtle.circle(60)
>>> turtle.right(90)
>>> turtle.circle(40)
```



```
>>> turtle.updo()  
>>> turtle.undo()
```

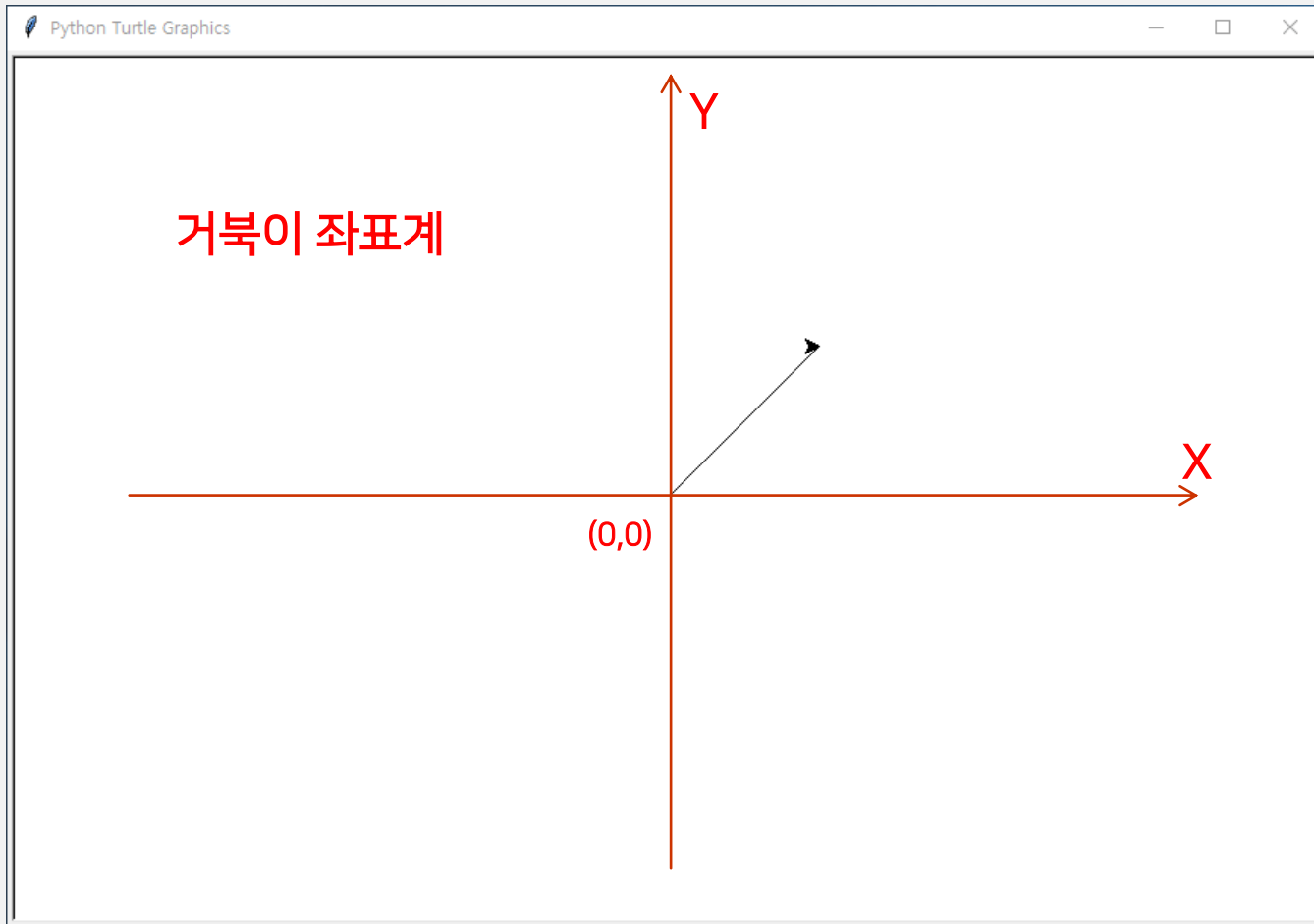


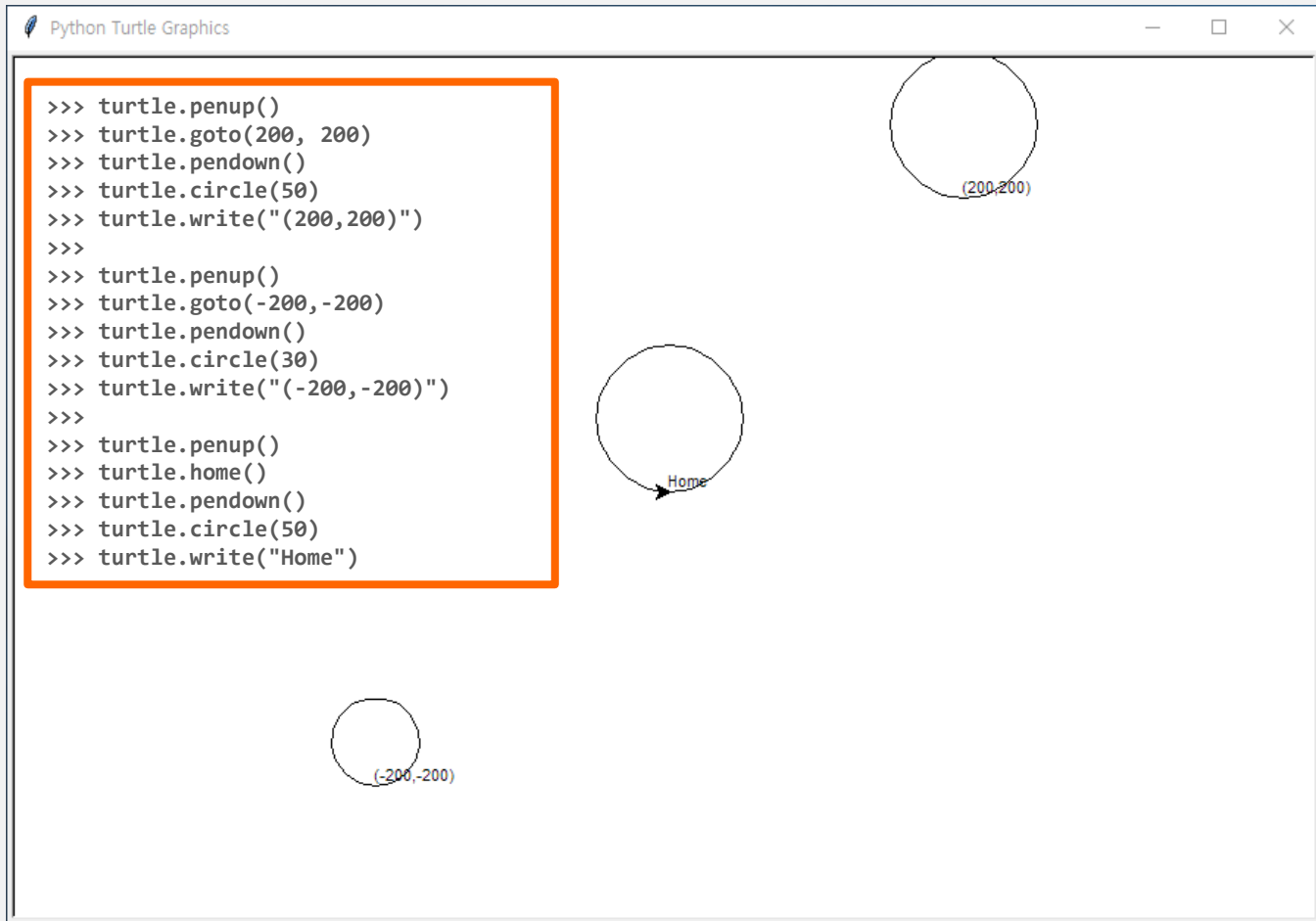
마지막에 그렸던 원이 없어짐.
이전 상태로 되돌아감.

