

# 응용 프로그래밍

## python

# 수업내용

- 파이썬 소개
- 파이썬 설치 방법
- 파이썬 실행 환경

python

## 파이썬 소개

- 파이썬의 특징

# 파이썬(Python) ?

- 1991년 귀도 반 로섬(Guido van Rossum)이 개발한 대화형 프로그래밍 언어
  - ◆ 1989년 크리스마스에 할 일이 없어 혼자 집에 있다가 파이썬을 개발
  - ◆ 귀도 반 로섬이 좋아하는 코미디 프로그래밍인 "Monty Python's Flying Circus"에서 'Python'이라는 이름을 따왔다고 함



- 동적인 데이터타입 결정 지원
  - ◆ 파이썬은 동적으로 데이터 타입을 결정하므로 데이터 타입에관계없는 일반화된 코드를 작성할 수 있다.
- 파이썬은 인터프리터 방식이다.
  - ◆ 파이썬은 인터프리터 방식이다. 즉, 코드가 작성되는 즉시 실행될 수 있다.
- 파이썬은 무료다.
  - ◆ 파이썬은 오픈소스이며, 자유롭게 배포된다.
  - ◆ 파이썬 커뮤니티에서 유지 및 관리한다.
- 파이썬 코드는 심플하다.
  - ◆ 파이썬 코드는 배우기 쉽고 간단한 구조를 가지고 있다.
  - ◆ 사실, 강력한 언어 정의 방법은 코드를 읽기 쉽게 만드는 것이다.

- 파이썬은 플랫폼에 독립적인 언어이다.
  - ◆ 플랫폼은 흔히 말하는 운영체제(Operating System)를 뜻하며, 윈도우나 맥 OS, 리눅스 등이 이에 속한다.
  - ◆ '독립적'이란 '관계없는'이라는 뜻이다.
  - ◆ 플랫폼에 독립적인 언어란 어떤 운영체제든 상관없이 사용할 수 있는 언어라는 의미이다.
- 파이썬은 객체지향 언어이다.
  - ◆ 객체지향 언어는 해당 프로그램이 해결해야 할 문제의 구성요소를 요소별로 정의한 뒤, 각 요소의 기능(method)과 속성(attribute)을 정의하여 요소들을 결합하고, 프로그램을 작성하는 방식이다.

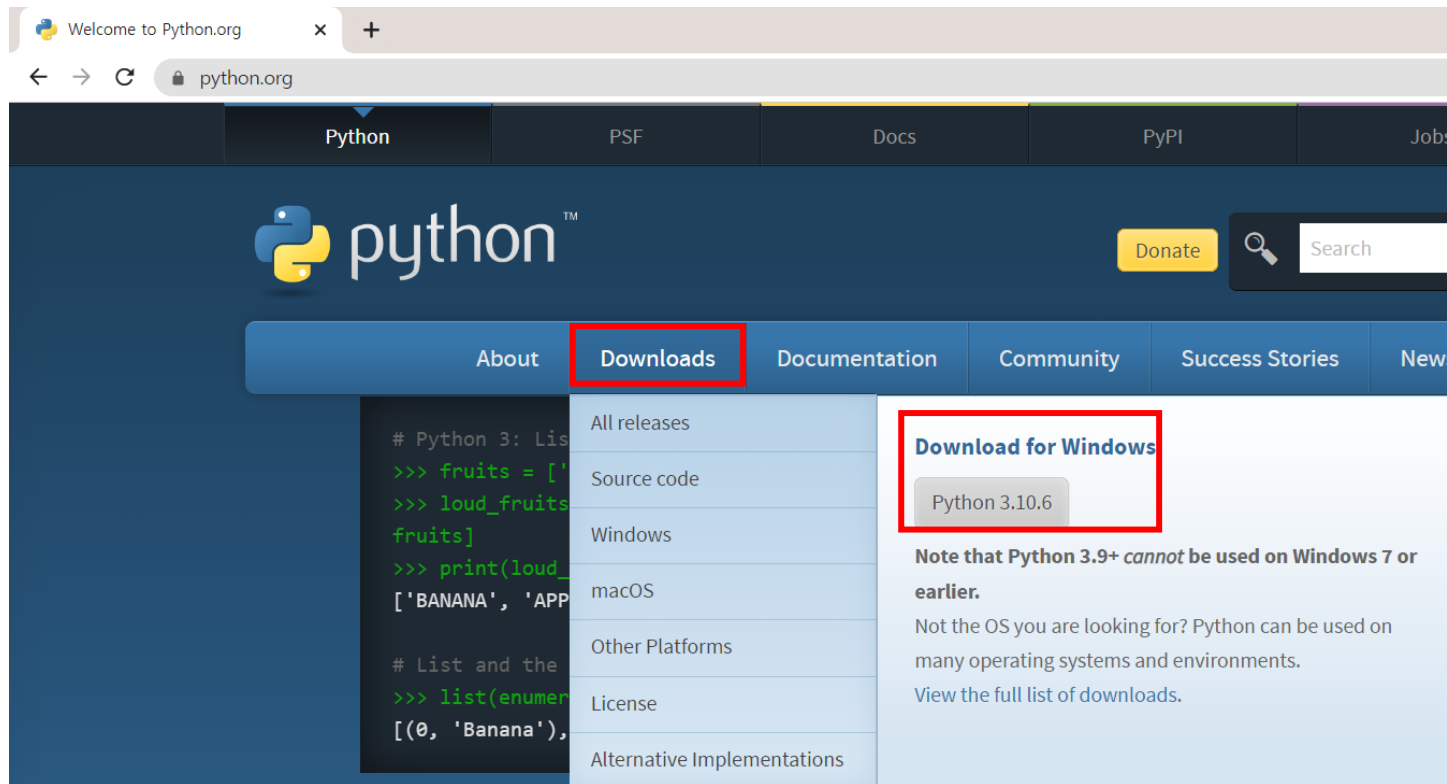
python

## 파이썬 설치 방법

- 파이썬 3.10 설치하기

# 파이썬 설치하기

- 파이썬 공식 홈페이지 <https://www.python.org/> 에서 윈도우용 파이썬 언어 패키지를 다운로드 한다.

