




[Hello, KyungBock! 2nd semester]

[2학기 수업을 위한 준비]

구글 드라이브 폴더 정리

내 드라이브 > 2019_Python_김선경 ▾			
이름 ↑	소유자	마지막으로 수정한 날짜	파일 크기
 1학기	나	오전 9:31 나	—

1학기 폴더를 만들어 1학기의 모든 자료를 넣으세요

2학기 수업 자료 탑재 : **github**

https://github.com/swKyungbock/2019_Python/wiki

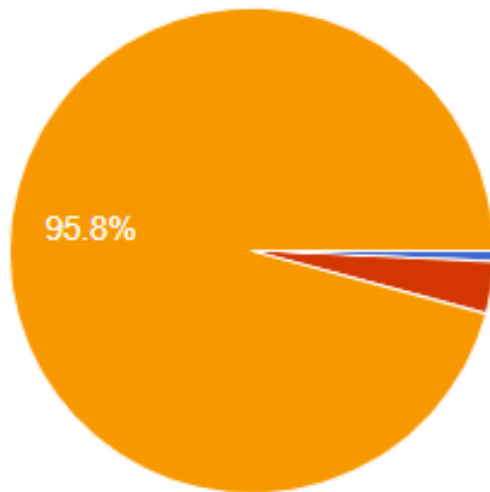
[1학기의 수업 다시보기]

#데이터형(Data Type) , 변수

3월

1. 데이터형(Data Type), 기본 연산자

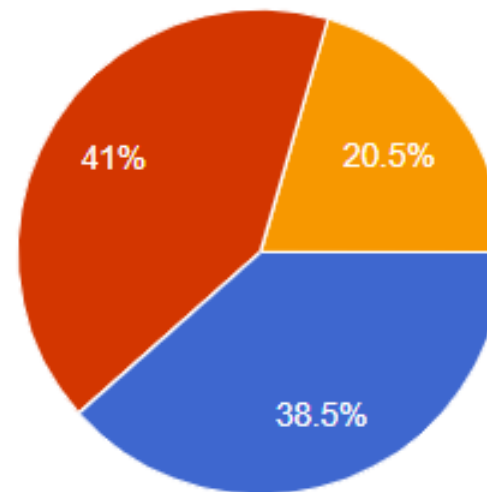
응답 144개



7월

3. 변수의 개념

응답 78개



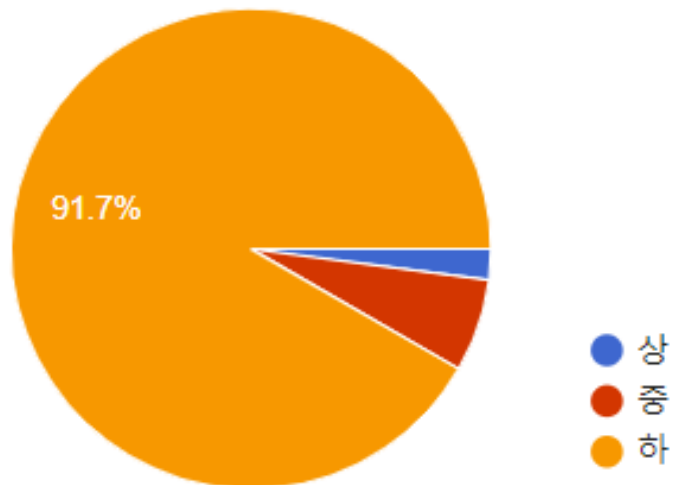
[1학기의 수업 다시보기]

#조건문 : if-elif-else

3월

2. 조건문(if statement)

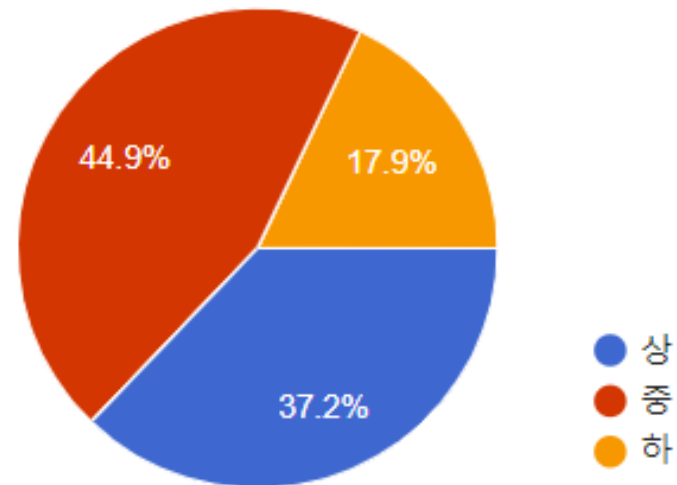
응답 144개



7월

4. 조건문(if statement)

응답 78개



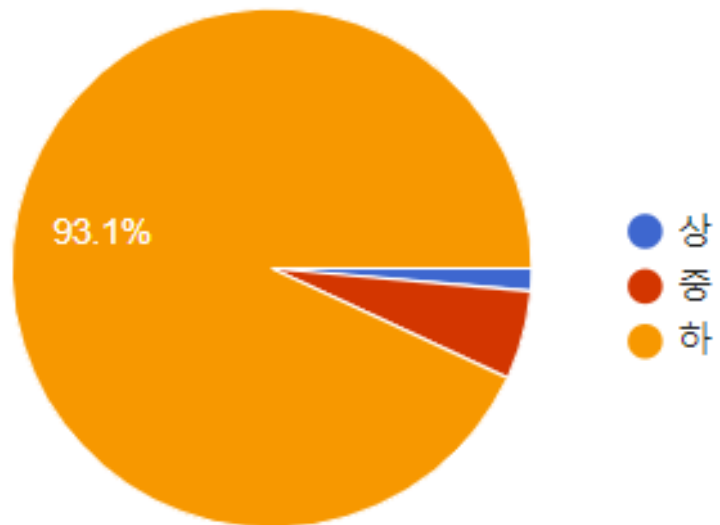
[1학기의 수업 다시보기]

#반복문 : while

3월

3. 반복문(while, for statement)

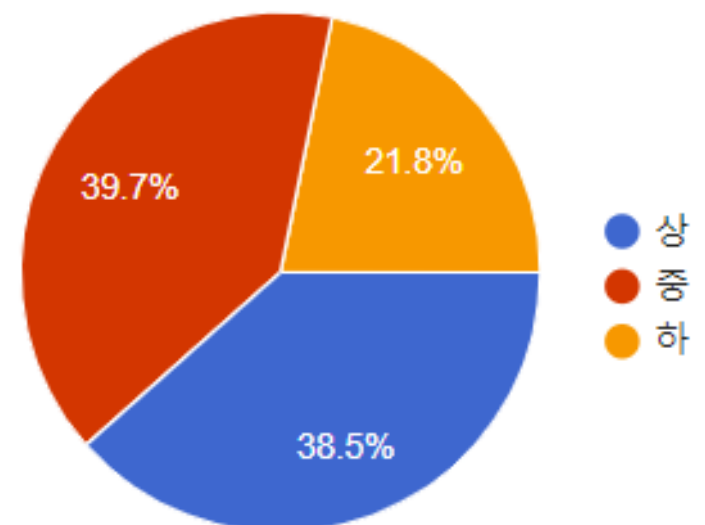
응답 144개



7월

5. 반복문(while문)

응답 78개



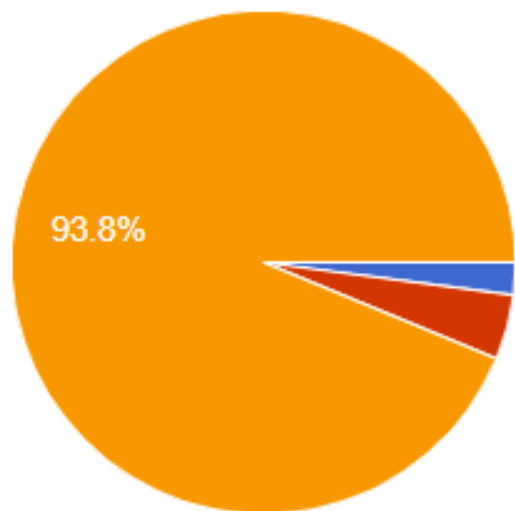
[1학기의 수업 다시보기]

#함수 작성 : def

3월

4. 함수 작성하기

응답 144개

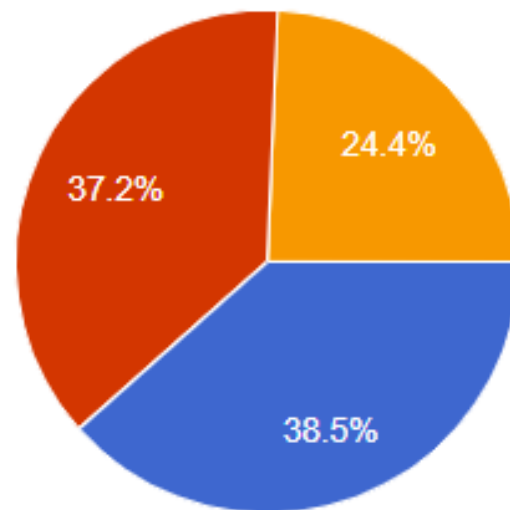


● 상
● 중
● 하

7월

6. 함수 작성하기

응답 78개



● 상
● 중
● 하



[Hello, Python Programming]

[2018 Top Programming Languages]

Language Types (click to hide)



Web



Mobile



Enterprise



Embedded

Language Rank

Types

Spectrum Ranking

1. Python



100.0

2. C++



98.4

3. C



98.2

4. Java



97.5

5. C#



89.8

6. PHP



85.4

7. R



83.3

8. JavaScript



82.8

9. Go



76.7

10. Assembly



74.5

[왜 Python을 쓸까요?]



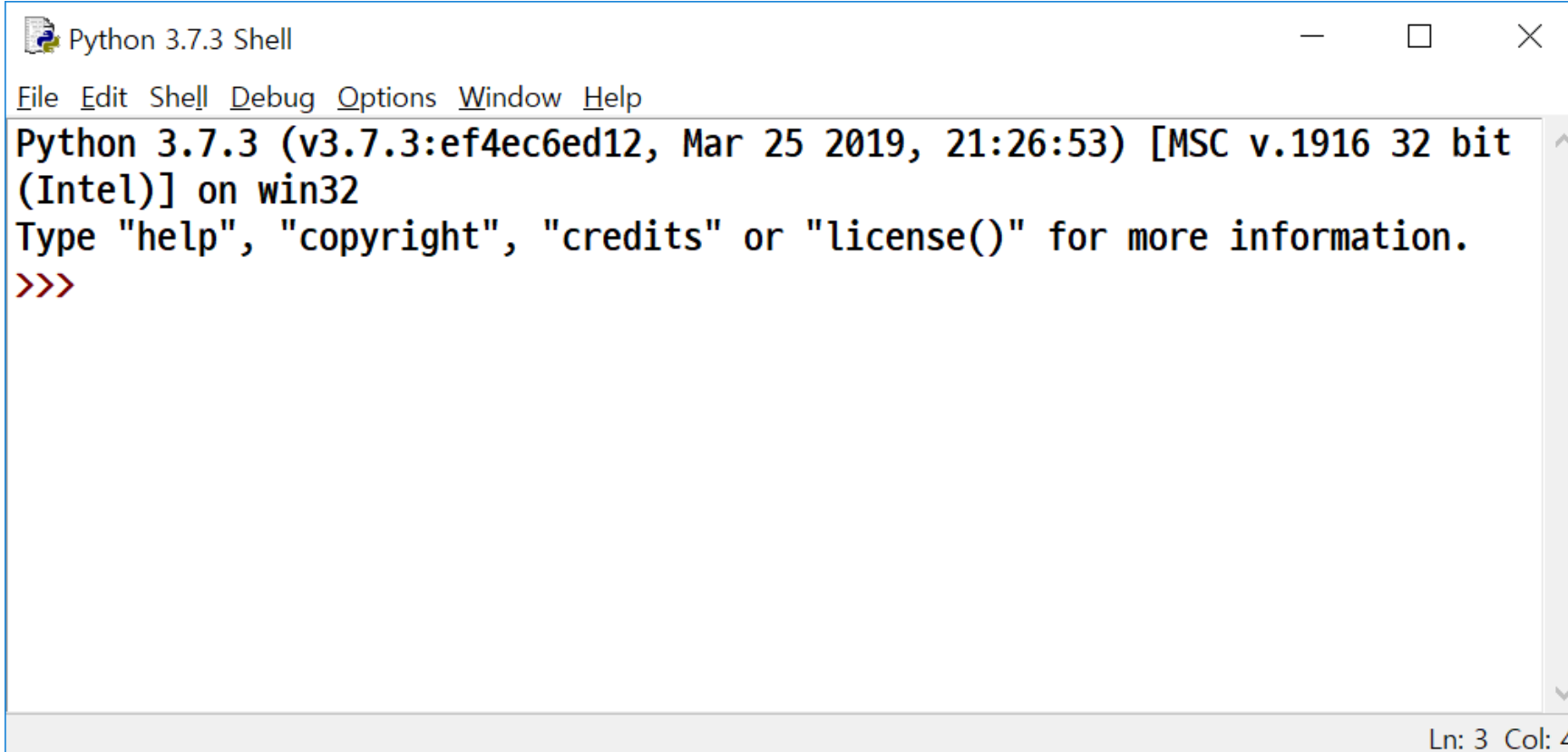
[Python IDLE 시작하기]