```
>>> turtle.reset()  
>>> turtle.goto(100, 100)
```

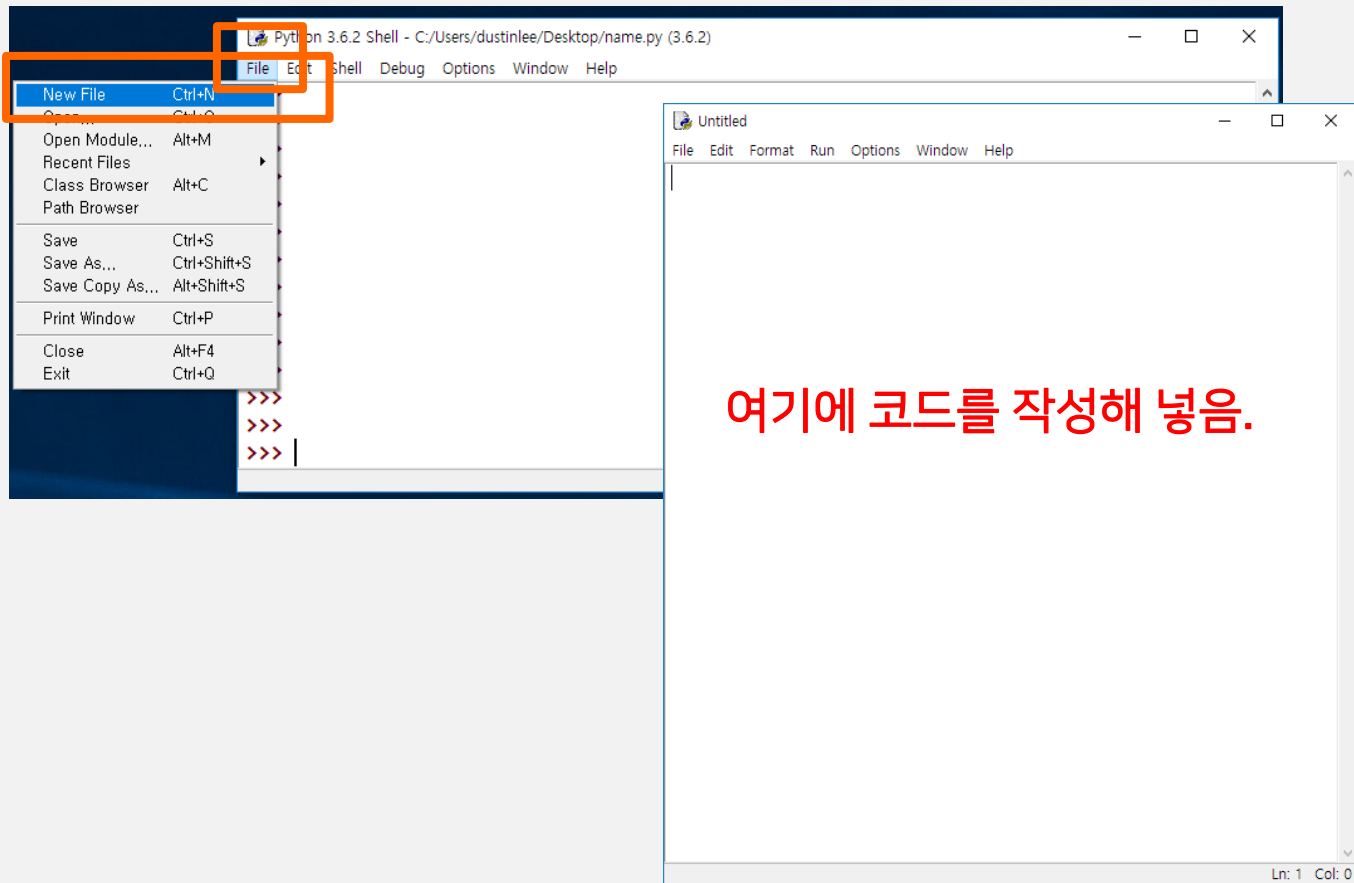


(100,100) 으로 이동한다.
거북이의 머리방향은 변함없이 여전히 오른쪽 방향.

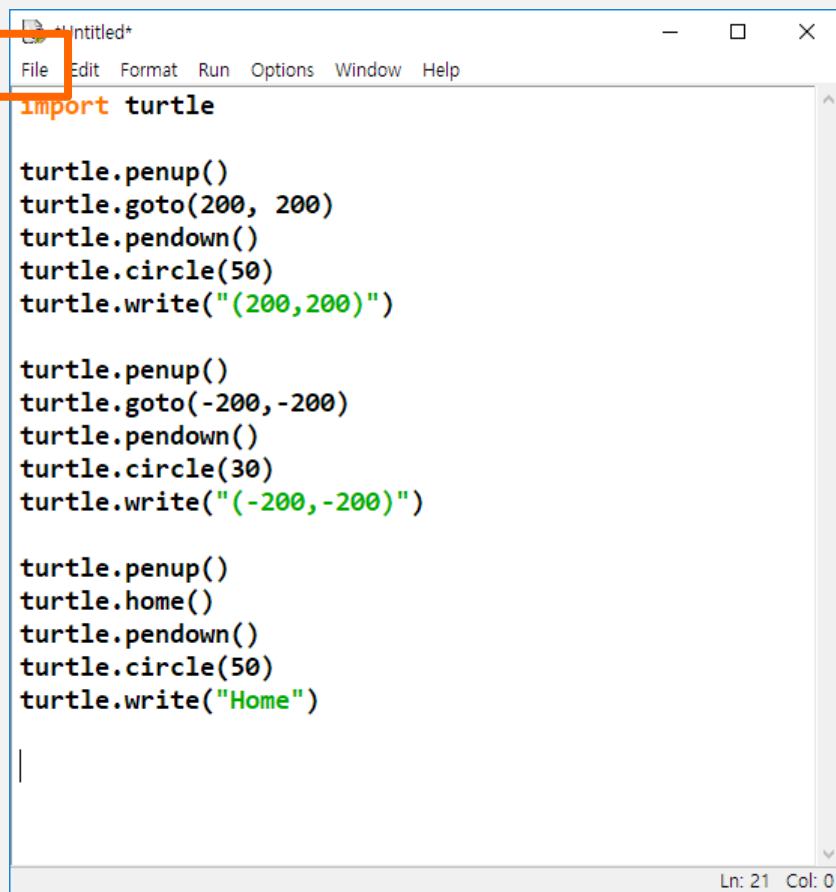




프로그램을 파일로 만들어서 저장



New File	Ctrl+N
Open...	Ctrl+O
Open Module...	Alt+M
Recent Files	
Class Browser	Alt+C
Path Browser	
Save	Ctrl+S
Save As...	Ctrl+Shift+S