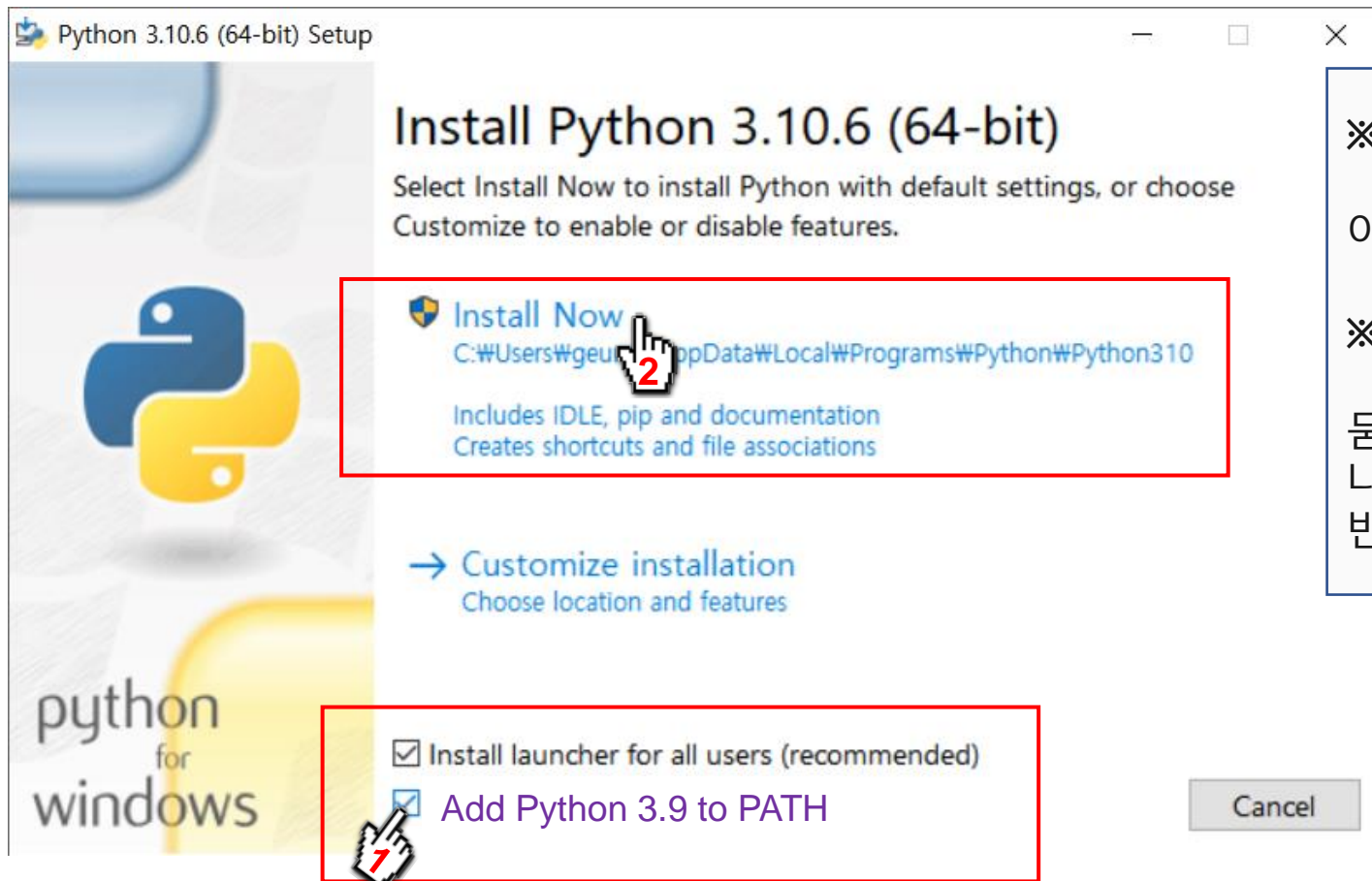


- ① [Downloads] 선택
- ② [Download for Windows] 선택



## ■ Install Python-3.10.6 (64-bit)

- ① 파이썬이 어느 곳에서든지 실행될 수 있도록 하단의 [Add Python 3.10 to PATH] 옵션을 선택
- ② [Install Now] 선택



### ※ Install launcher for all users(recommended)

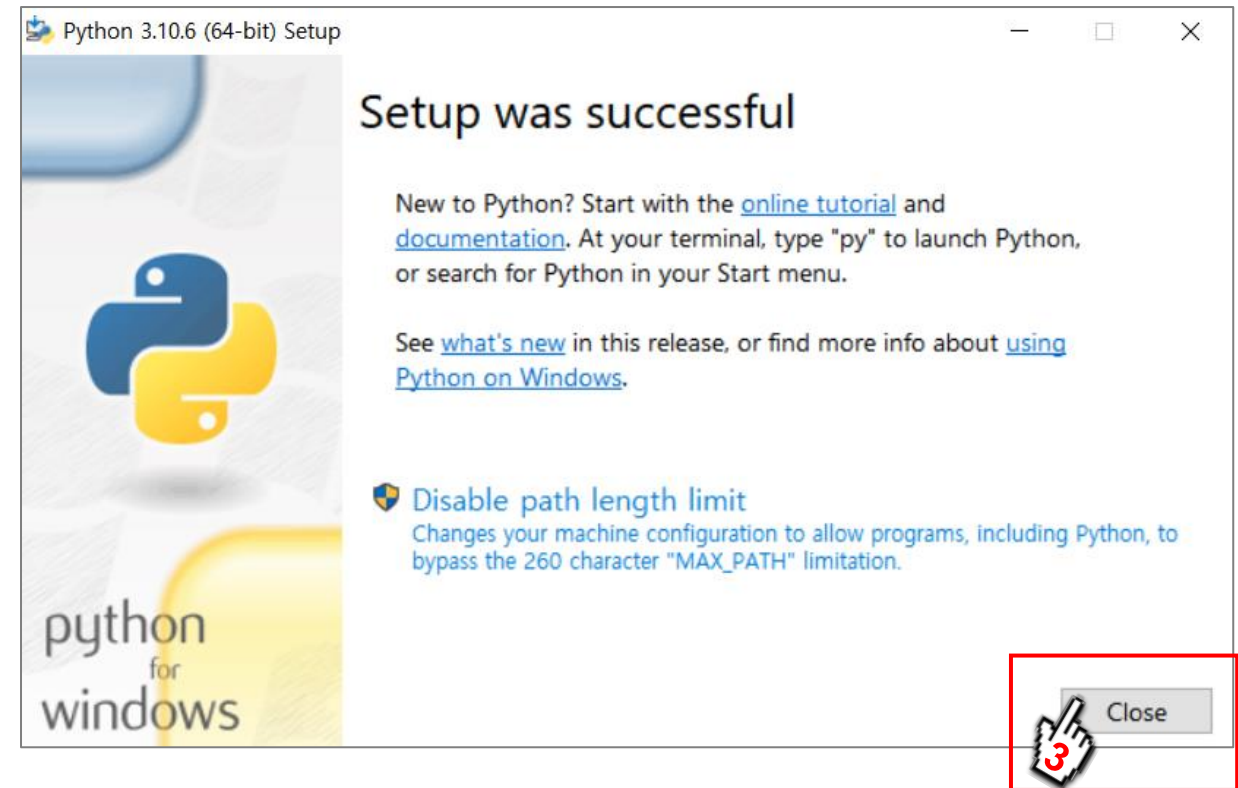
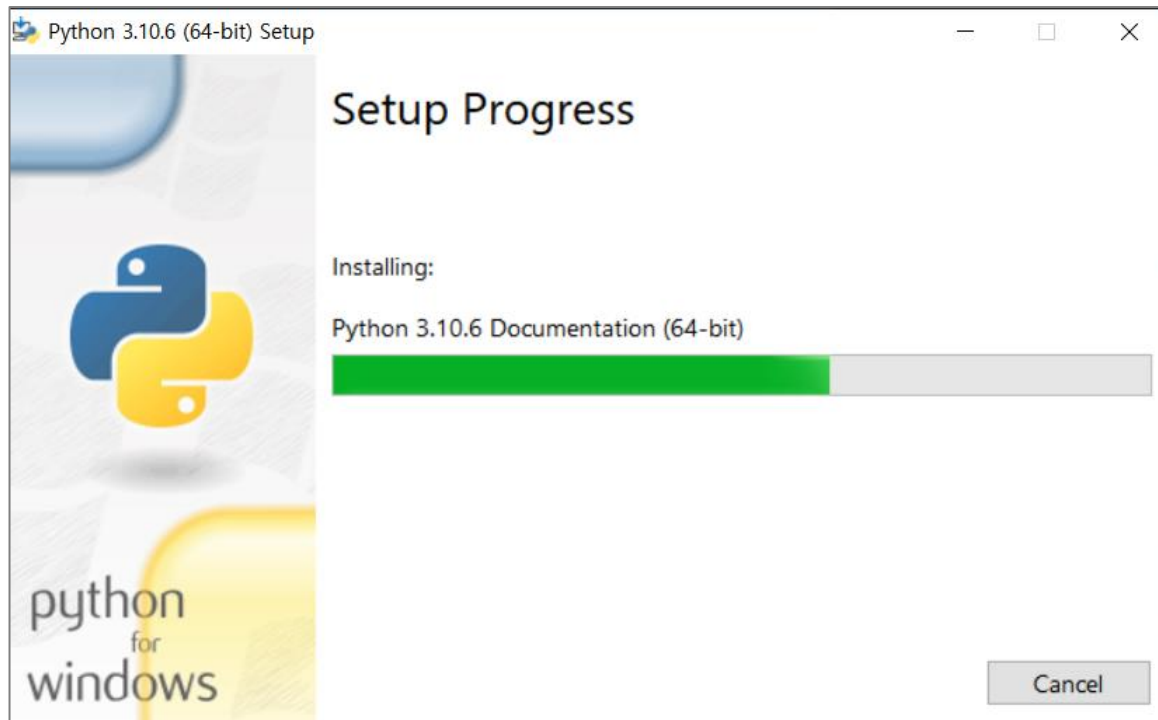
: 컴퓨터 사용자 전체가 사용하게 할 것인지, 아니면 현재의 사용자만 사용할 것인지를 묻는 옵션