[시작]메뉴에서

[프로그램]-[**Python 3.X(XX-bit)**]를 선택



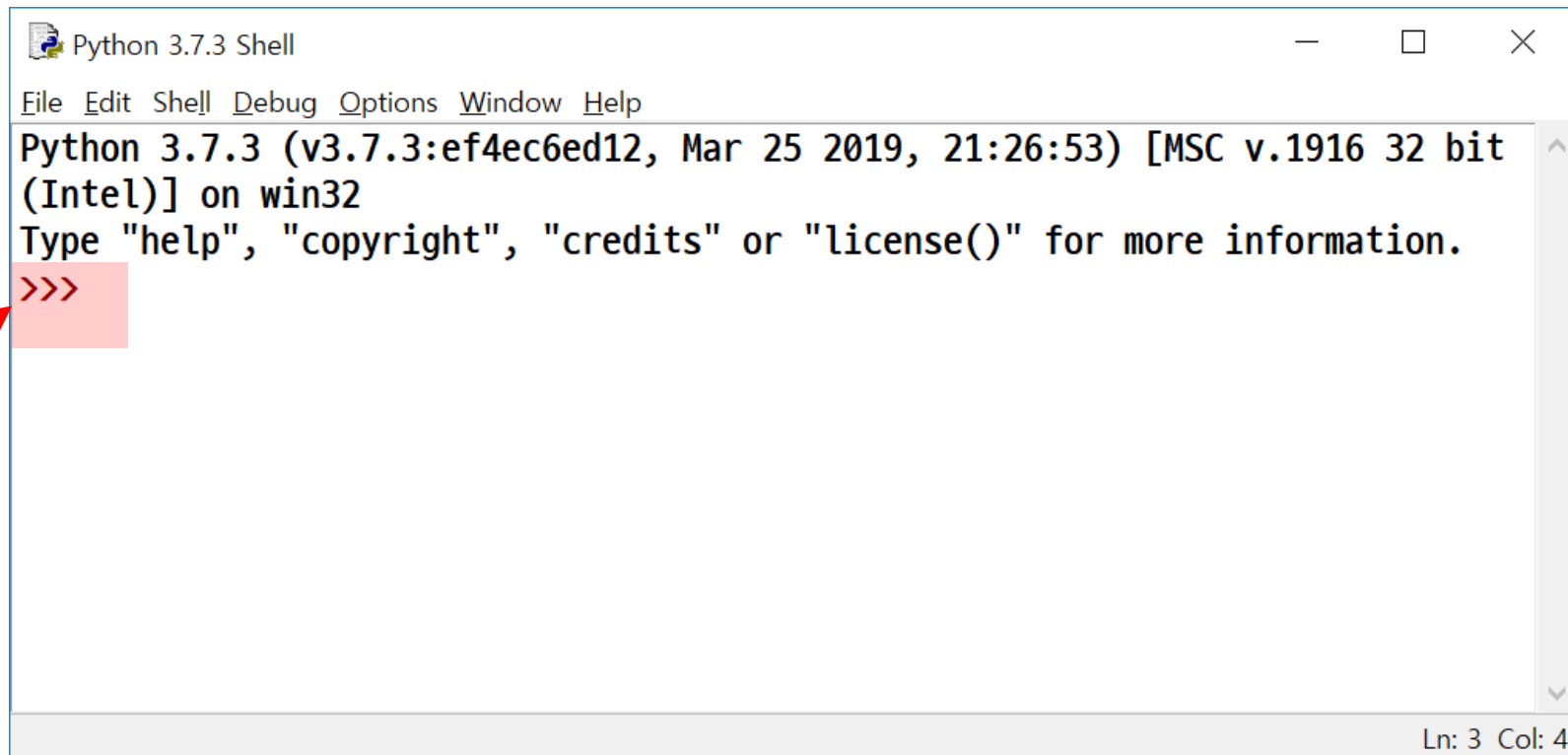
IDLE (Python
3.7 32-bit)

The screenshot shows the Python 3.7.3 Shell window. The title bar says "Python 3.7.3 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following information: "Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32", followed by "Type 'help', 'copyright', 'credits' or 'license()' for more information.", and the prompt ">>>". The status bar at the bottom right shows "Ln: 3 Col: 4".

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

[Python IDLE 시작하기]

이 것을
프롬프트라고
해요!

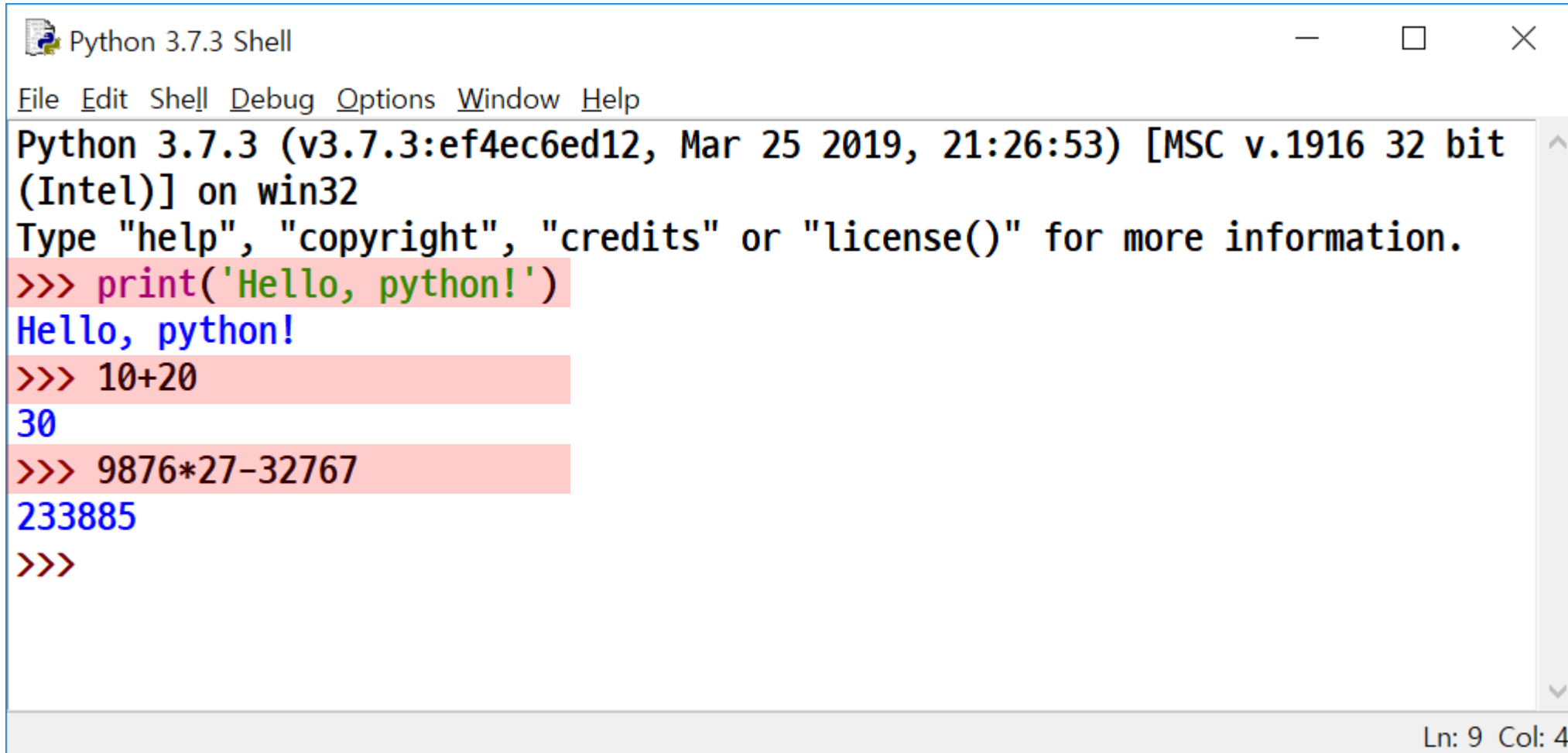
A screenshot of the Python 3.7.3 Shell window. The window has a title bar that says "Python 3.7.3 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the Python version and build information: "Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32". It also displays the prompt instructions: "Type 'help', 'copyright', 'credits' or 'license()' for more information." The prompt itself, ">>>", is highlighted with a red rectangular background. A red arrow points from the Korean text "이 것을 프롬프트라고 해요!" to this prompt. The status bar at the bottom right shows "Ln: 3 Col: 4".

```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Ln: 3 Col: 4

[Python IDLE시작하기]

프롬프트에 다양한 명령어를 입력해 보세요.

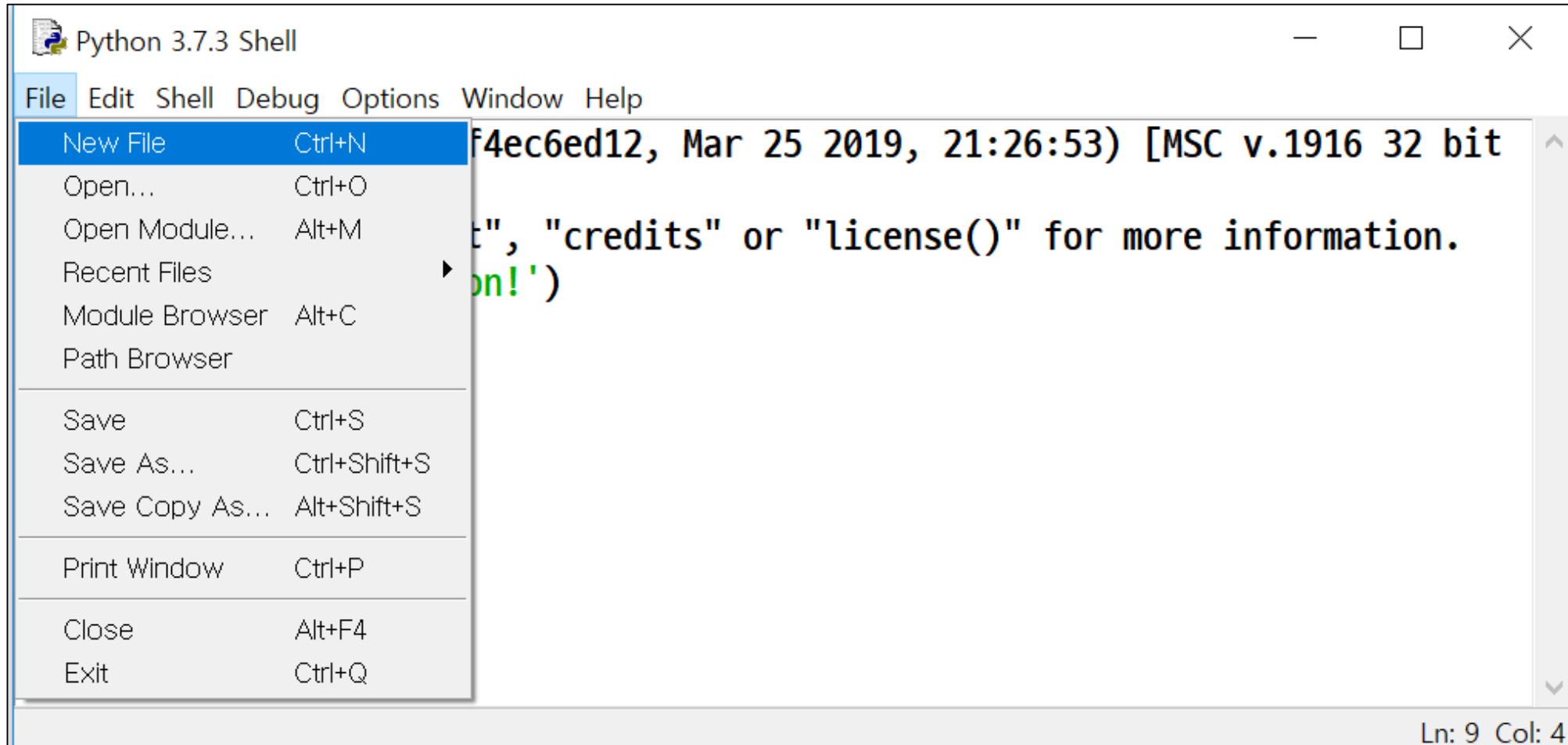