Save Copy As...	Alt+Shift+S
Print Window	Ctrl+P
Close	Alt+F4
Exit	Ctrl+Q



The screenshot shows a Python IDE window titled "Untitled*" with a menu open. The menu is located on the left side of the window, and the "Save" option is highlighted. The code in the editor is as follows:

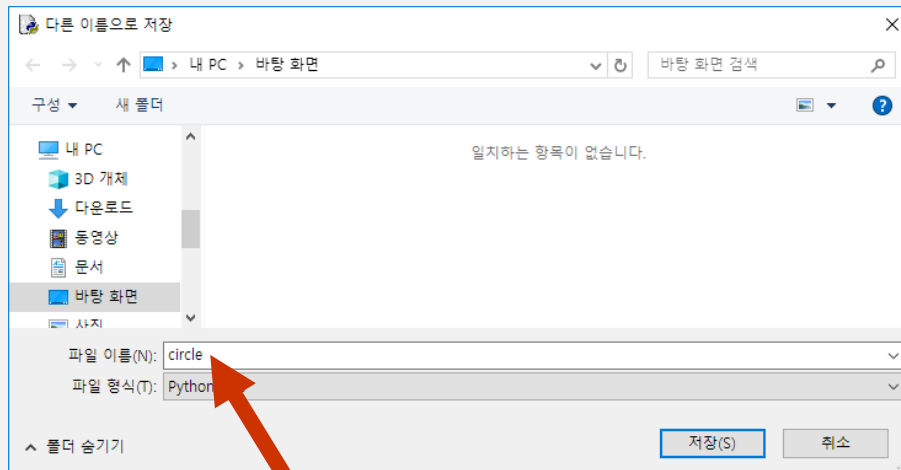
```
import turtle