### ※ Add Python 3.10 to PATH

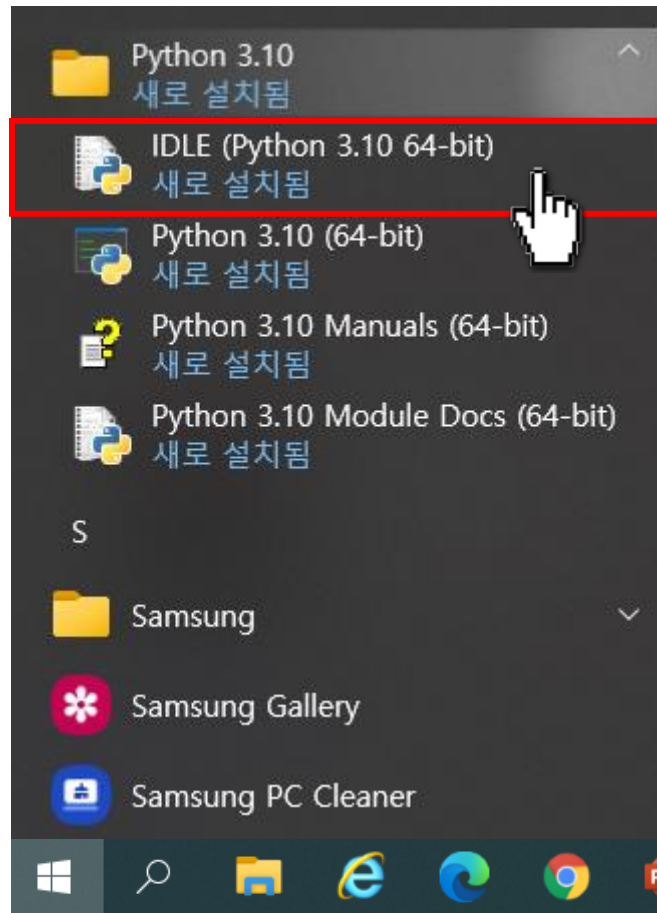
: PATH 환경 변수에 파이썬을 추가할 것인지를 묻는 옵션, 이 부분을 설치 시 체크하지 않으면 나중에 직접 환경변수 PATH를 변경하여야 한다. 반드시 체크하도록 해요~!

## ■ Install Python-3.10.6 (64-bit)

③ 설치가 완료되면 [close] 선택

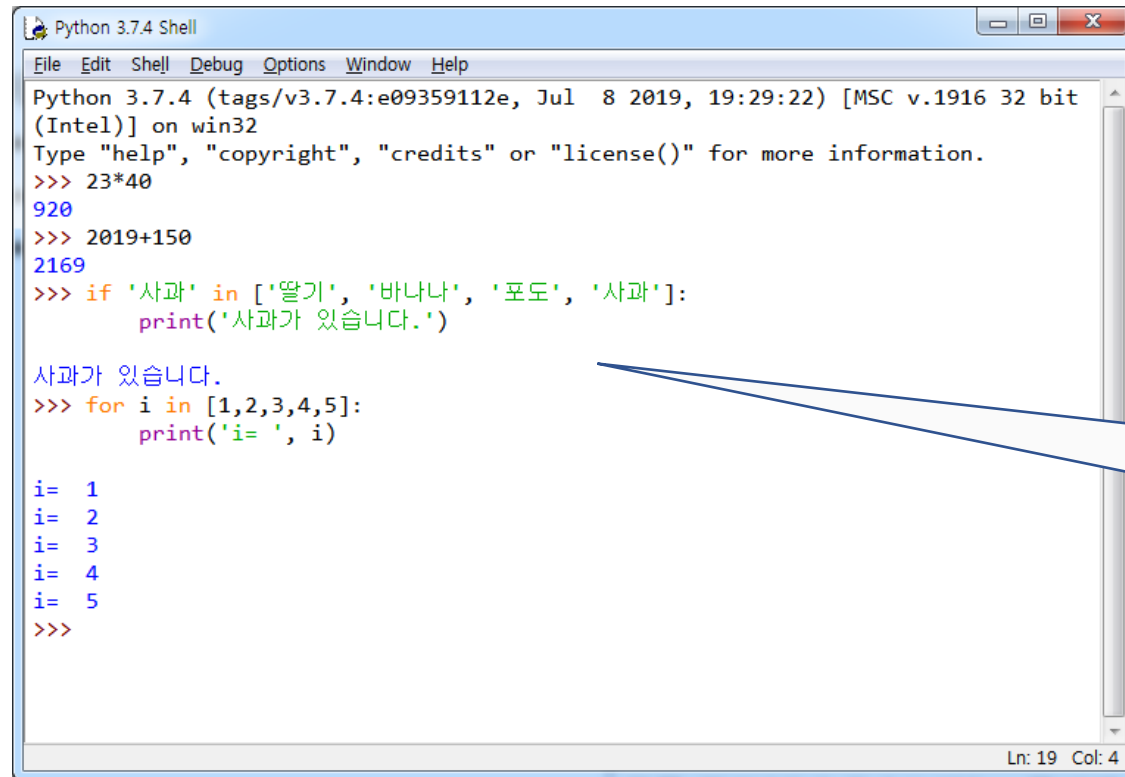


- Python-3.10 설치가 정상적으로 설치되었다면 다음과 같이 프로그램에서 메뉴에서 **[IDLE(Python 3.10 64-bit)]** 선택



## ■ IDLE(통합 개발 & 실행 환경)

- ◆ 파이썬은 윈도우 운영체제에서 사용 가능한 통합 개발 및 실행 윈도우를 지원한다.
- ◆ IDLE에서는 쉘 윈도우를 사용해서 기본적인 문법 오류를 나타내는 구문 강조 기능을 지원한다.
- ◆ IDLE는 구문 강조, 자동 완성, 자동 들여쓰기 등을 지원하는 다중 텍스트 편집 창을 지원한다.



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 23*40
920
>>> 2019+150
2169
>>> if '사과' in ['알기', '바나나', '포도', '사과']:
    print('사과가 있습니다.')

사과가 있습니다.
>>> for i in [1,2,3,4,5]:
    print('i= ', i)