```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello, python!')
Hello, python!
>>> 10+20
30
>>> 9876*27-32767
233885
>>>
```

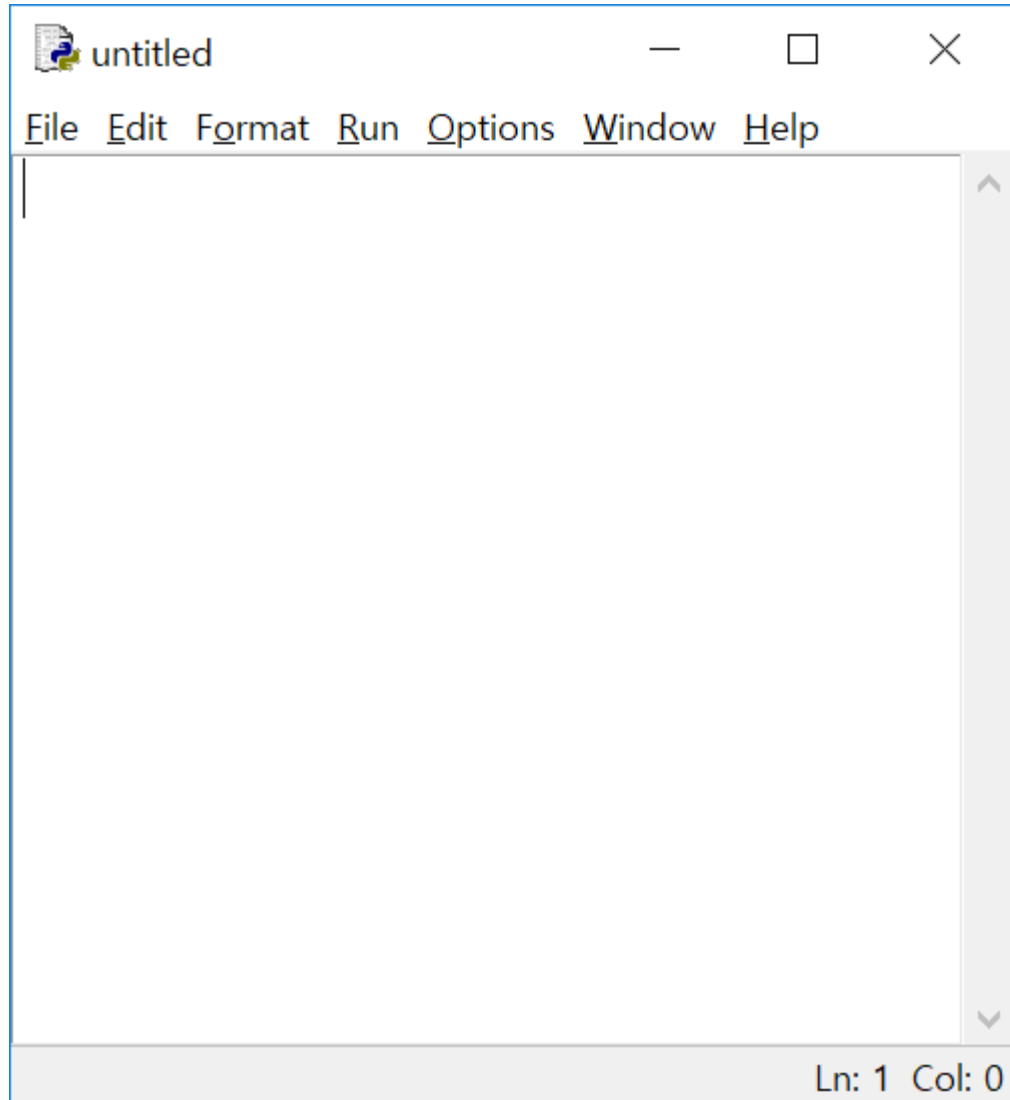
Ln: 9 Col: 4

[Python IDLE 여러 줄의 프로그램 작성하기]

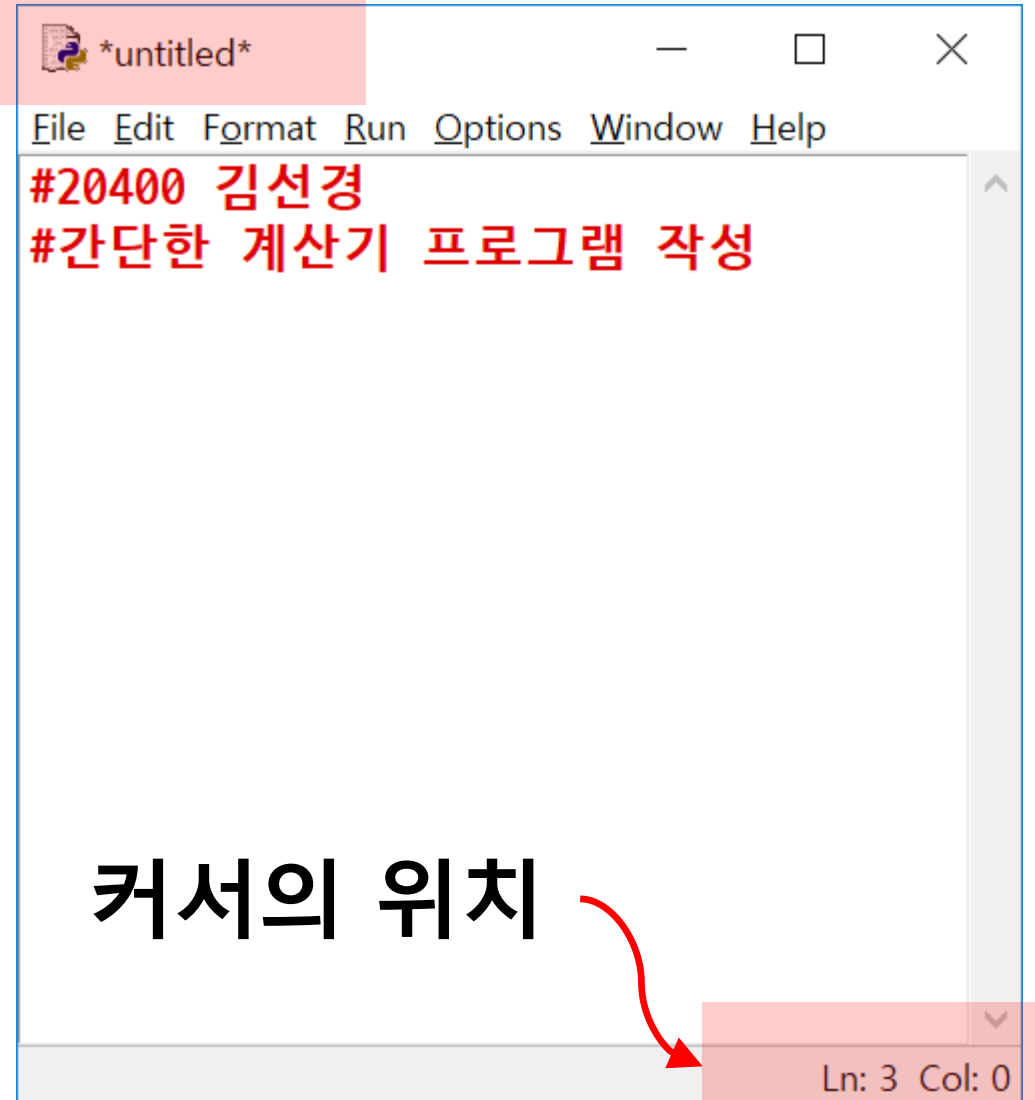
[File]-[New File]



[Python IDLE 여러 줄의 프로그램 작성하기]



파일명

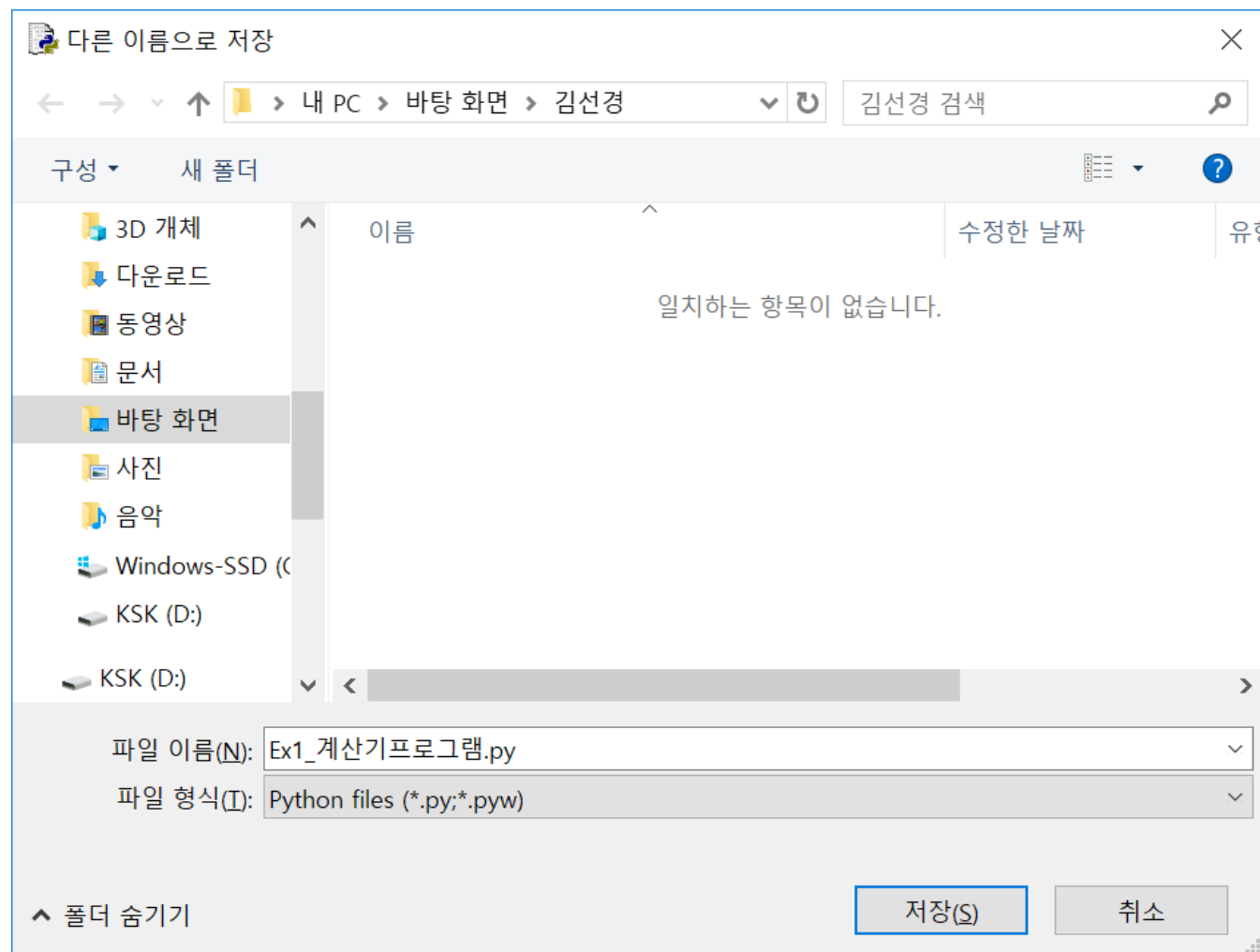
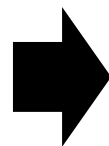
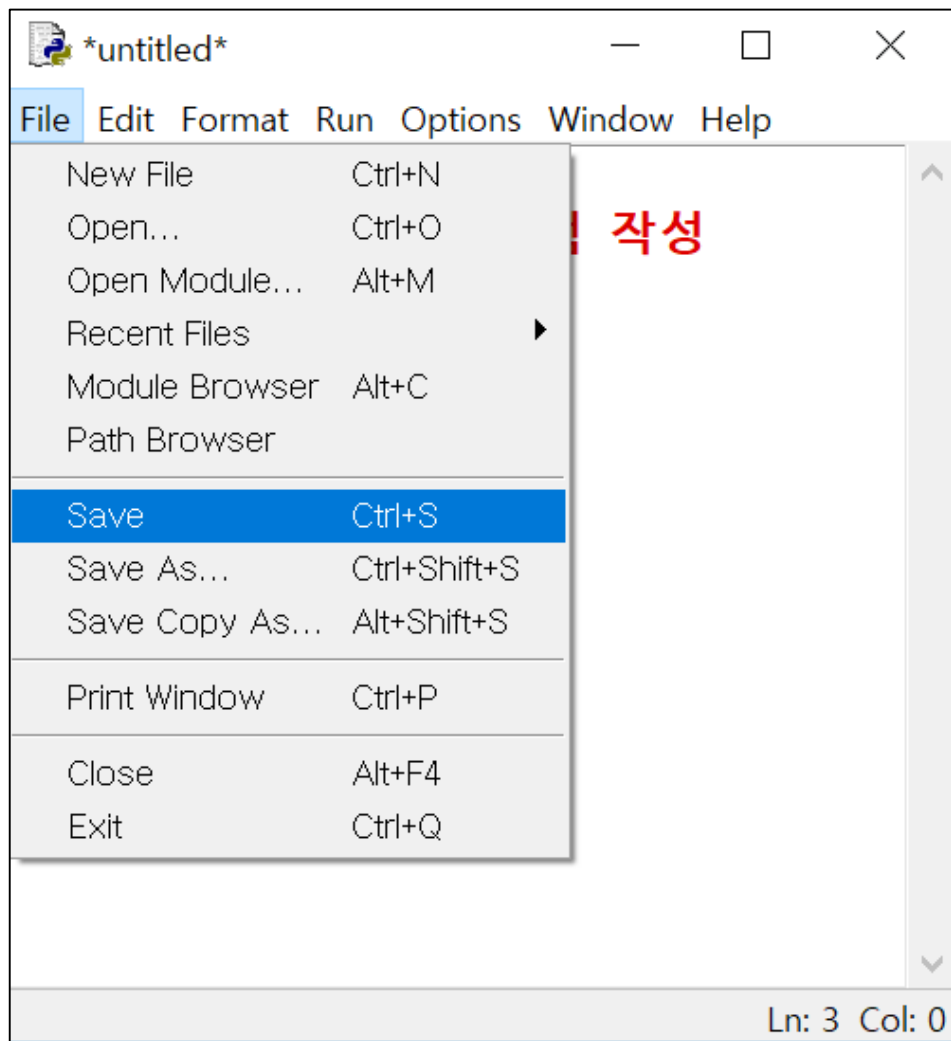


커서의 위치

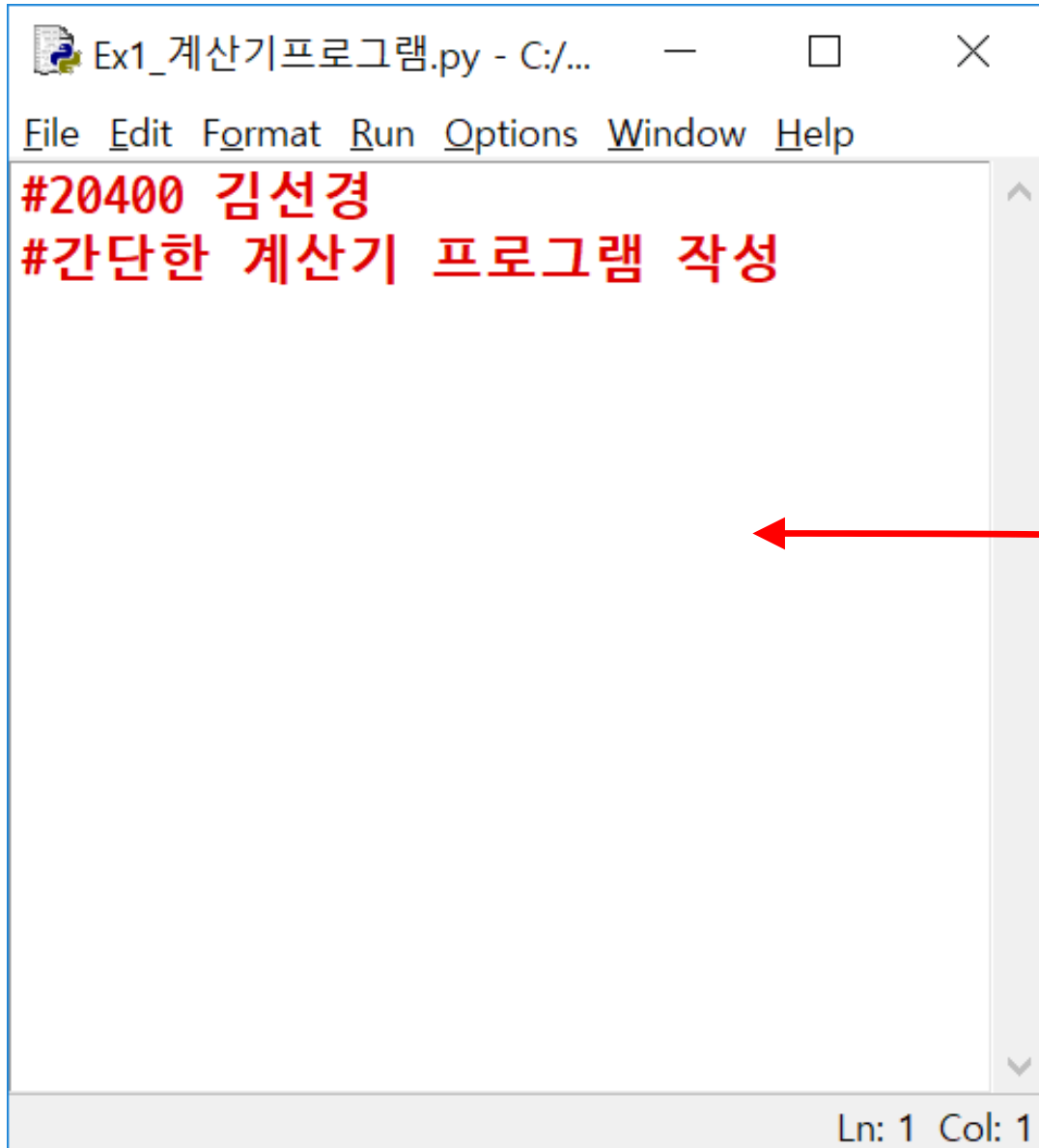
[Python IDLE 파일 저장]

[File]-[Save]

Ctrl + S



[Python IDLE 여러 줄의 프로그램 작성하기]



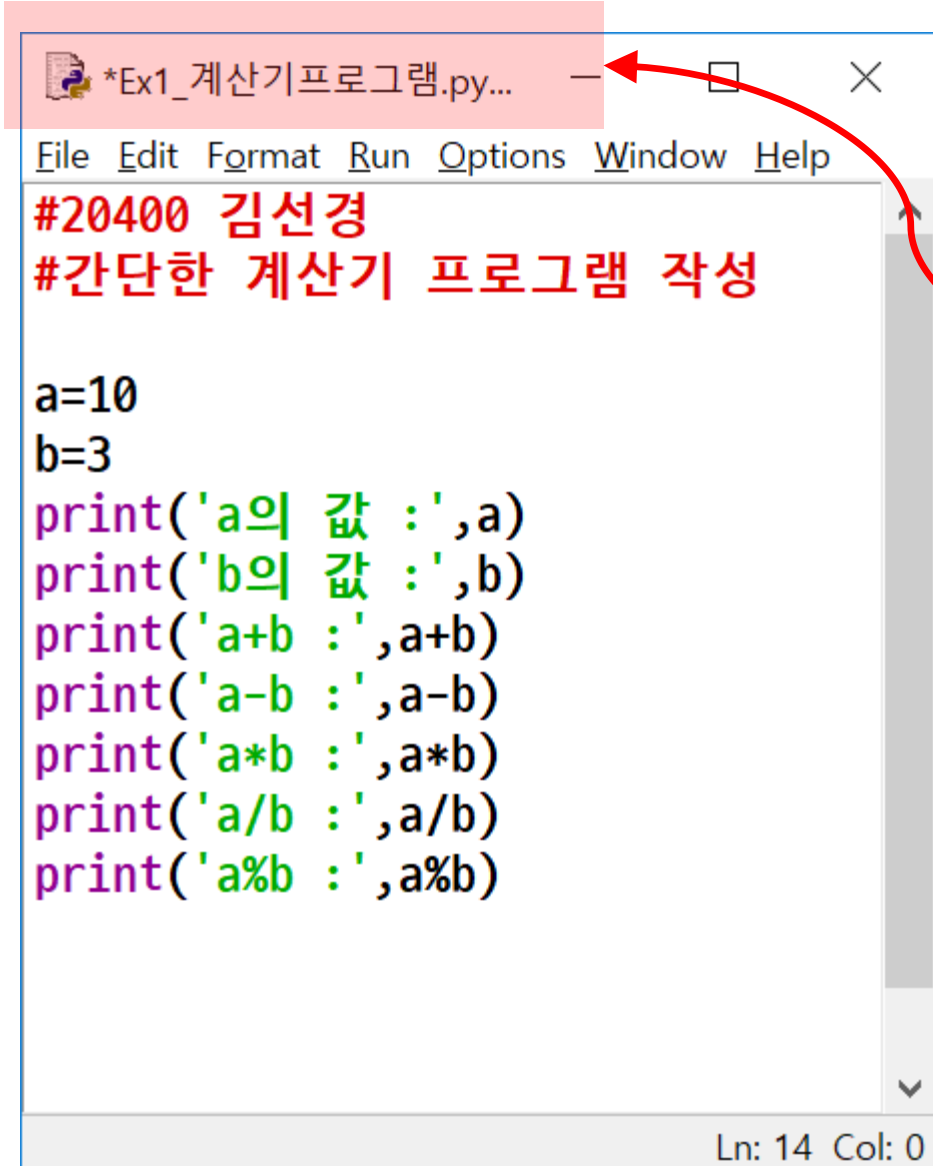
The image shows a screenshot of the Python IDLE environment. The window title is "Ex1_계산기프로그램.py - C:/...". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The main text area contains two lines of red text: "#20400 김선경" and "#간단한 계산기 프로그램 작성". A red arrow points from the text "이곳에 프로그램을 작성" to the text area. The status bar at the bottom right shows "Ln: 1 Col: 1".

