

# Python

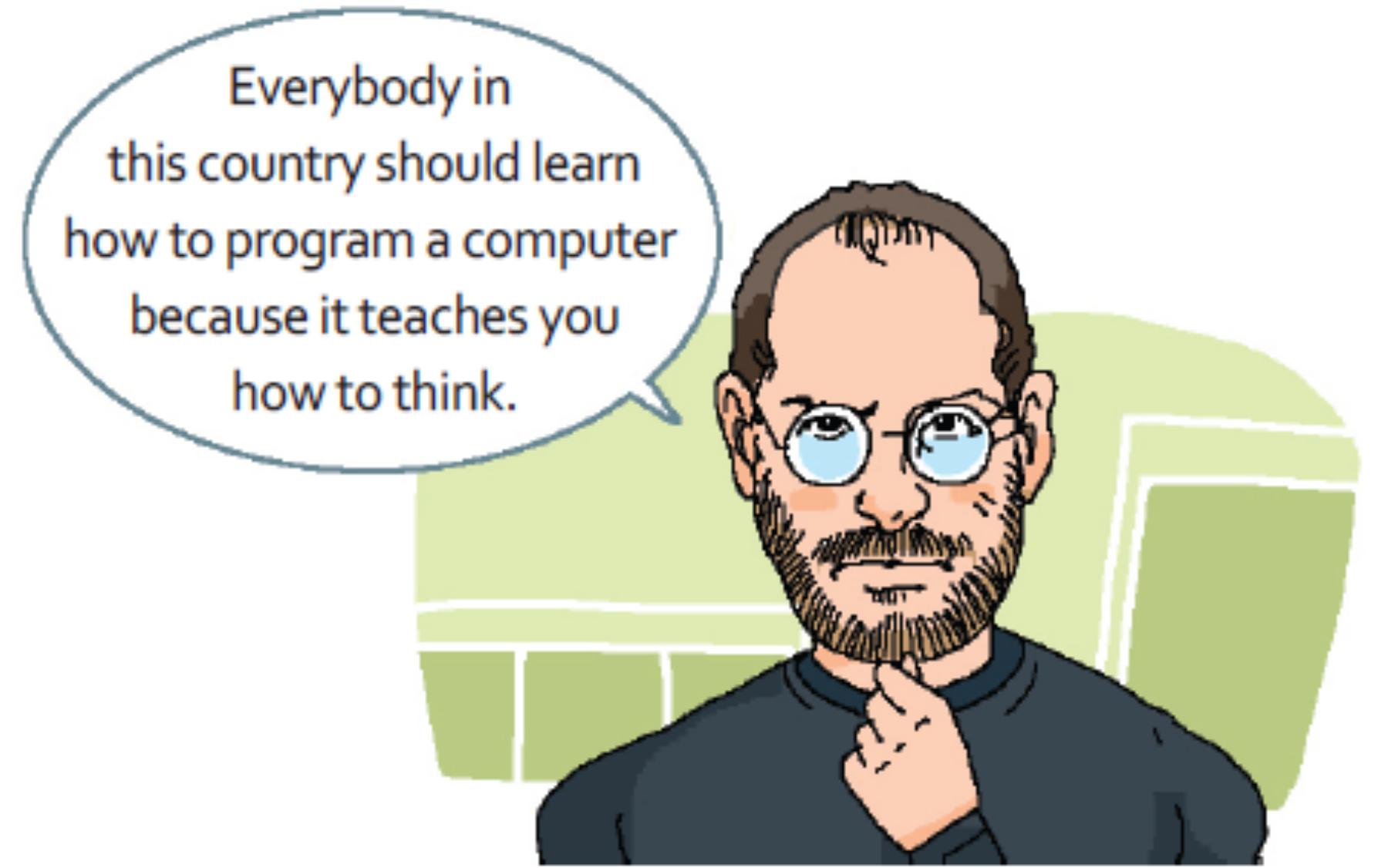
Day 1

“  
컴퓨터가 무엇을 할지는 **소프트웨어**에 의해 결정된다.  
”

“  
소프트웨어 Software = 프로그램 program  
”

“ 소프트웨어를 만드는 과정을  
프로그래밍 programming이라고 한다. ”

“ **프로그래밍 언어** programming Language는  
프로그램을 작성하기 쉽게 만든 언어. ”



이 나라 모든 사람들이 컴퓨터 프로그래밍을  
배워야 하는 이유는 사고하는 법을 가르쳐주기 때문입니다.