i= 1
i= 2
i= 3
i= 4
i= 5
>>>
```

구문 강조(High-light) 기능 : 문법에 맞는지  
틀렸는지, 변수인지, 연산자인지, 함수인지 등을  
구분해서 다른 색깔로 구분하는 기능

python

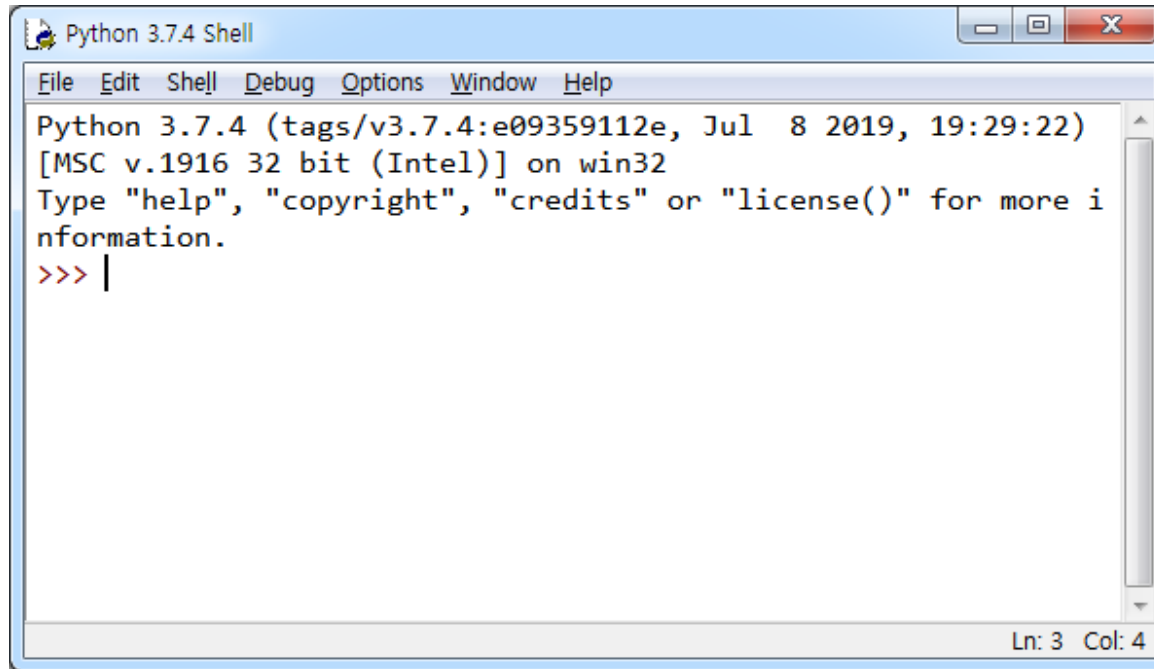
## 파이썬 실행 환경

- 파이썬 셸 윈도우
- 파이썬 코드 윈도우

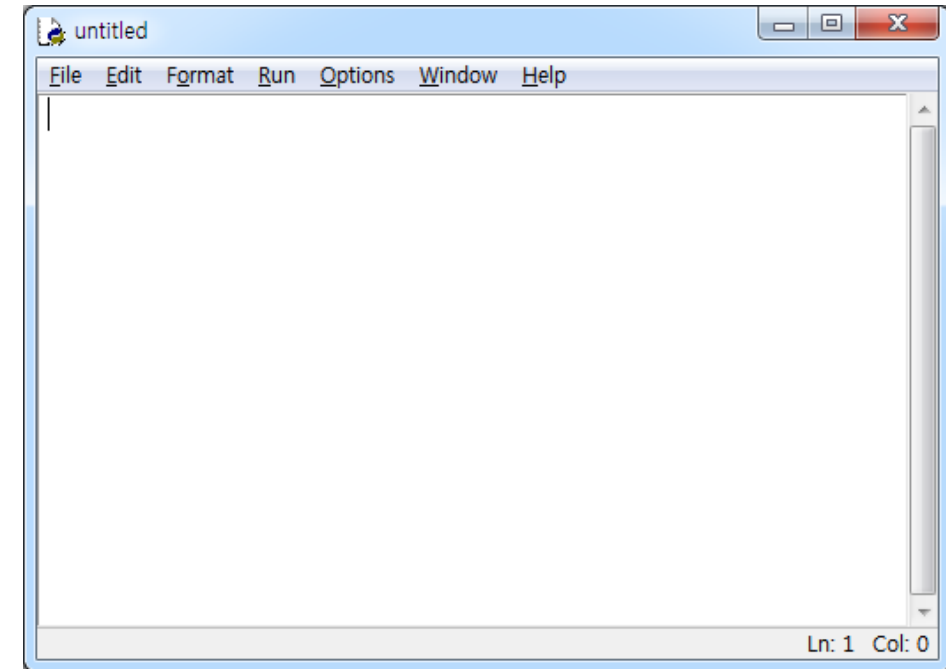
# IDLE 창(Window)

- IDLE는 크게 두 가지 창으로 구성된다 – 셸 창(shell window)과 코드 창(code window)

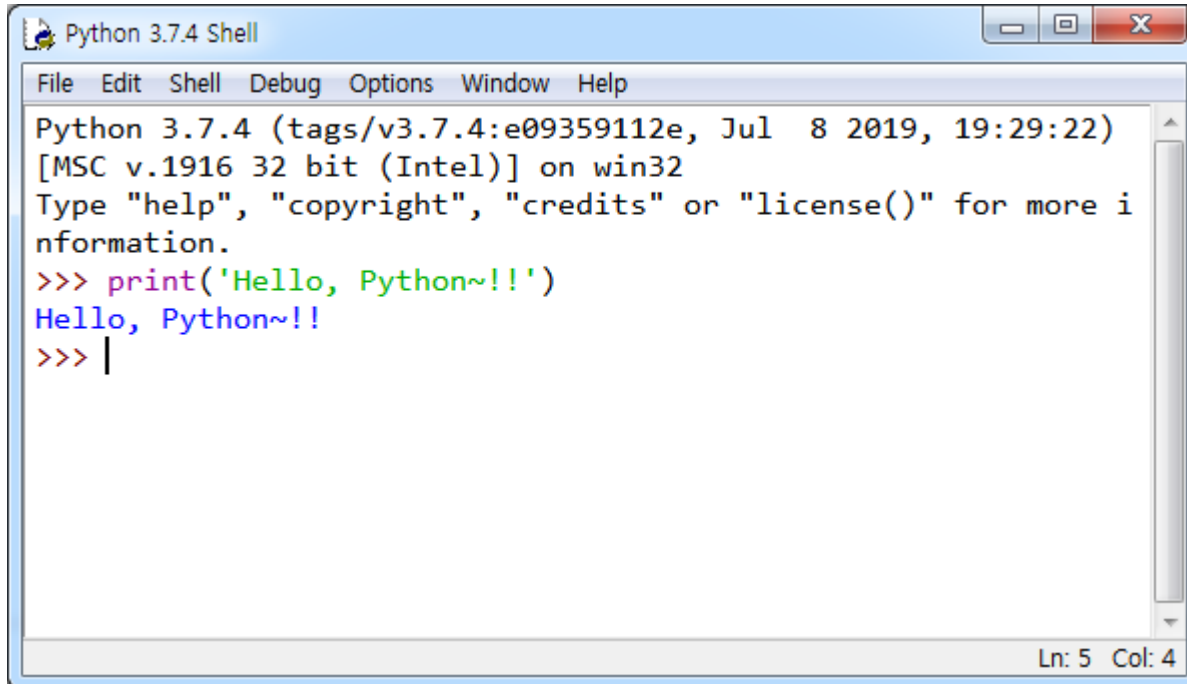
셸 창(shell window)



코드 창(code window)



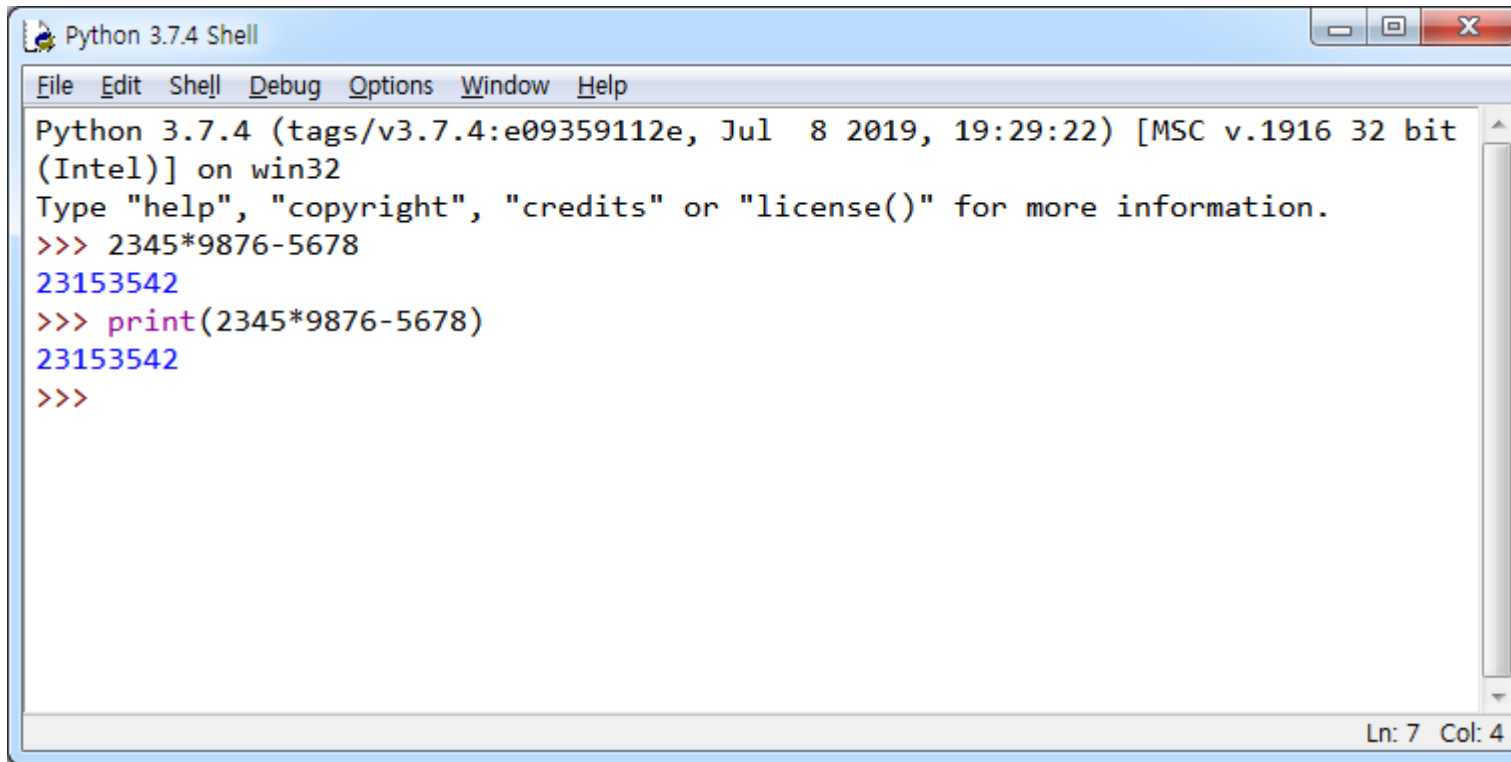
- 파이썬 IDLE를 실행하거나 코드 창에서 작성된 프로그램을 실행하면 셸 창이 자동으로 나타난다.
- 셸 창에서는 한번에 하나의 명령이 실행되고 실행 결과가 즉시 화면에 나타난다.
- **>>>**은 **프롬프트**라고 불리고 여기에 우리가 명령을 입력하고 엔터키를 누르면 명령이 실행되고 결과가 화면에 출력된다.



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22)
[MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more i
nformation.
>>> print('Hello, Python~!!')
Hello, Python~!!
>>> |
Ln: 5 Col: 4
```

# 셸 창(shell window)

- 숫자의 자료형은 정수형
- 계산 결과를 출력할 때는 **print( )** 를 사용할 수도 있다.
- 파이썬 셸 창에서 다음의 두 가지 문장은 동일하다.