```
#20400 김선경  
#간단한 계산기 프로그램 작성
```

Ln: 1 Col: 1

이곳에 프로그램을 작성

[Python IDLE 여러 줄의 프로그램 작성하기]



```
*Ex1_계산기프로그램.py...
File Edit Format Run Options Window Help
#20400 김선경
#간단한 계산기 프로그램 작성

a=10
b=3
print('a의 값 :',a)
print('b의 값 :',b)
print('a+b :',a+b)
print('a-b :',a-b)
print('a*b :',a*b)
print('a/b :',a/b)
print('a%b :',a%b)

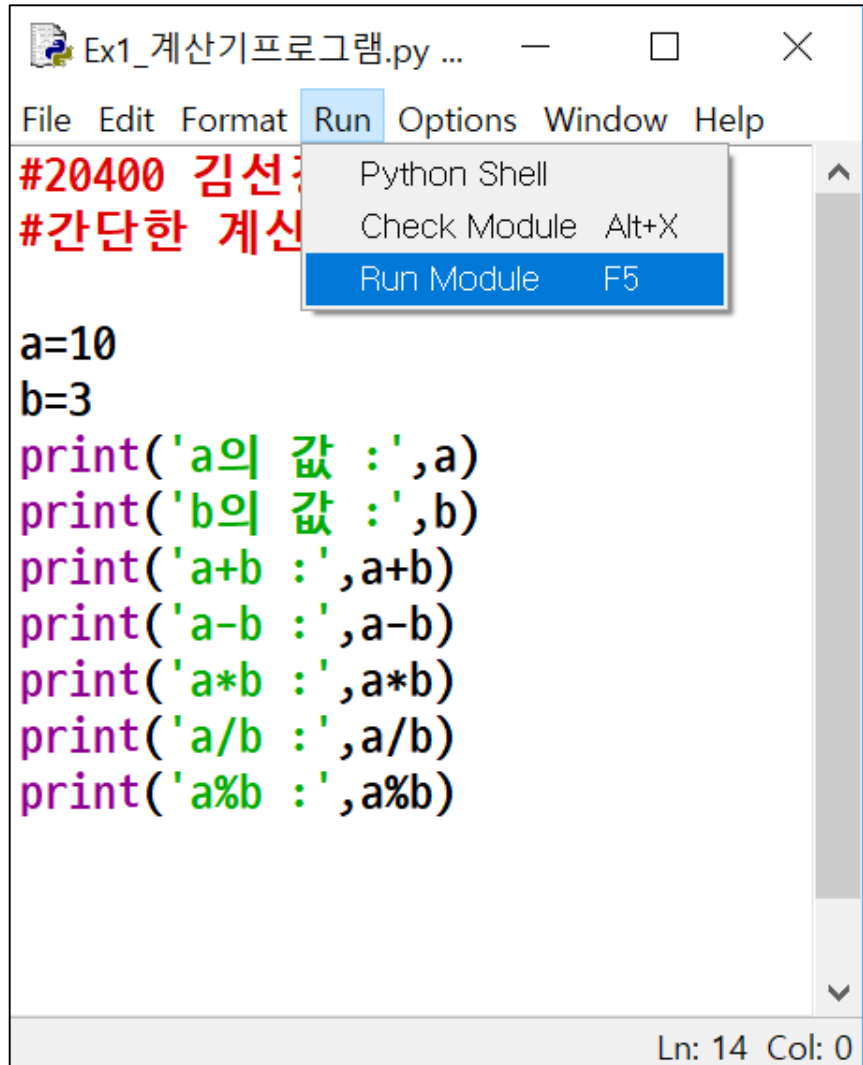
Ln: 14 Col: 0
```

파일에 변화가 생기면
파일명 앞에 * 이 표시되요!

[Python IDLE 작성한 프로그램 실행하기]

[Run]-[Run Module]

F5



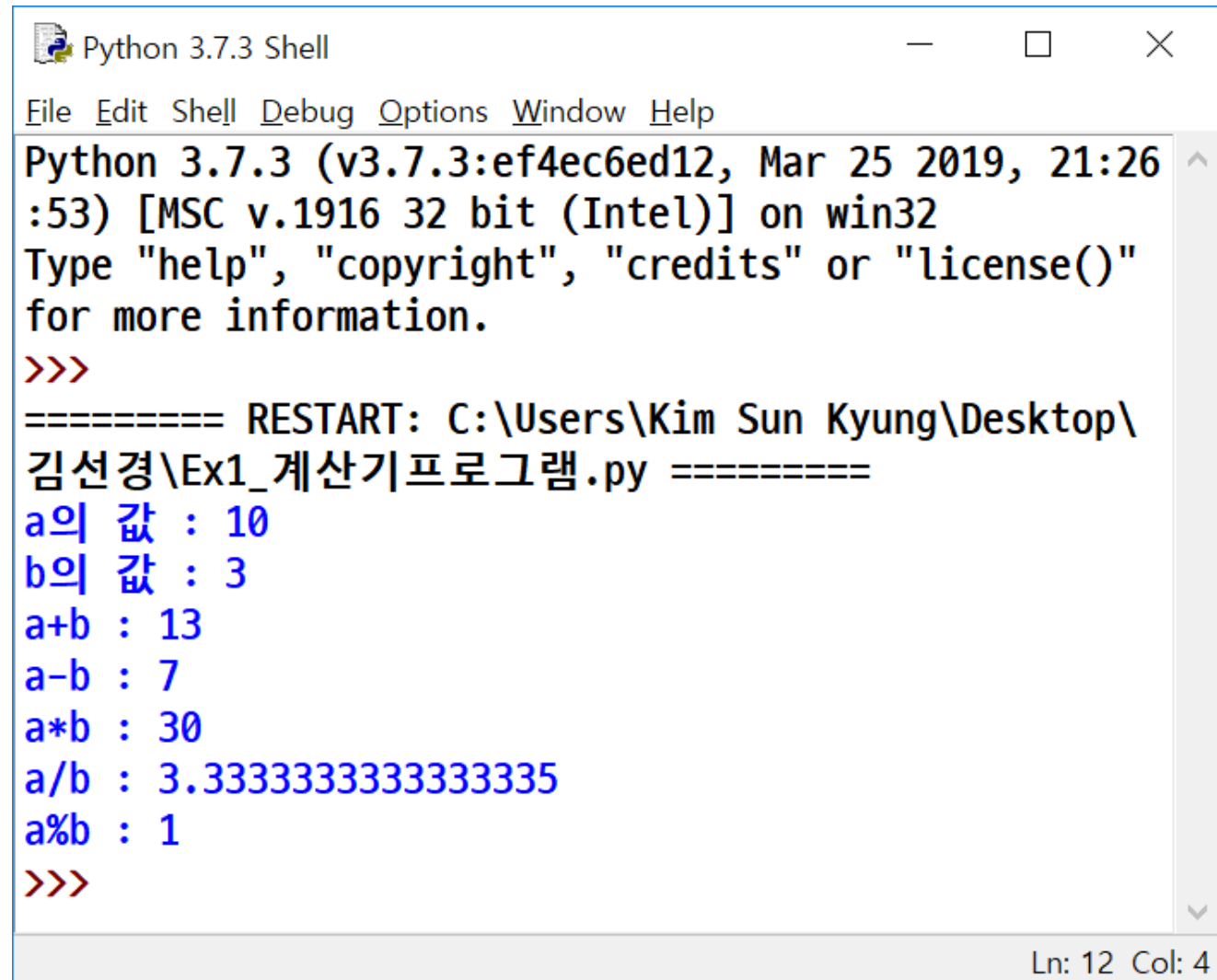
Ex1_계산기프로그램.py ...

File Edit Format Run Options Window Help

#20400 김선경
#간단한 계산기