– 스티브 잡스(Steve Jobs)

## MOST POPULAR PROGRAMMING LANGUAGES TO LEARN IN 2019



### PYTHON

Python is easy to learn. Python has a simple syntax which makes it suitable for learning to program as the first language.



### JAVA

Java is an Object-oriented, general purpose programming language which is very helpful to create.



### C

A powerful reason is memory allocation. Unlike most programming languages, C allows programmers to write directly.

01

02

03

04

05

06



### GO

Typically, when you build an app using Java or other JVM-based languages when compiling your project.



### SWIFT

If you choose to learn iOS development, Swift is the programming language that is recommended for use.



### RUBY

Ruby runs on many platforms, like as Windows, Mac OS, and Linux. The syntax of the ruby is quite similar to C and Java.

## Top 7 Programming Languages and Their Uses



For more job search tips and advice, go to :

“  python™ is a **programming language** that lets you work quickly and integrate systems more effectively. ”

[www.python.org](http://www.python.org)

Welcome to Python.org

python.org

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!

# Input, assignment
>>> name = input('What is your name?\n')
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.
```

Quick & Easy to Learn

Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [» Learn More](#)

---

 Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

[Start with our Beginner's Guide](#)

---

 Download

Python source code and installers are available for download for all versions!

Latest: [Python 3.9.0](#)

---

 Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

[docs.python.org](#)

---

 Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.

[jobs.python.org](#)

Download Python | Python.org

python.org/downloads/

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

**Download the latest version for Mac OS X**

[Download Python 3.9.0](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases



Active Python Releases

For more information visit the [Python Developer's Guide](#).

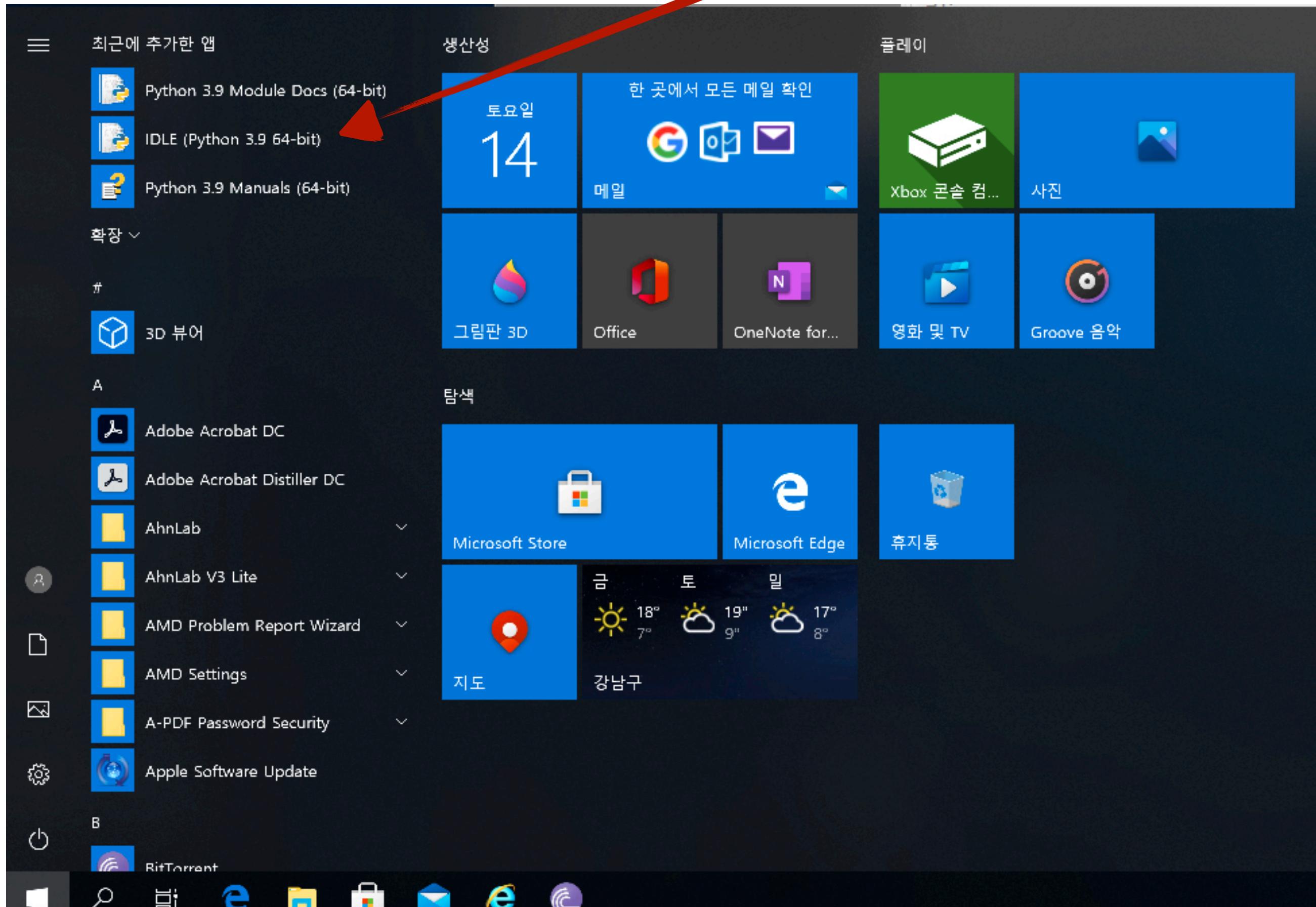