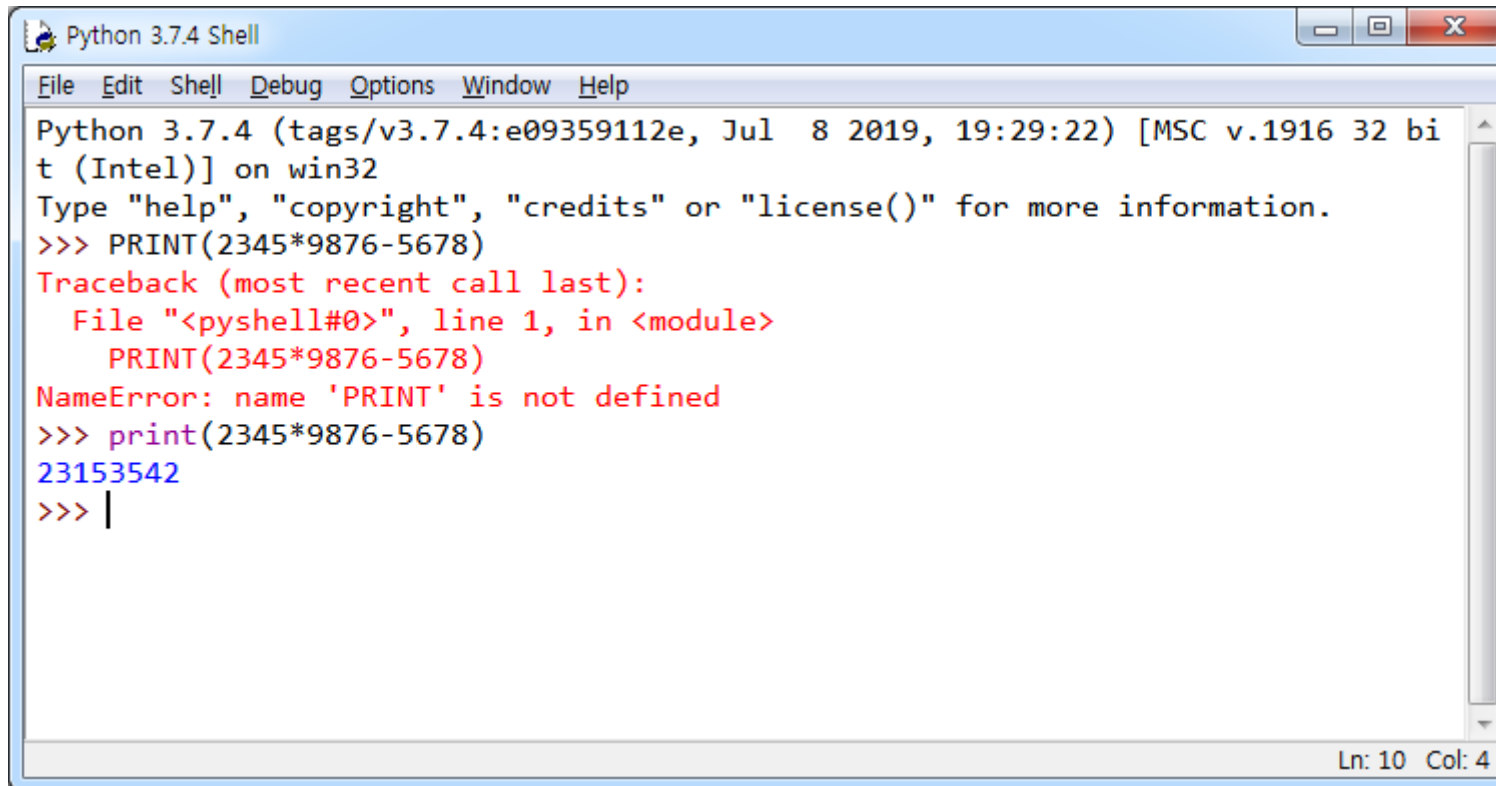
```

Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2345*9876-5678
23153542
>>> print(2345*9876-5678)
23153542
>>>
Ln: 7 Col: 4
  
```



# 셸 창(shell window)

- 파이썬은 대/소문자를 구분한다.
  - ◆ "JOHN" 과 "john"을 다른 것으로 인식한다.
  - ◆ 대문자로 명령하면 파이썬은 알아 듣지 못한다.
  - ◆ 반드시 **소문자로만 사용**하도록 한다.

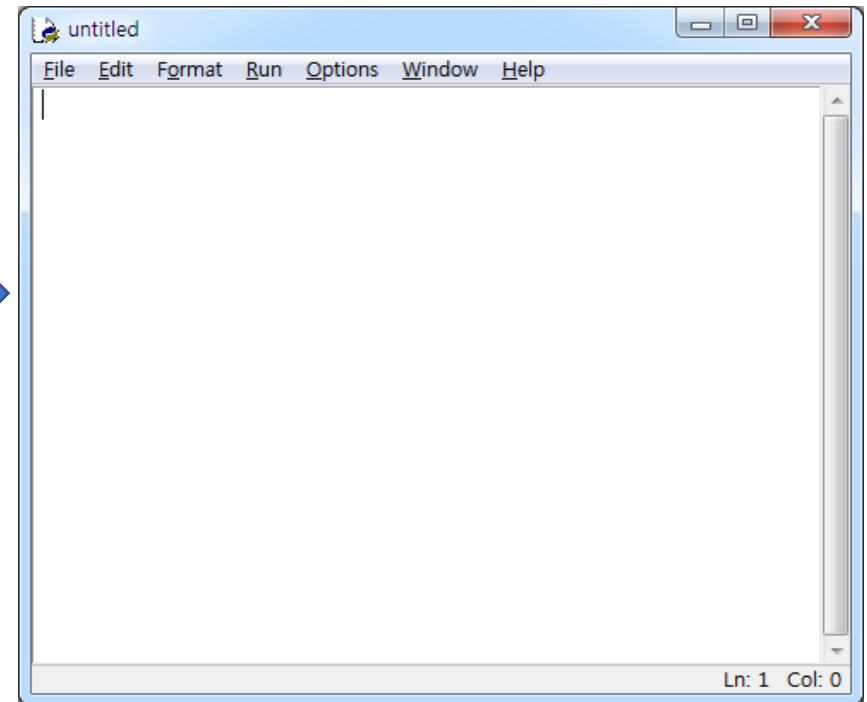
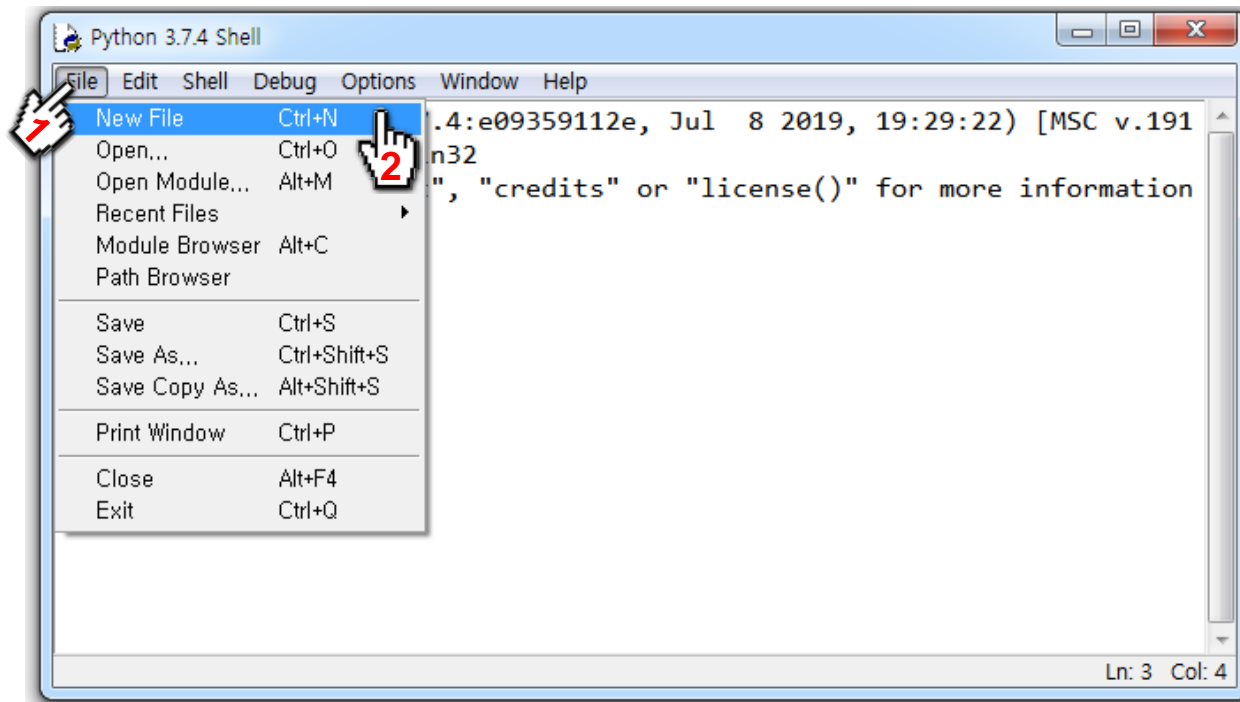


```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> PRINT(2345*9876-5678)
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    PRINT(2345*9876-5678)
NameError: name 'PRINT' is not defined
>>> print(2345*9876-5678)
23153542
>>> |
```