```
a=10  
b=3  
print('a의 값 :',a)  
print('b의 값 :',b)  
print('a+b :',a+b)  
print('a-b :',a-b)  
print('a*b :',a*b)  
print('a/b :',a/b)  
print('a%b :',a%b)
```

Ln: 14 Col: 0



Python 3.7.3 Shell

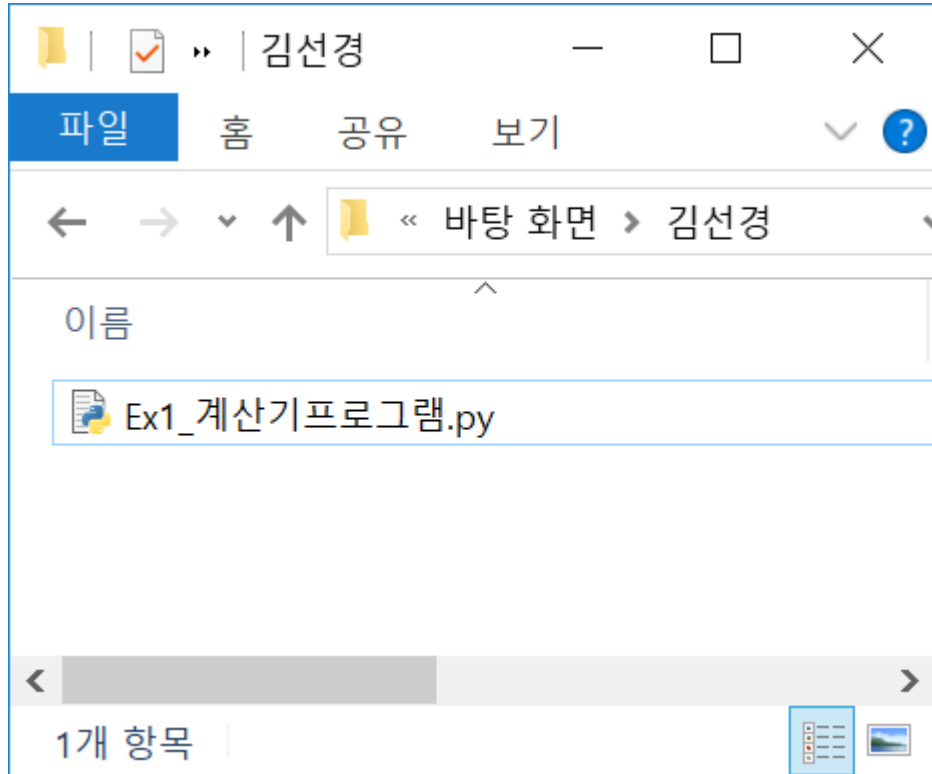
File Edit Shell Debug Options Window Help

Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

```
>>>  
===== RESTART: C:\Users\Kim Sun Kyung\Desktop\김선경\Ex1_계산기프로그램.py =====  
a의 값 : 10  
b의 값 : 3  
a+b : 13  
a-b : 7  
a*b : 30  
a/b : 3.3333333333333335  
a%b : 1  
>>>
```

Ln: 12 Col: 4

[Python IDLE 작성한 프로그램 확인하기]

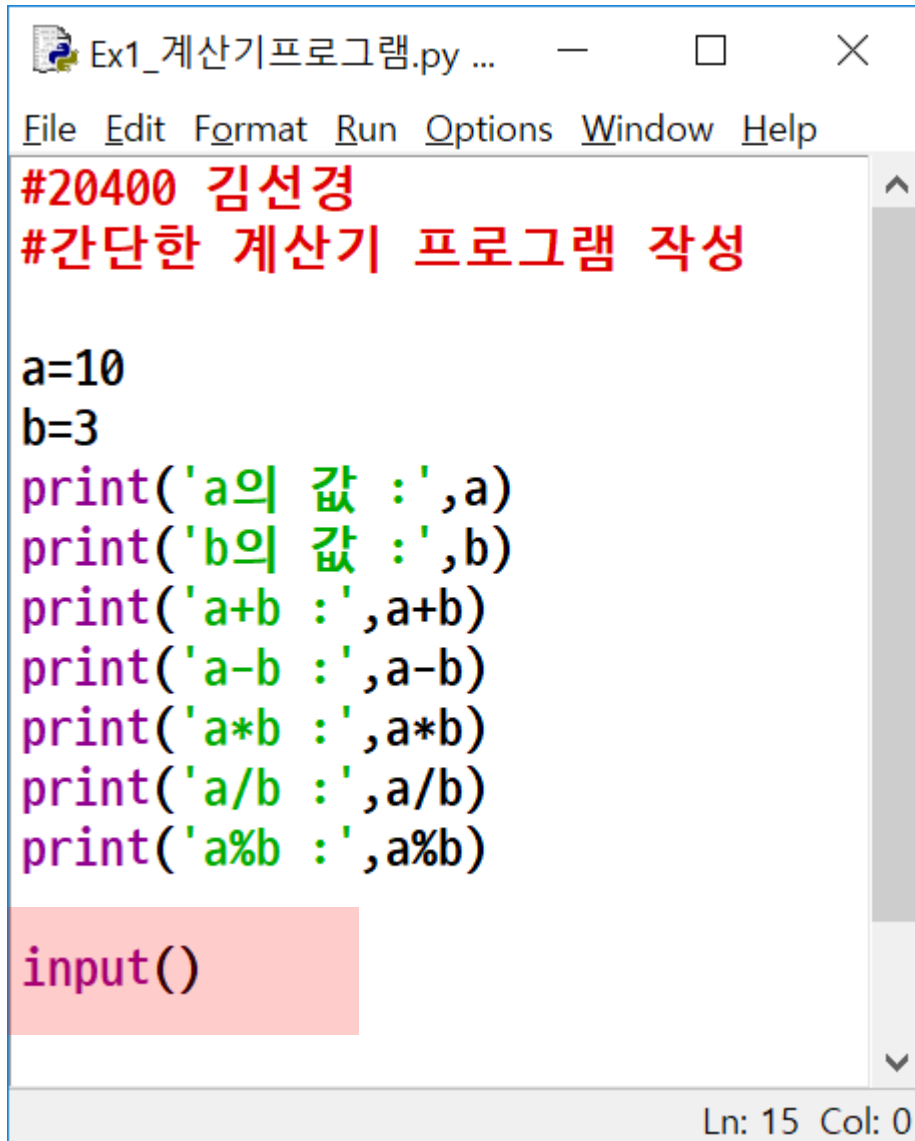


1. 파이썬 파일의 확장자는 **.py**

2. 이 프로그램을 **더블클릭하여 실행**하면?

실행 창이 그냥 꺼져서
내용을 확인할 수가 없어요ㅠㅠ

[Python IDLE 작성한 프로그램 확인하기]



The image shows a Python IDLE window titled 'Ex1_계산기프로그램.py ...'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The script content is as follows:

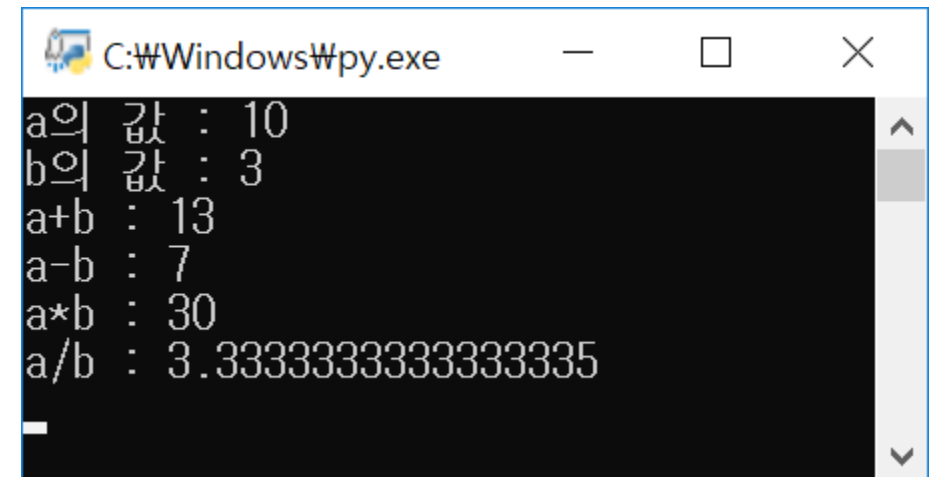
```
#20400 김선경
#간단한 계산기 프로그램 작성

a=10
b=3
print('a의 값 :',a)
print('b의 값 :',b)
print('a+b :',a+b)
print('a-b :',a-b)
print('a*b :',a*b)
print('a/b :',a/b)
print('a%b :',a%b)