turtle.penup()
turtle.goto(200, 200)
turtle.pendown()
turtle.circle(50)
turtle.write("(200,200)")

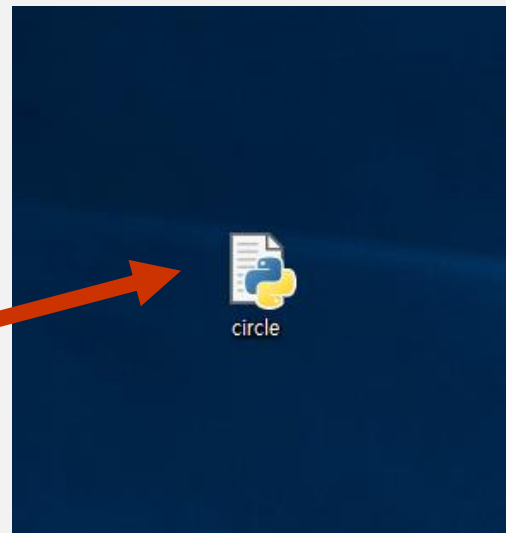
turtle.penup()
turtle.goto(-200,-200)
turtle.pendown()
turtle.circle(30)
turtle.write("(-200,-200)")

turtle.penup()
turtle.home()
turtle.pendown()
turtle.circle(50)
turtle.write("Home")
```