Ln: 10 Col: 4

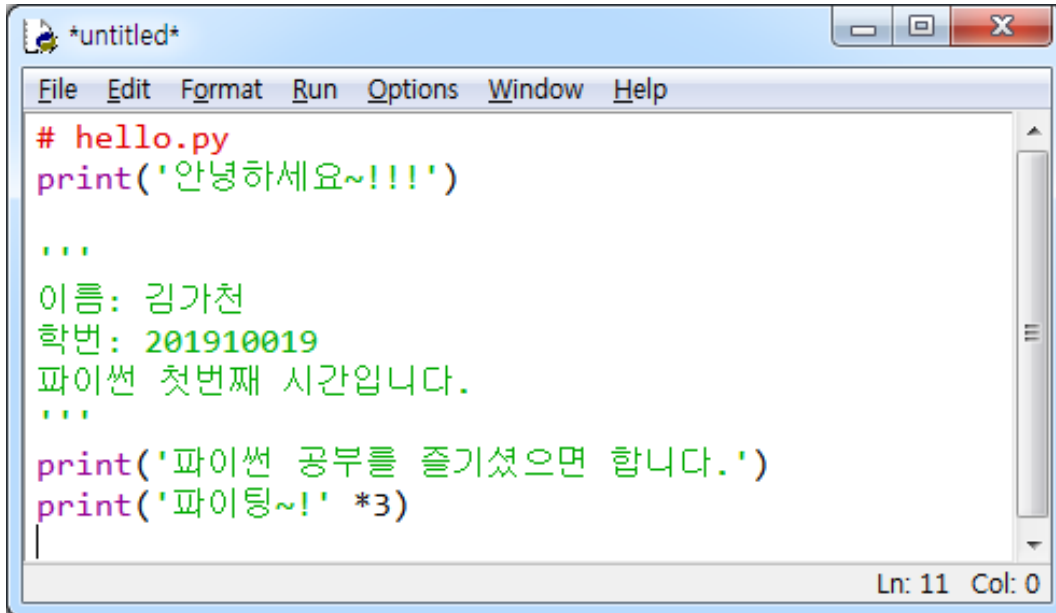
# 코드 창(code window)

- 여러 줄을 한꺼번에 작성하거나 여러 번 사용해야 하는 프로그램은 코드 창에 작성한다.
- 코드 창에서 파일을 작성하고 이 파일 안에 코드를 저장한다.
- 셸 창에서 **[File] → [New File]** 메뉴를 선택한다.



## ■ 주석(comment)

- ◆ 코드에 대한 자세한 설명을 작성하거나, 특정 코드를 임시로 사용하지 않도록 만들 때 사용
- ◆ 주석은 파이썬 인터프리터가 처리하지 않으므로 프로그램의 실행에는 영향을 주지 않는다.
- ◆ 한 줄 주석 : **#**
- ◆ 여러 줄에 대한 주석: **큰 따옴표(")** 또는 **작은 따옴표(')**를 **시작과 끝에 3개 연속으로 사용**



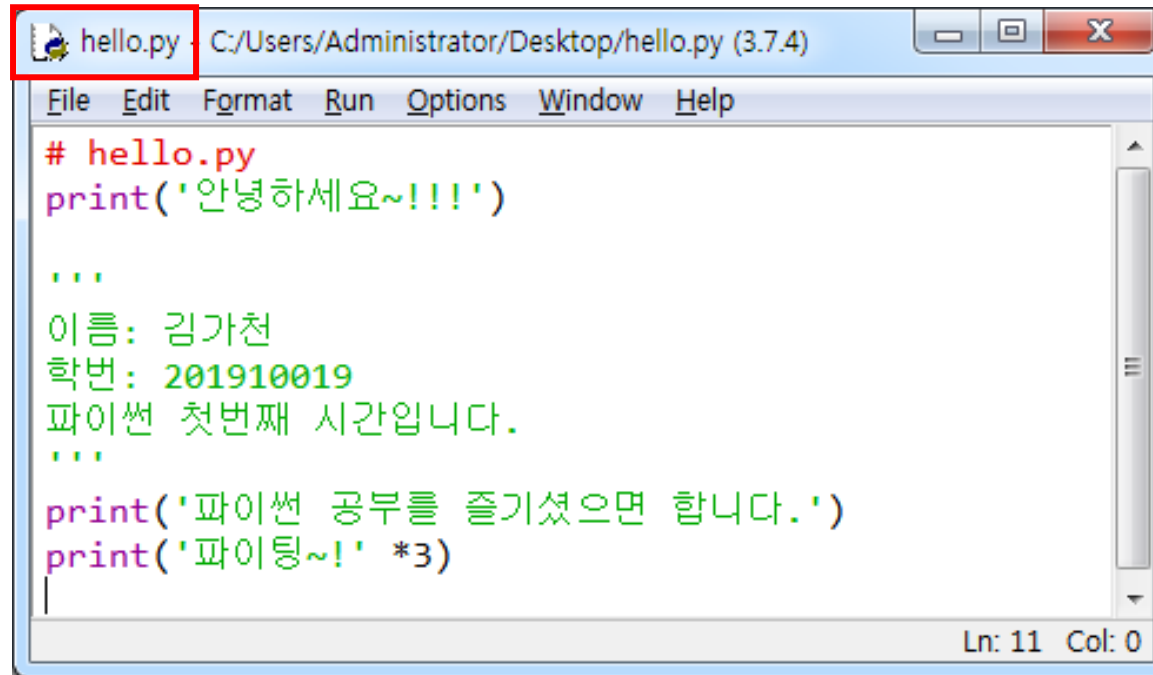
```
*untitled*
File Edit Format Run Options Window Help
# hello.py
print('안녕하세요~!!!')

...
이름: 김가천
학번: 201910019
파이썬 첫번째 시간입니다.
...
print('파이썬 공부를 즐기셨으면 합니다.')
print('파이팅~!' *3)

Ln: 11 Col: 0
```