| Python version | Maintenance status | First released | End of support | Release schedule |
|----------------|--------------------|----------------|----------------|------------------|
| 3.9            | bugfix             | 2020-10-05     | 2025-10        | PEP 596          |
| 3.8            | bugfix             | 2019-10-14     | 2024-10        | PEP 569          |
| 3.7            | security           | 2018-06-27     | 2023-06-27     | PEP 537          |
| 3.6            | security           | 2016-12-23     | 2021-12-23     | PEP 494          |
| 2.7            | end-of-life        | 2010-07-03     | 2020-01-01     | PEP 373          |



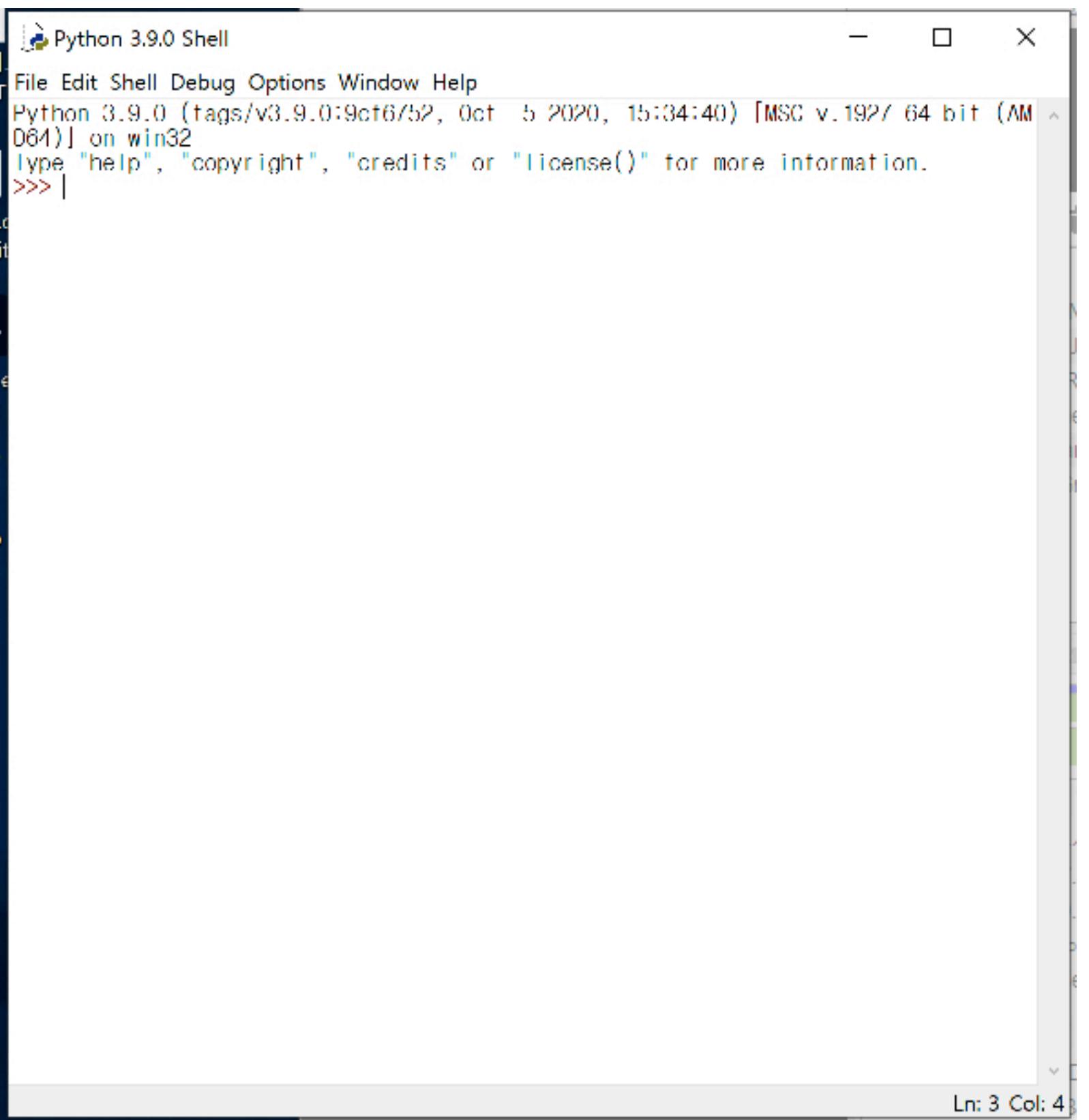
1. 설치 체크박스에 체크를 합니다

2. 01. 설치 버튼을 클릭합니다

2. 이 부분 'IDLE'을 선택한다

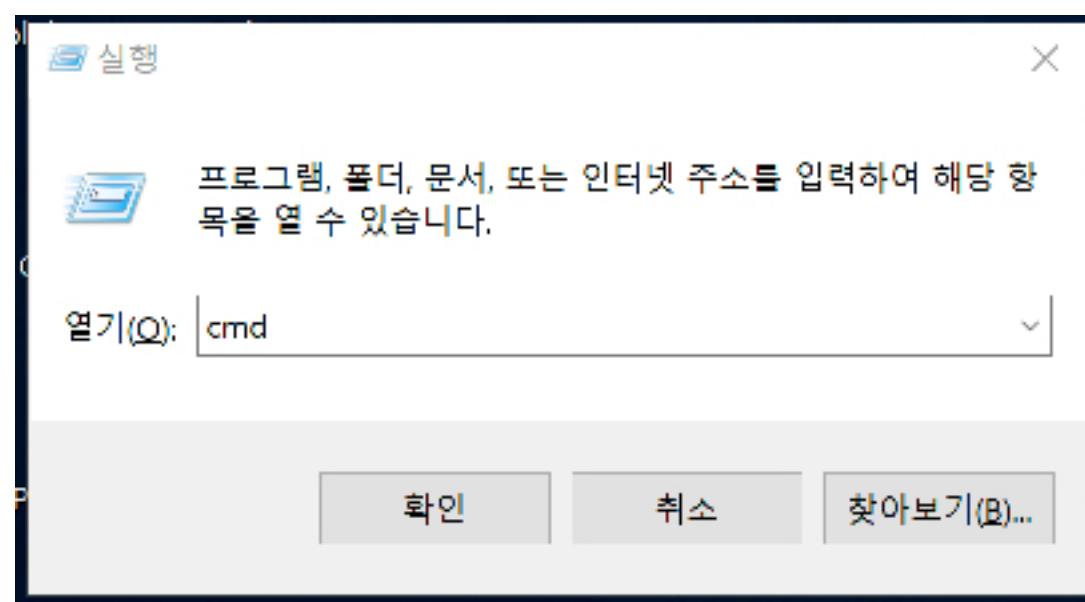


1. 윈도우즈 windows의 시작메뉴





2. %I% 창에 'cmd'를 입력하고 <enter> key



2. 커맨드 창 command window.에서 Python을 입력하고 <enter> key

A screenshot of a Windows Command Prompt window titled "cmd C:\Windows\system32\cmd.exe - python". The window displays the following text:

```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 10.0.17763.1518]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\cmp>python
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

다음을 계산해보자!

$$234 + 324 \times 23$$

이 정도라면 계산기를 사용하면 되지 뭐하러 Python을 이용하지?

스마트폰에 있는 계산기를 이용해 계산해보자.

답은 7686.

Python의 >>> prompt 다음에  $234 + 324 \times 23\frac{2}{3}$  입력해보자.

```
>>> 234 + 324 * 23
```

X기호 대신 \*기호로 곱하기 기호를 사용한다

또, 다음을 계산기로 계산해보자!

123456789123456789 ×

123456789123456789

Python으로

123456789123456789 × 123456789123456789을 7자리수로 보자.

>>> 123456789123456789 \* 123456789123456789

여러분이 뒷마당을 파서 20개의 금화가 담기 가방을 발견했다고 가정해보자.

다음 날, 지하실에 몰래 내려가서 훔아내지가 불령한 스텔링식의 복제장치에 금화를 넣어 봤다(그 기계 안에는 20개의 동전만 들어 갈 수 있다). 몇시간 후에 가자기 우~하는 소리와 함께 불꽃이 튀더니 10개의 금화가 더 나왔다.

1. 매일 한번씩 1년 동안 이러한 작업을 했다면 몇 만큼의 금화를 갖게 될까?

$$\underline{10} \times 365 = 3650$$

복제 장치에서 얻을 수 있는 금화의 수

$$20 + \underline{3650}$$

뒷 마당에서 발견한 가방 속의 금화

2. 이제, 까마귀가 여러분의 침실에 둔 바짝이는 금화를 발견하고 매주 3개씩 가져 갔다고 해보자. 일년이 지나면 몇 개의 금화가 남아 있을까?

$$\underline{3} \times 52 = 156$$

까마귀가 매주 가지런 금화의 수

$$3650 - 156 = ?$$

이 금화 문제를 파이썬으로 코딩해보자.

```
>>> 20 + 10 * 365
```

```
>>> 20 + 10 * 365
```

```
3670
```

```
>>> 3 * 52
```

```
>>> 20 + 10 * 365
```

```
3670
```

```
>>> 3 * 52
```

```
156
```

```
>>> 3670 - 156
```

앞의 수식을 한 줄로 바꿀 수 있다. 해보자!

$$20 + 10 * 365 - 3 * 52$$

자 이제 금화 문제를 조금 바꿔보자. 빨간한 금화가 35개였다면?

그리고, 복제 장치에서 10개를 복제하는 것이 아니라 9개를 복제한다면?

$$\underline{20} + \underline{10 * 365} - \underline{3 * 52}$$

20: 뒷 마당에서 발견한 개방 속의 금화 수  
10 \* 365: 복제장치가 만들어 낸 금화 수  
3 \* 52: 가마구가 매주 가지던 금화 수

$$35 + 9 * 365 - 3 * 52$$

**변수** variable : 값을 기억(memory)하기 위한 저장소

found\_coins = 20

magic\_coins = 10

stolen\_coins = 3

found\_coins + magic\_coins \* 365 - stolen\_coins \* 52

found\_coins = 35

magic\_coins = 9

stolen\_coins = 3

found\_coins + magic\_coins \* 365 - stolen\_coins \* 52

# 자료형 (data type)

## 1. 숫자형

| 항목   | 사용 예                    |
|------|-------------------------|
| 정수   | 123, -345, 0            |
| 실수   | 123.45, -1234.5, 3.4e10 |
| 8진수  | 0o34, 0o25              |
| 16진수 | 0x2A, 0xFF              |

# 자료형 (data type)

2. 문자열(string) : 문자, 단어 등으로 구성된 문자들의 집합

“Life is too short, You need Python”

“a”

“123”

# 자료형 (data type)

## 문자열 표현

“Life is too short, You need Python”

‘Life is too short, You need Python’

“Life is too short,\n You need Python”

‘Life is too short,\n You need Python’

““ Life is too short,

You need Python””

“““ Life is too short,

You need Python”””

# 자료형 (data type)

3. 불(Boolean) 자료형 : 참(True)과 거짓(False)을 나타내는 자료형

True

False

4. 리스트(list), 튜플(tuple), 딕셔너리(dictionary), 집합(set)