The status bar at the bottom right of the window indicates "Ln: 21 Col: 0".



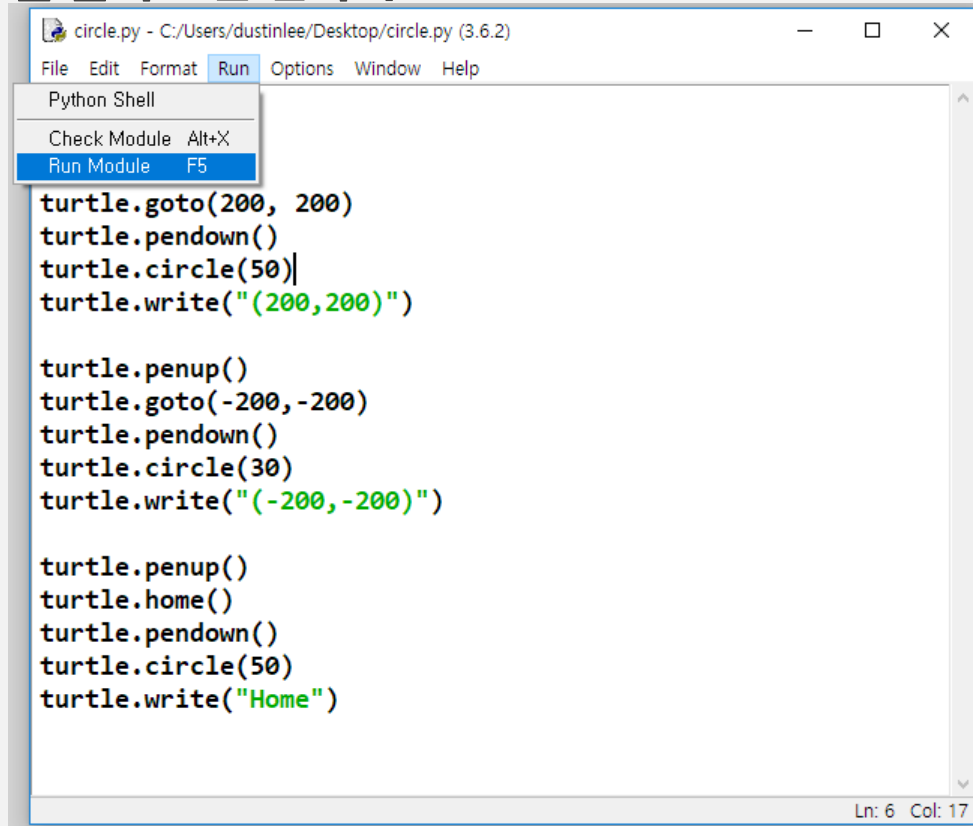
circle이라는 이름으로 바탕
화면에 저장.



바탕화면에 circle.py 라는
이름의 파일이 생성됨.

프로그램 실행 방법 #1

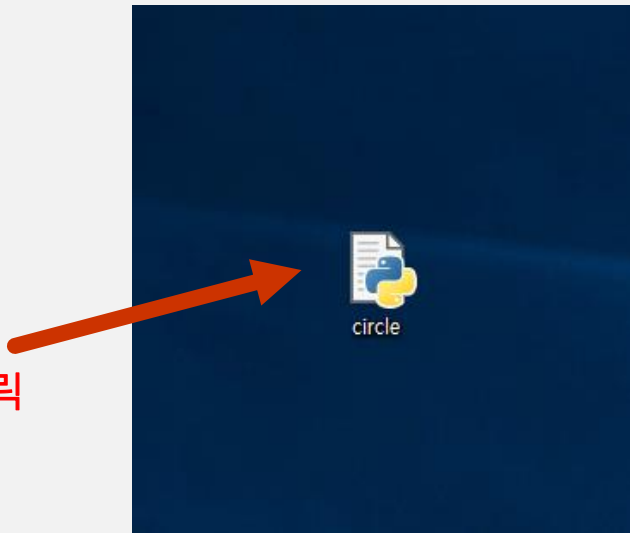
- Run → Run Module 을 클릭 또는 단축기 F5

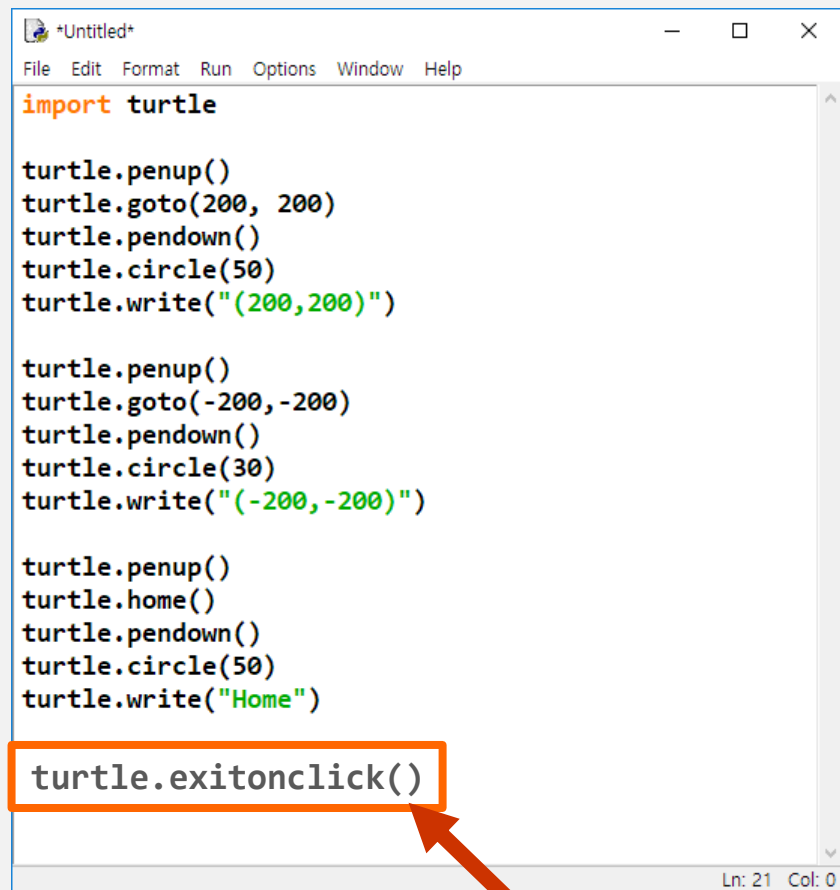


프로그램 실행 방법 #2

- 프로그램 파일을 더블 클릭하여 실행.
- 문제점은?

circle.py 를 더블 클릭