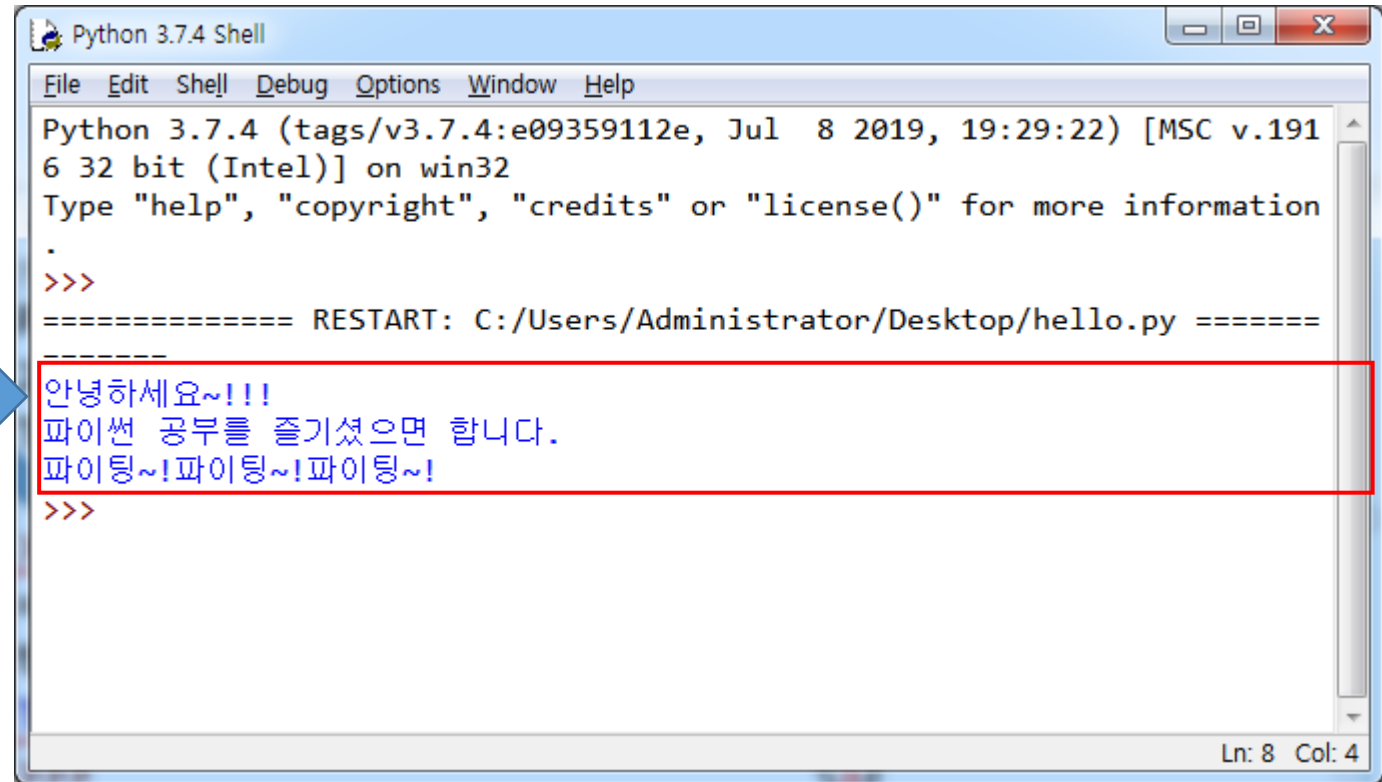
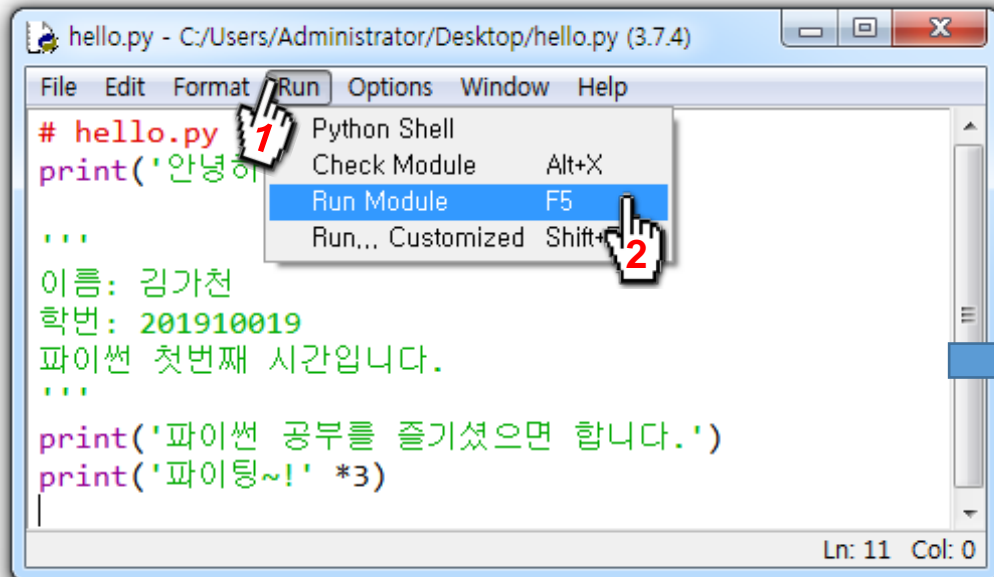
# 코드 창(code window)

- 코드 창 메뉴에서 **[File]→[Save]** 선택하여 기존 파일을 저장하거나 또는 **[File]→[Save As...]** 메뉴를 선택하여 새로운 이름으로 파일을 저장한다.
- 이때 저장하는 파일의 확장자는 **'.py'** 로 설정한다.



# 코드 창(code window)

- 작성한 코드를 실행하기 위해서는 [Run]→[Run Module]을 선택한다(단축키: F5).
- 코드 창에서 실행시킨 코드의 결과는 셸 창에 나타난다.



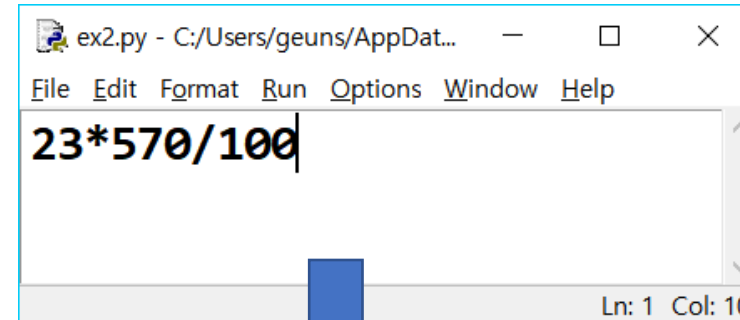
# 코드 창(code window)

- 코드 창에서는 **print( )**를 사용해야만 작성한 코드의 결과값을 셸 창에 나타낼 수 있다.
  - ◆ 셸 창에서는 코드를 작성한 후 곧 바로 결과를 표시할 수 있지만, 코드 창에서는 셸 창이 정상적으로 호출만 되고 코드 결과는 실행 되지 않는다.
  - ◆ 예를 들어, **23\*570/100** 두 가지 창에서 실행해 보세요.

## 셸 창(shell window)

```
>>> 23*570/100
131.1
```

## 코드 창(code window)

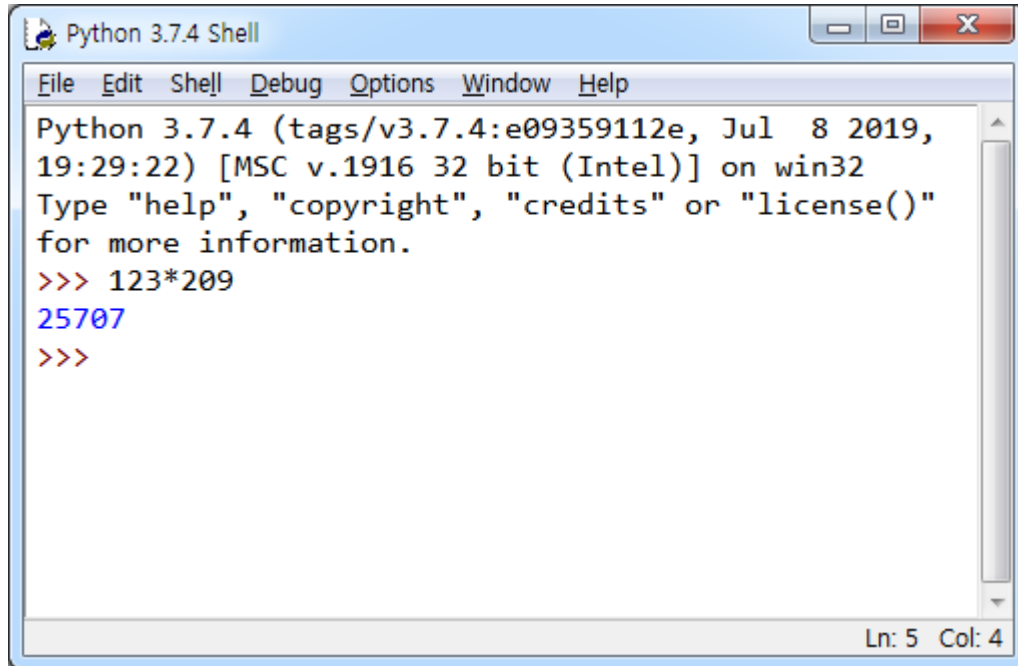


== RESTART: C:/Users/geuns/AppData/Local/Programs/Python/Python37-32/ex2.py ==

```
>>> |
```