# 기본 입출력 함수

`input()`

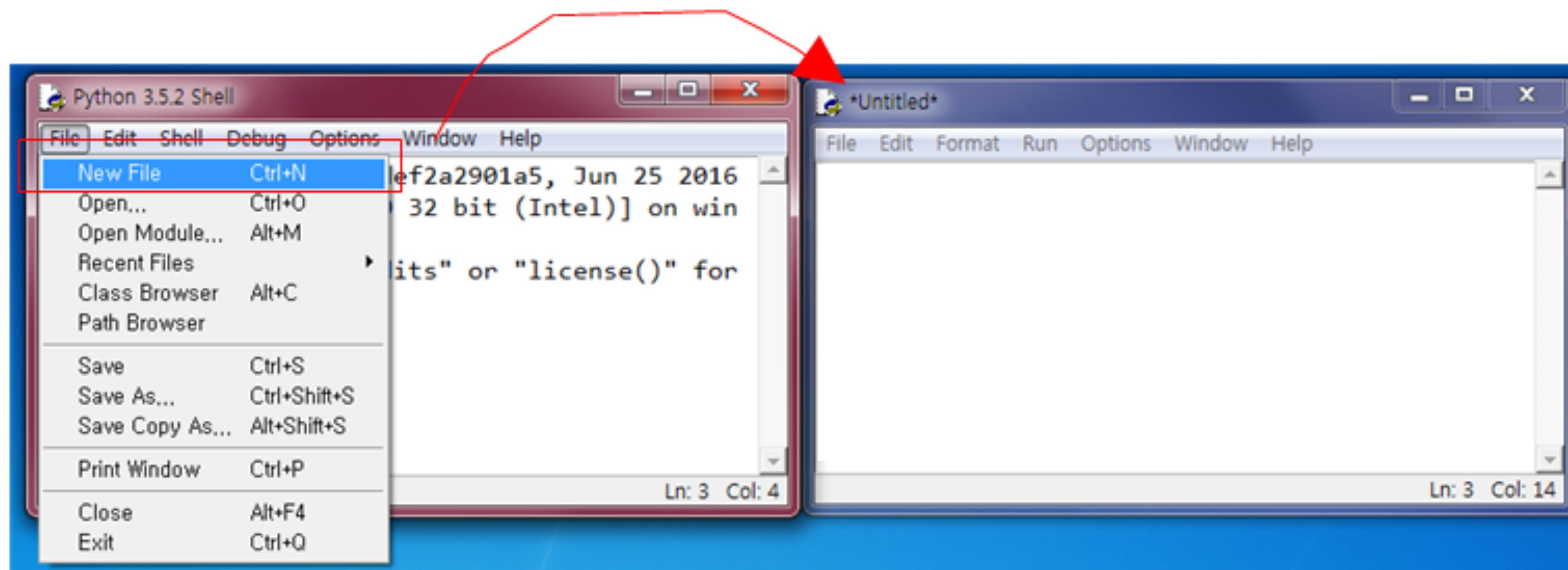
`input("질문 내용")`

`print(출력을 원하는 값)`

# 기본 입출력 함수

```
found_coins = input("발견한 금화의 갯수")
magic_coins = input("복제기계가 만든 금화의 갯수")
stolen_coins = input("까마귀가 가져간 금화의 갯수")
print(found_coins + magic_coins * 365 - stolen_coins * 52)
```