```
*Untitled*
File Edit Format Run Options Window Help

import turtle

turtle.penup()
turtle.goto(200, 200)
turtle.pendown()
turtle.circle(50)
turtle.write("(200,200)")

turtle.penup()
turtle.goto(-200, -200)
turtle.pendown()
turtle.circle(30)
turtle.write("(-200, -200)")

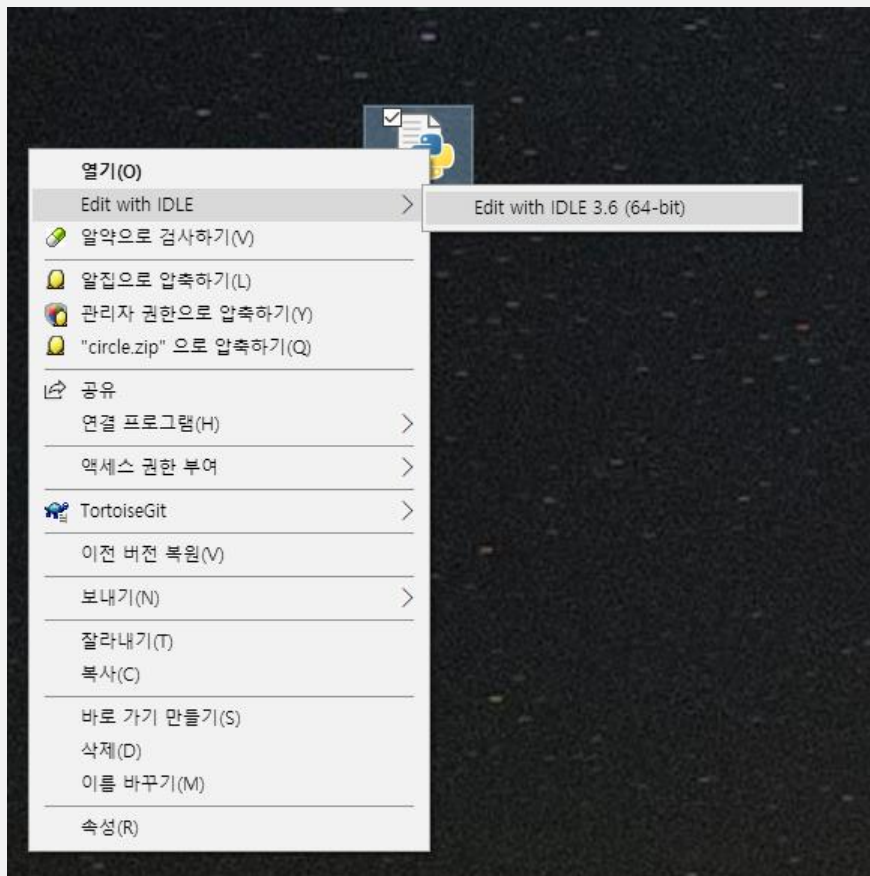
turtle.penup()
turtle.home()
turtle.pendown()
turtle.circle(50)
turtle.write("Home")