input()
```

The status bar at the bottom indicates 'Ln: 15 Col: 0'.

1. 프로그램의 **마지막 줄에 input()을 추가**
2. 이 프로그램을 **더블클릭하여 실행**하면?



The image shows the output of the program in a window titled 'C:\Windows\py.exe'. The output is as follows:

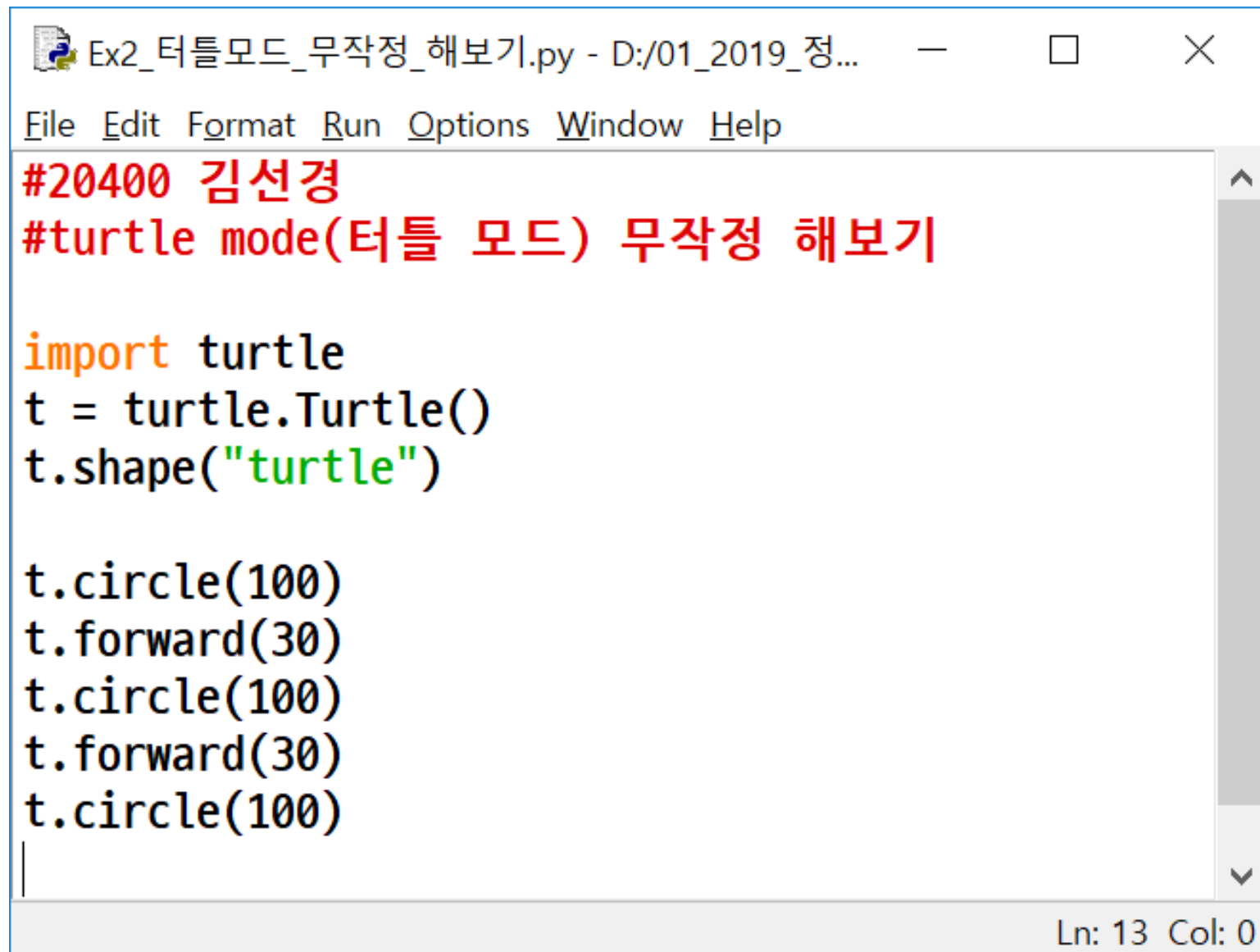
```
a의 값 : 10
b의 값 : 3
a+b : 13
a-b : 7
a*b : 30
a/b : 3.3333333333333335
_
```

<https://docs.python.org/3/library/turtle.html>

 **[파이썬 거북이 모듈(Python Turtle Module)]**

[파이썬 거북이 모듈 - 무작정 해보기!]

#1. 코드 작성



The image shows a screenshot of a Python IDE window. The title bar reads "Ex2_터틀모드_무작정_해보기.py - D:/01_2019_정...". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
#20400 김선경
#turtle mode(터틀 모드) 무작정 해보기

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

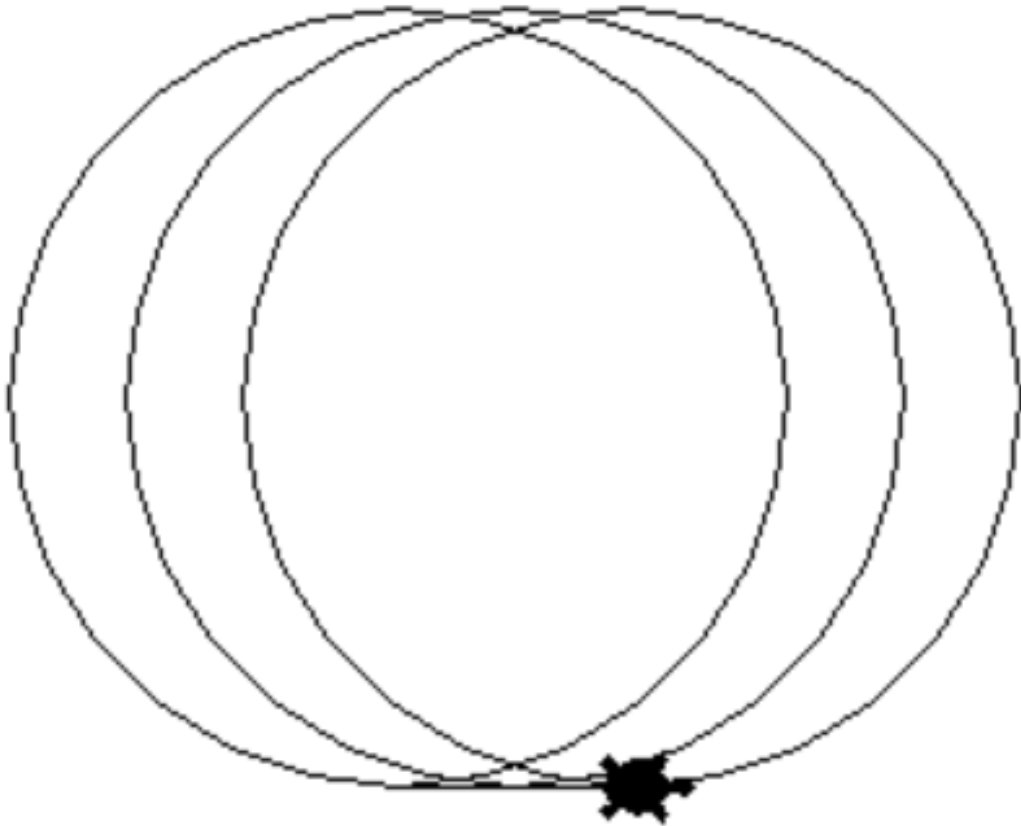
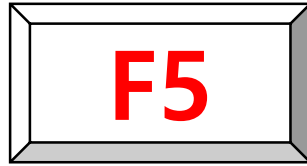
t.circle(100)
t.forward(30)
t.circle(100)
t.forward(30)
t.circle(100)
|
```

The status bar at the bottom right indicates "Ln: 13 Col: 0".

[파이썬 거북이 모듈 - 무작정 해보기!]

#2. 실행하기

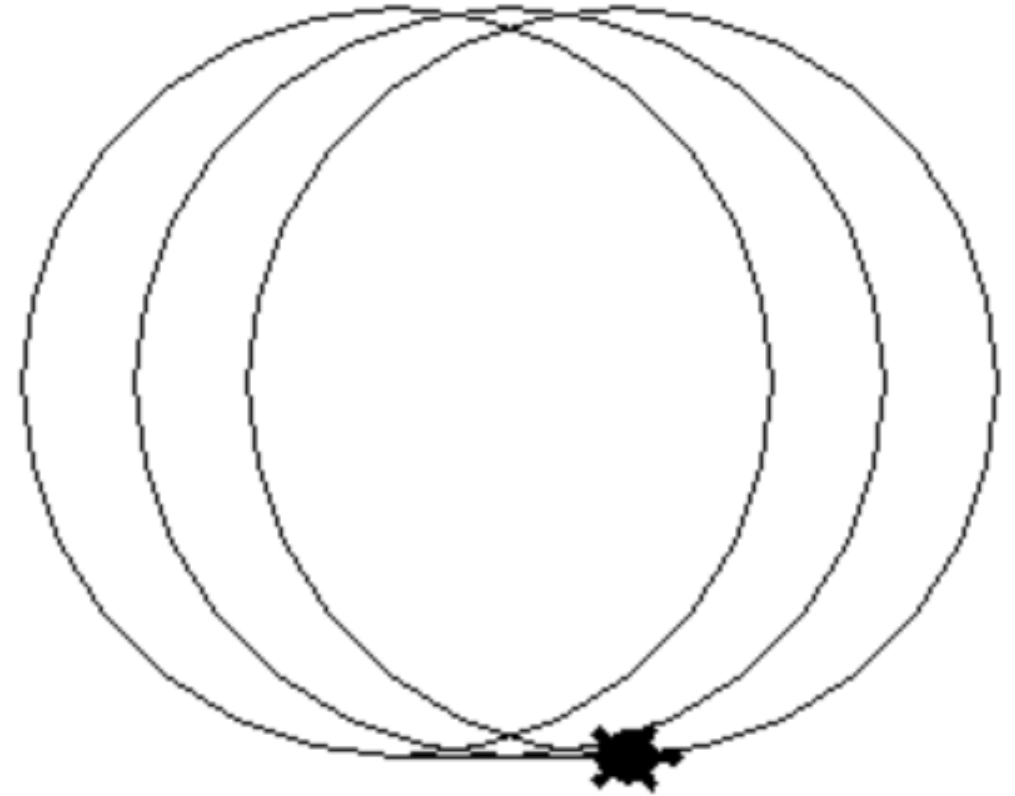
[Run]-[Run Module]



[파이썬 거북이 모듈 - 무작정 해보기!]

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

```
t.circle(100)  
t.forward(30)  
t.circle(100)  
t.forward(30)  
t.circle(100)
```



[파이썬 거북이 모듈 - 무작정 해보기!]

```
import turtle #turtle 그래픽 모듈을 불러온다  
t = turtle.Turtle() #Turtle을 1개 만들고 변수 t에 저장한다  
t.shape("turtle") #t의 모양을 거북이로 한다
```

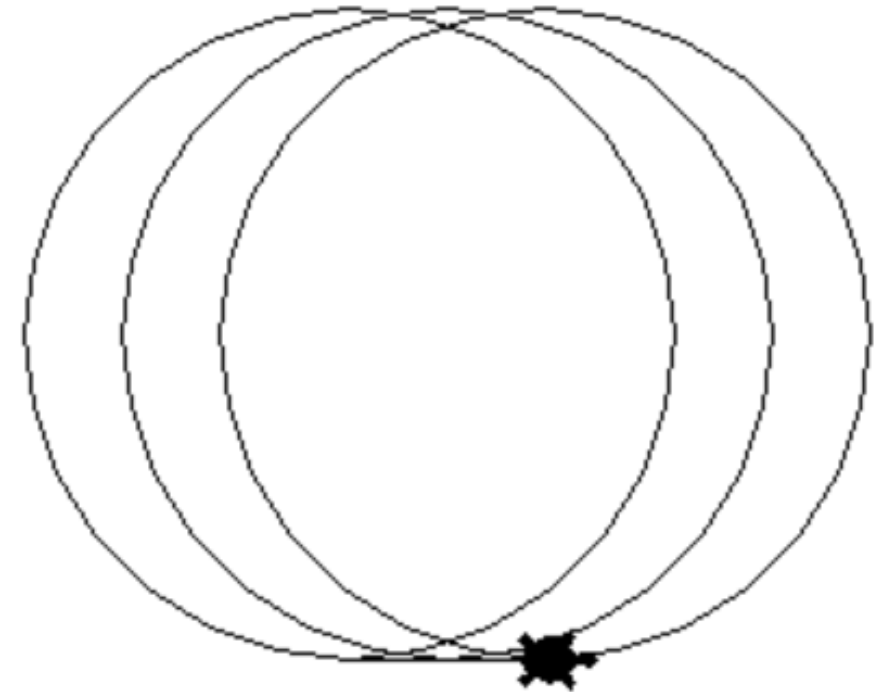
```
t.circle(100) # A
```

```
t.forward(30) # B
```

```
t.circle(100)
```

```
t.forward(30)
```

```
t.circle(100)
```



[파이썬 거북이 모듈 - 무작정 해보기!]



표준 라이브러리 문서(레퍼런스)를 보는 것은
매우매우 중요해요!!!