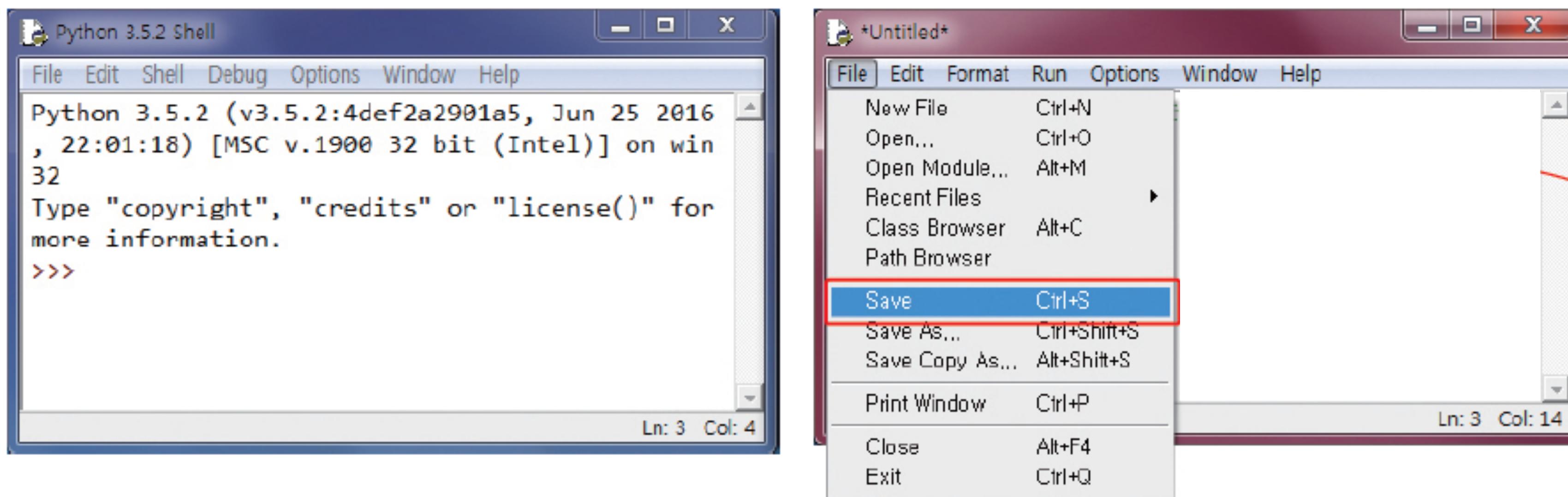
# File > New File



새로 생긴 창에 아래 코드를 입력해 보자

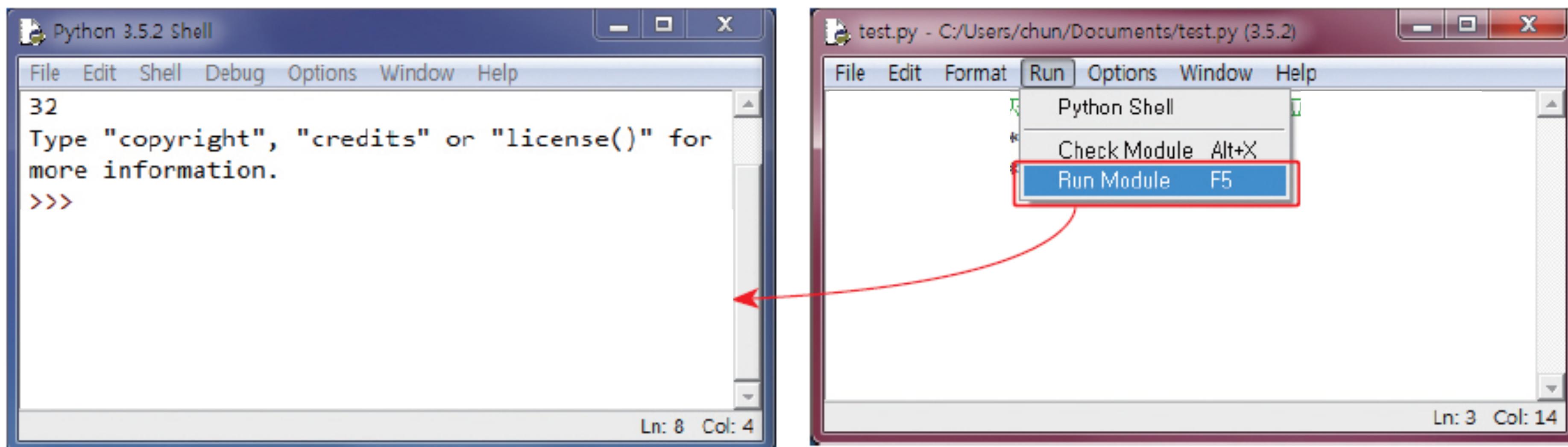
```
found_coins = input("발견한 금화의 갯수")
magic_coins = input("복제기계가 만든 금화의 갯수")
stolen_coins = input("까마귀가 가져간 금화의 갯수")
print(found_coins + magic_coins * 365 - stolen_coins * 52)
```

# File > Save



İstediğiniz **ex1** adını girin.

# Run > Run Module



# Jump to Python

<https://wikidocs.net/book/1>

Q & A