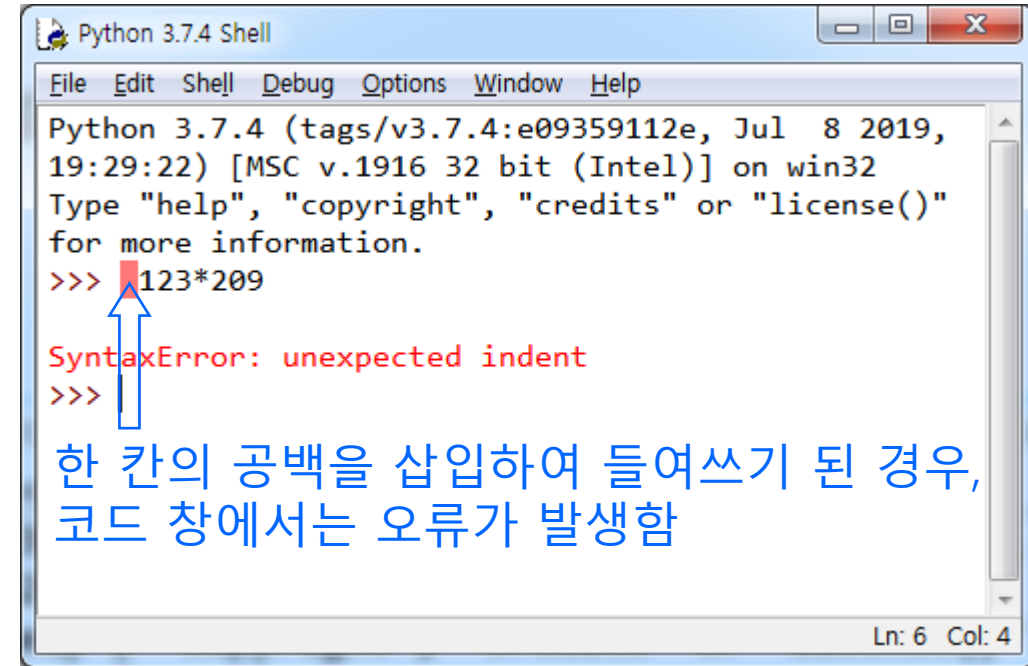
## ■ 들여쓰기(indentation)

- ◆ 코드를 읽기 쉽도록 일정한 간격을 띄워서 작성하는 방법이다.
- ◆ 파이썬은 들여쓰기 자체가 문법이다.



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 123*209
25707
>>>
```

Ln: 5 Col: 4



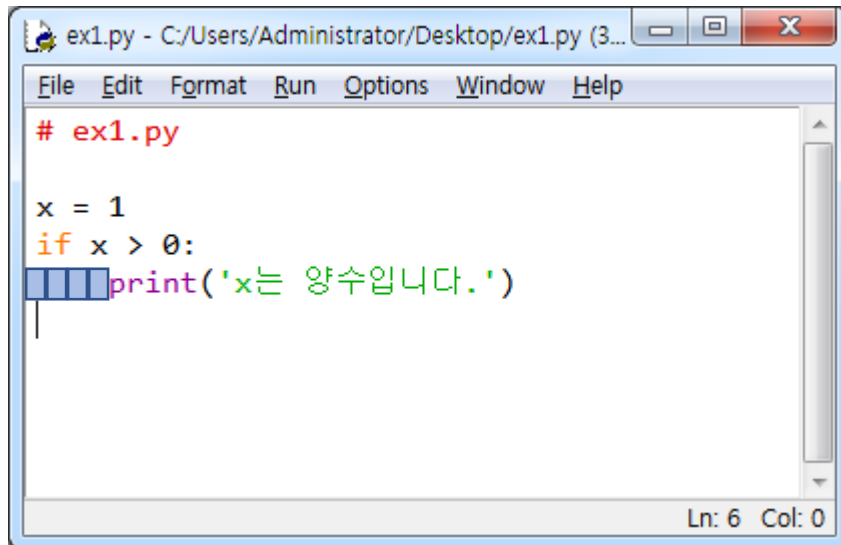
```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 123*209
SyntaxError: unexpected indent
>>>
```

한 칸의 공백을 삽입하여 들여쓰기 된 경우, 코드 창에서는 오류가 발생함

Ln: 6 Col: 4

# 셸 창에서 들여쓰기(Indentation)

- 예를 들어 if의 다음 줄은 항상 들여쓰기를 해야 한다. 만약 들여쓰기를 하지 않으면 문법 에러이므로 코드가 실행되지 않는다.
- 일반적으로 파이썬에서 들여쓰기는 **공백(스페이스) 4칸**을 사용하는 것이 좋다.

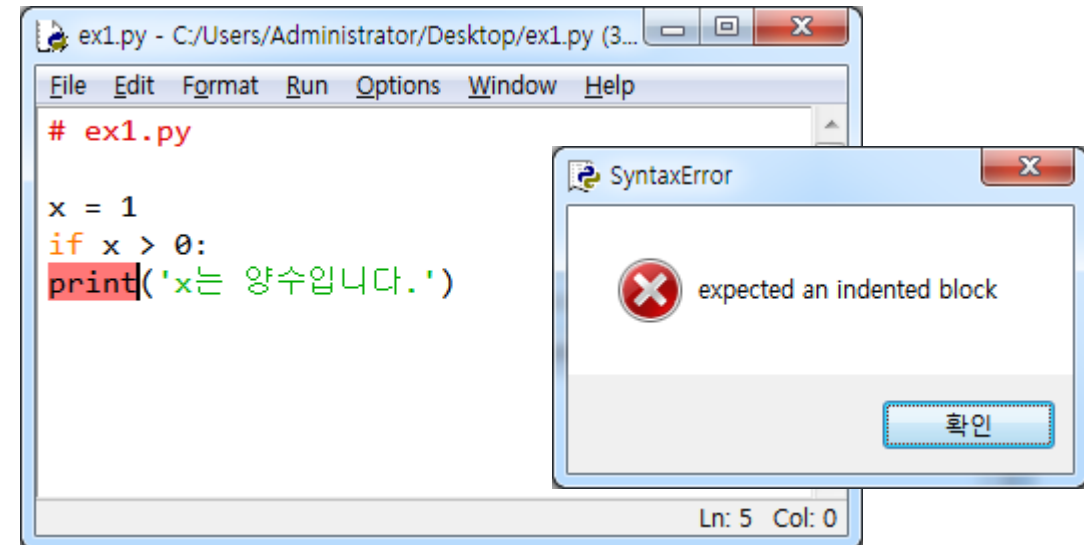


A screenshot of a Python IDE window titled 'ex1.py - C:/Users/Administrator/Desktop/ex1.py (3...'. The code inside is as follows:

```
# ex1.py

x = 1
if x > 0:
    print('x는 양수입니다.')
```

The code is correctly indented. The status bar at the bottom right shows 'Ln: 6 Col: 0'.



A screenshot of a Python IDE window titled 'ex1.py - C:/Users/Administrator/Desktop/ex1.py (3...'. The code inside is as follows:

```
# ex1.py

x = 1
if x > 0:
print('x는 양수입니다.')
```

The code is not indented, which causes a syntax error. A 'SyntaxError' dialog box is displayed over the code, with the message 'expected an indented block' and a red 'X' icon. The status bar at the bottom right shows 'Ln: 5 Col: 0'.



# Exercise

- 파이썬의 IDLE 쉘 창에서 다음과 같은 계산을 해 보세요.

1)  $3.141592 * 10.0 * 128.0$



2)  $(1/100) * 1234$



3)  $(123456789 * 123456789) - 123456789$



Q&A