<https://docs.python.org/3/library/turtle.html>

Turtle methods ¶

Turtle motion

Move and draw

`forward()` | `fd()`

`backward()` | `bk()` | `back()`

`right()` | `rt()`

`left()` | `lt()`

`goto()` | `setpos()` | `setposition()`

`setx()`

`sety()`

`setheading()` | `seth()`

`home()`

`circle()`

`dot()`

`stamp()`

`clearstamp()`

`clearstamps()`

`undo()`

`speed()`

[파이썬 거북이 모듈 - 무작정 해보기!]

```
import turtle #turtle 그래픽 모듈을 불러온다  
t = turtle.Turtle() #Turtle을 1개 만들고 변수 t에 저장한다  
t.shape("turtle") #t의 모양을 거북이로 한다
```

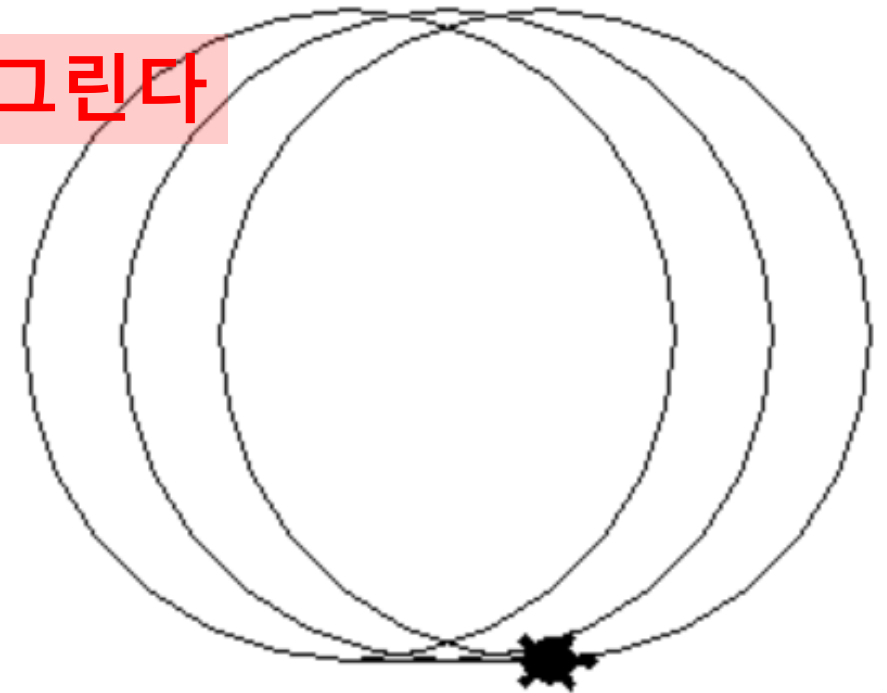
```
t.circle(100) # 반지름이 100인 원을 그린다
```

```
t.forward(30) # 30만큼 앞으로 간다
```

```
t.circle(100)
```

```
t.forward(30)
```

```
t.circle(100)
```



[갑자기 반지름을 모두 50으로 바꿔야 한다면?]

[파이썬 거북이 모듈 - 무작정 해보기!]

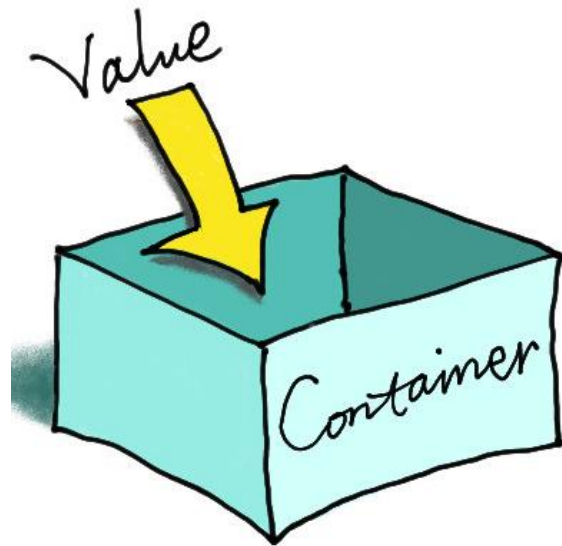
```
import turtle  
t = turtle.Turtle()  
t.shape("turtle")
```

```
r=50    # 변수 r에 50을 넣는다
```

```
t.circle(r)  
t.forward(30)  
t.circle(r)  
t.forward(30)  
t.circle(r)
```

[RUR-PLE을 아련히 떠올려 봅시다!]

[변수] : 특정 숫자나 글자 등을 저장할 수 있는 기억 공간

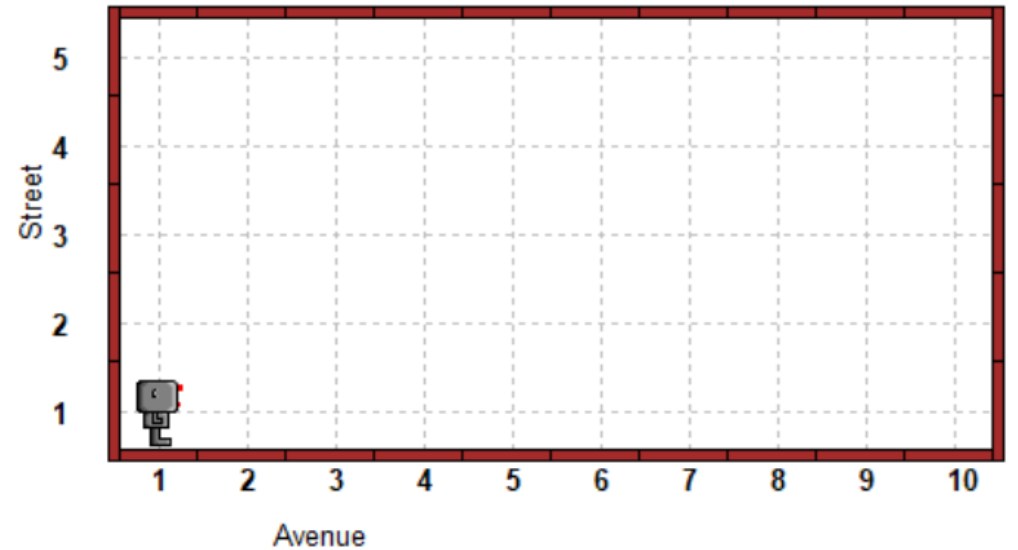


x = 10

x에 10을 넣는다.

[RUR-PLE을 아련히 떠올려 봅시다!]

```
steps=0
def move_and_step_count():
    ... move()
    ... steps=steps+1
#program start
repeat(move_and_step_count, 9)
print(steps)
turn_off()
```



[갑자기 이동 거리를 모두 100으로 바꿔야 한다면?]

[파이썬 거북이 모듈 - 무작정 해보기!]

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

```
r=50
```

```
d=100
```

```
t.circle(r)
```

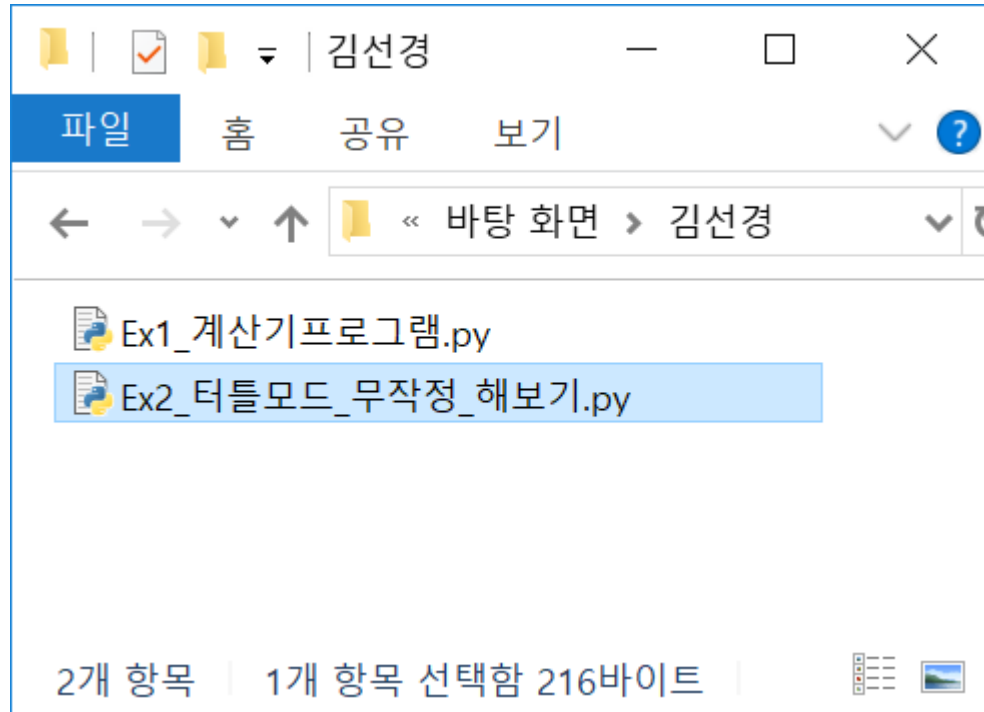
```
t.forward(d)
```

```
t.circle(r)
```

```
t.forward(d)
```

```
t.circle(r)
```

[파이썬 거북이 모듈 - 파일 확인하기]

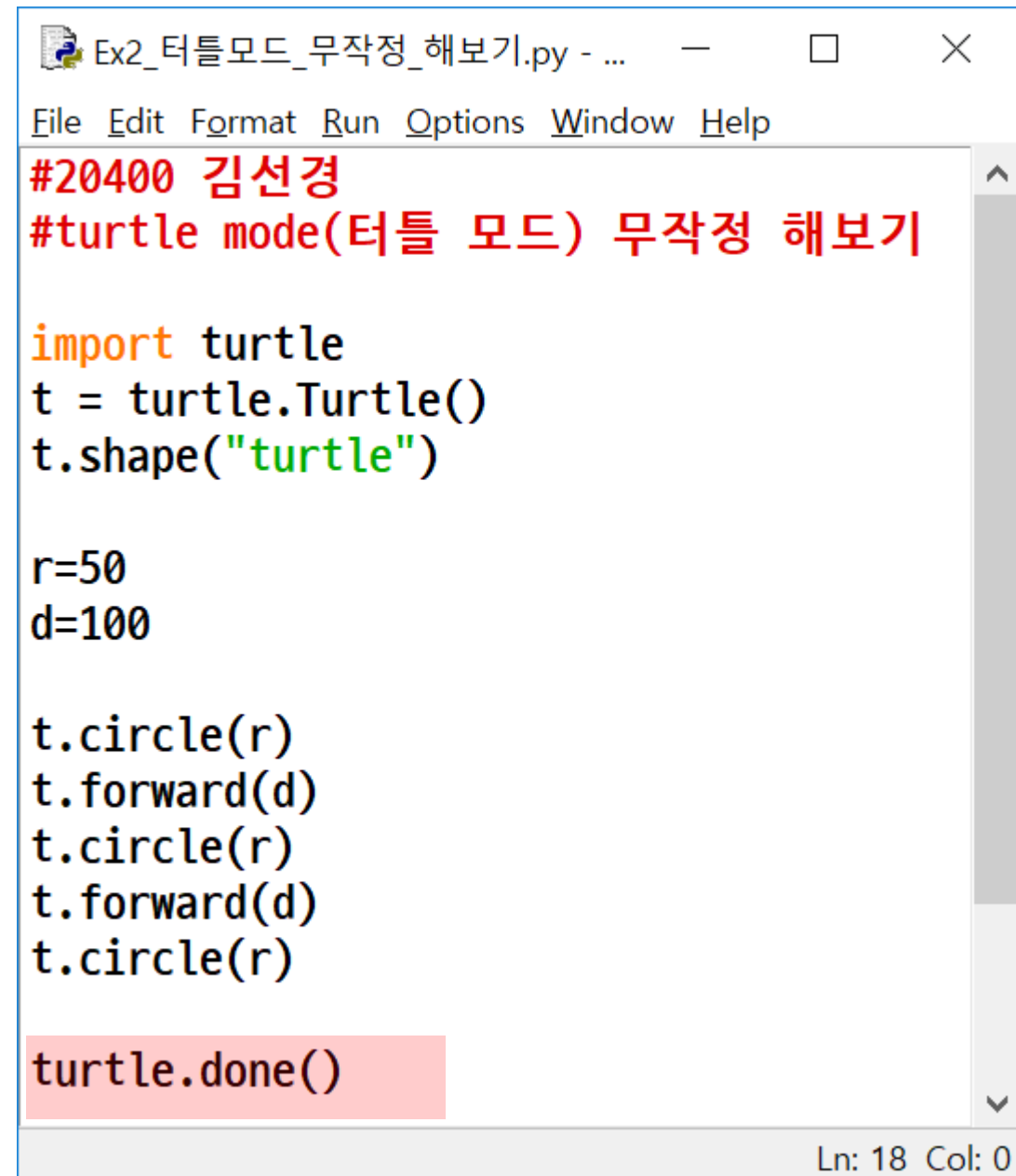


1. 파이썬 파일의 확장자는 **.py**

2. 이 프로그램을 **더블클릭하여 실행**하면?

거북이가 프로그램을 실행한 후
창이 그냥 꺼져요ㅠㅠㅠ

[파이썬 거북이 모듈 - 파일 확인하기]



The image shows a screenshot of a Python IDE window titled "Ex2_터틀모드_무작정_해보기.py - ...". The window contains a Python script that uses the turtle module to draw a series of circles. The script is as follows:

```
#20400 김선경
#turtle mode(터틀 모드) 무작정 해보기

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

r=50
d=100

t.circle(r)
t.forward(d)
t.circle(r)
t.forward(d)
t.circle(r)

turtle.done()
```

The status bar at the bottom of the window indicates "Ln: 18 Col: 0".

1. 프로그램의 마지막 줄에 **turtle.done()**
2. 이 프로그램을 **더블클릭하여 실행**하면?