turtle.exitonclick()
```

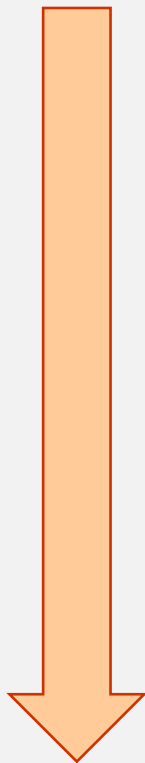
Ln: 21 Col: 0

코드 마지막 부분에 exitonclick() 추가.

마우스 오른쪽 버튼을 클릭하면, 소스코드를 직접 편집 가능.



파이썬 문장은 위에서부터 아래로 차례로 실행



```
circle.py - C:\Users\dustinlee\Desktop\circle.py (3.6.2)
File Edit Format Run Options Window Help

import turtle

turtle.penup()
turtle.goto(200, 200)
turtle.pendown()
turtle.circle(50)
turtle.write("(200,200)")

turtle.penup()
turtle.goto(-200, -200)
turtle.pendown()
turtle.circle(30)
turtle.write("(-200,-200)")

turtle.penup()
turtle.home()
turtle.pendown()
turtle.circle(50)
turtle.write("Home")

turtle.exitonclick()
```

Ln: 1 Col: 0

문법: 조건문 (Conditional Statement)

- 조건을 검사하여, 그 결과에 따라 처리를 하는 문장

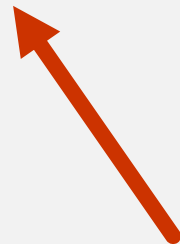
```
import turtle
```

```
shape = input("Enter Shape: ")
```

```
if shape == 'circle':
```

```
    turtle.reset()
```

```
    turtle.circle(50)
```



만약 shape 이 'circle'이면, 원을 그린다.

이 조건이 참(True)이면,

```
if shape == 'circle':
```

```
    turtle.reset()  
    turtle.circle(50)
```

들여쓰기(indentation)

*** 매우 중요 ***

일반적으로 공백4개씩

조건이 참이면, 들여쓰기된 블록을 실행함.

여기 블록에 적힌 대로 실행하라.

test_age.py

```
shape = input("Enter Shape: ")  
if shape == 'circle':  
    turtle.reset()  
    turtle.circle(50)
```

test_age.py

```
shape = input("Enter Shape: ")  
if shape == 'circle':  
    turtle.reset()  
turtle.circle(50)
```



문법: 조건문 (Conditional Statement) 확장형

```
if shape == 'circle':  
    turtle.reset()  
    turtle.circle(50)  
elif shape == 'triangle':  
    turtle.reset()  
    turtle.forward(50); turtle.left(120)  
    turtle.forward(50); turtle.left(120)  
    turtle.forward(50)  
else:  
    print('Unknown shape')
```

문법: while 반복문 (Iteration Statement)

- 어떤 조건을 만족하는 동안, 계속해서 반복적으로 실행하는 문장.

```
while <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    <수행할 문장3>  
    ...
```

```
import turtle
```

```
count = 10
```

```
while (count > 0):  
    turtle.forward(100)  
    turtle.left(30)  
    count = count - 1
```



count가 0 보다 크면 계속해서 반복한다. 뭘? (turtle을 앞으로 100 이동,
그리고 왼쪽으로 30도 회전, 그리고 count 값 하나 감소)

```
import turtle
```

```
count = 10
```

```
while (count > 0):
```

```
    turtle.forward(100)
```

```
    turtle.left(30)
```

```
    count -= 1
```

이 조건이 참(True)인 동안



들여쓰기(indentation)

*** 매우 중요 ***

여기 블록을 반복적으로 실행한다.