거북이가 프로그램을 실행한 후
창이 그대로 유지되요 :)



turtle 모듈 정리!

#처음 세줄, #마지막 한 줄, #당분간 공식처럼 사용

```
import turtle #turtle모듈을 불러온다  
t=turtle.Turtle() #Turtle을 1개 만들고 변수 t에 저장한다.  
t.shape("turtle") #t의 모양을 거북이로 한다.
```

#여기에 프로그래밍

```
turtle.done() #실행 후 창이 꺼지지 않도록 프로그램의 마지막 줄에 작성한다.
```



turtle 모듈 정리!

#처음 세줄, #마지막 한 줄, #당분간 공식처럼 사용

```
import turtle #turtle모듈을 불러온다  
t=turtle.Turtle() #Turtle을 1개 만들고 변수 t에 저장한다.  
t.shape("turtle") #t의 모양을 거북이로 한다.
```

#여기에 프로그래밍

프로그래밍 무엇?

```
turtle.done() #실행 후 창이 꺼지지 않도록 프로그램의 마지막 줄에 작성한다.
```



turtle 모듈 정리!

#처음 세줄, #마지막 한 줄, #당분간 공식처럼 사용

```
import turtle #turtle모듈을 불러온다  
t=turtle.Turtle() #Turtle을 1개 만들고 변수 t에 저장한다.  
t.shape("turtle") #t의 모양을 거북이로 한다.
```

#여기에 프로그래밍

<https://docs.python.org/3/library/turtle.html>

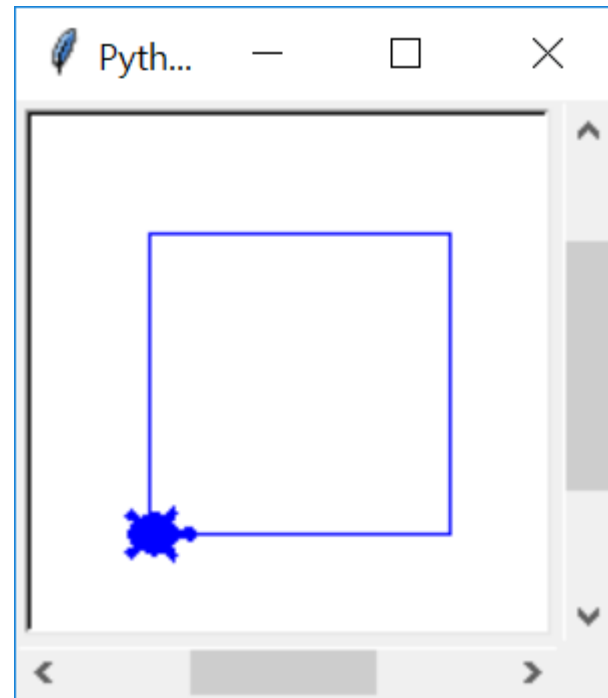
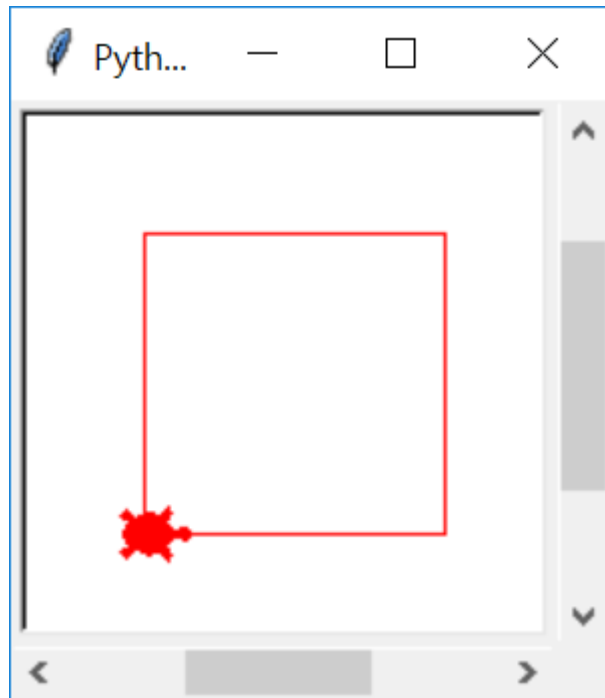
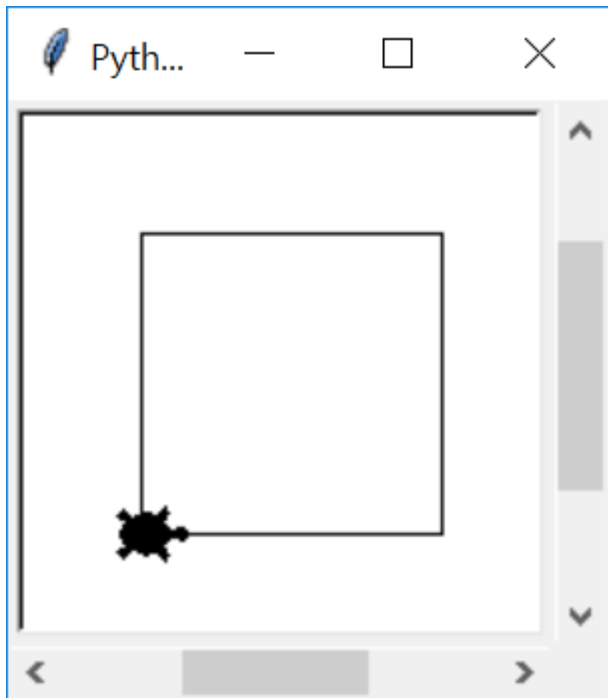
찾아보며 코드 작성

```
turtle.done() #실행 후 창이 꺼지지 않도록 프로그램의 마지막 줄에 작성한다.
```



Quiz1.

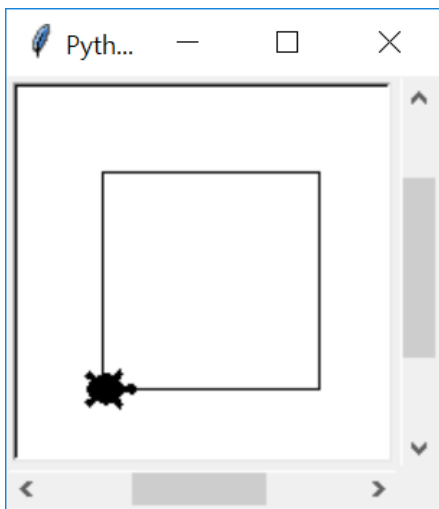
사각형을 그리는 거북이를 완성해 보세요!





Quiz1 : Hint

사각형을 그리는 거북이를 완성해 보세요!



```
import turtle #turtle모듈을 불러온다  
t=turtle.Turtle() #Turtle을 1개 만들고 변수 t에 저장한다.  
t.shape("turtle") #t의 모양을 거북이로 한다.
```

#여기에 프로그래밍

#1. t를 앞으로 100만큼 이동한다.

#2. t를 왼쪽으로 90도 회전한다.

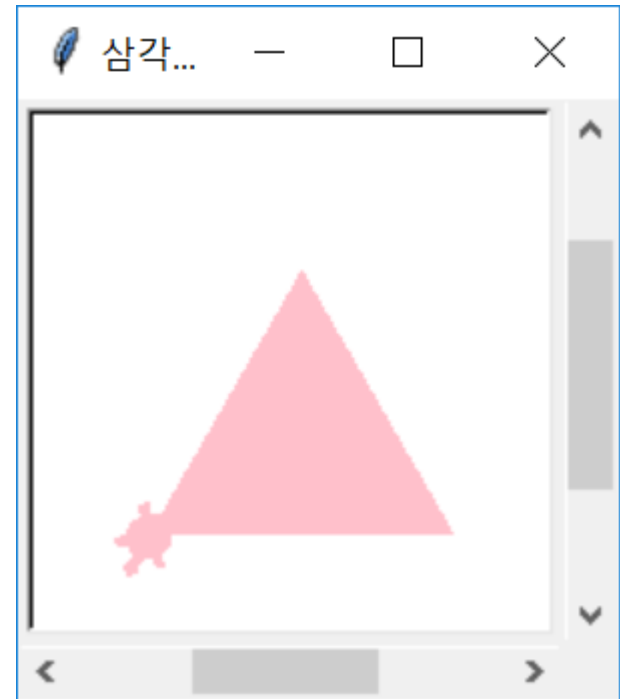
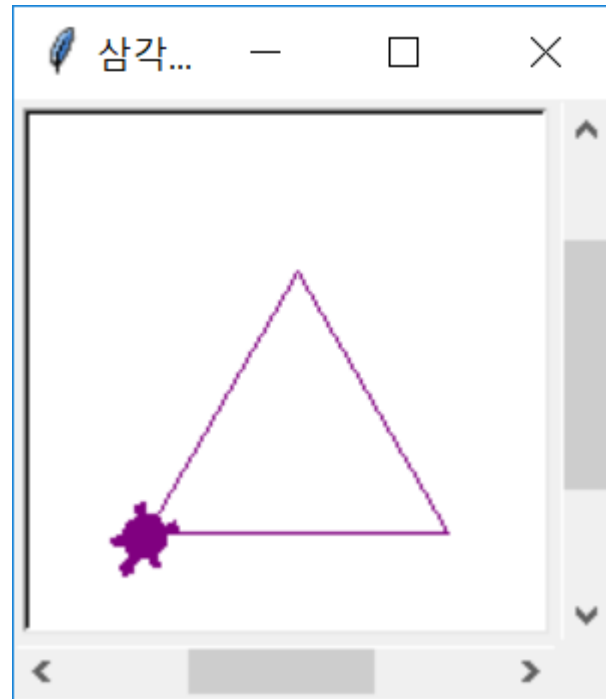
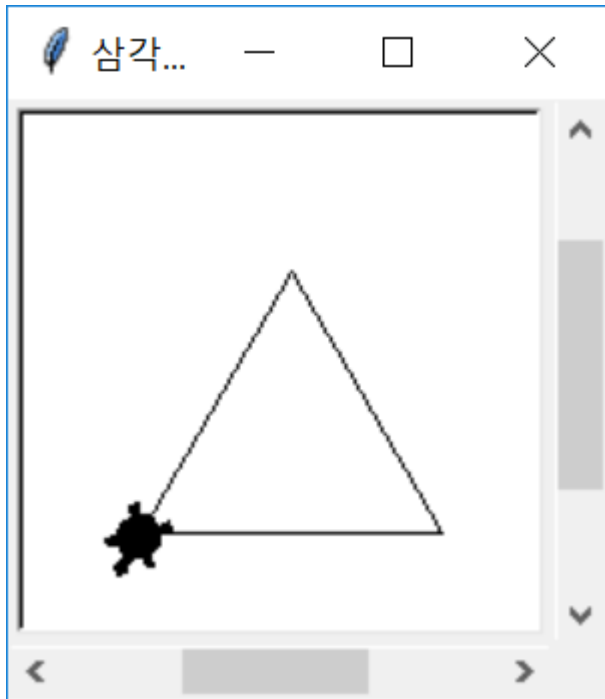
#1과 2의 과정을 3번 더 한다.

```
turtle.done() #실행 후 창이 꺼지지 않도록 프로그램의 마지막 줄에 작성한다.
```



Quiz2.

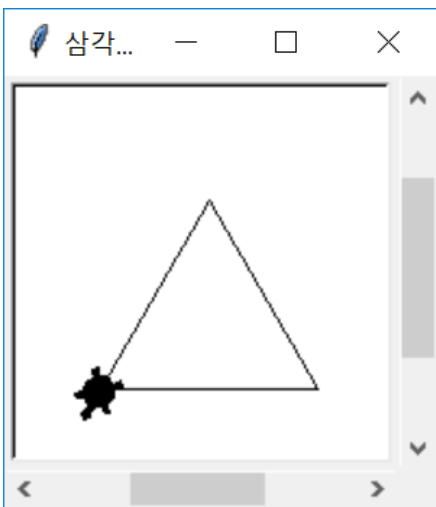
삼각형을 그리는 거북이를 완성해 보세요!





Quiz2 : Hint

삼각형을 그리는 거북이를 완성해 보세요!



```
import turtle #turtle모듈을 불러온다  
t=turtle.Turtle() #Turtle을 1개 만들고 변수 t에 저장한다.  
t.shape("turtle") #t의 모양을 거북이로 한다.
```

#여기에 프로그래밍

#1. t를 앞으로 100만큼 이동한다.


#2. t를 왼쪽으로 120도 회전한다.


#1과 2의 과정을 2번 더 한다.


```
turtle.done() #실행 후 창이 꺼지지 않도록 프로그램의 마지막 줄에 작성한다.
```


[수업 정리]

1. 구글 드라이브 <2019_Python_학번_이름> 폴더에
<오늘날짜_python_시작하기>폴더를 생성
2. 오늘 실습한 파일을 모두 업데이트(총 4개 파일)

 Ex1_계산기프로그램.py

 Ex2_터틀모드_무작정_해보기.py

 Q1_사각형그리기.py

 Q2_삼각형그리기.py

4. 수업 피드백 작성 :

<https://forms.gle/dYnmsXxTEJZscP958>