

2D 게임 프로그래밍

## 제0강 초간단 파이썬

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

한국산업기술대학교 이대현



github - Google Searchpython - Google Searchstart [이대현의 게임프로]

www.dae Hyunlee.com/doku.php

incognitek

startblogpersonalprojects

Trace: • start

### Dae-Hyun Lee's Knowledge Warehouse

한국산업기술대학교 게임공학과 이대현 교수의 강의 홈페이지입니다. 여기의 자료들은 누구나 자유롭게 이용하실 수 있습니다. 이 자료들이 여러분들의 게임 프로그래밍 실력 향상에 조금이라도 도움이 될 수 있기를 바랍니다. 그리고 이 자료들을 인용할 때, 출처를 밝혀주는 당신은 정말 멋쟁이!!! 저와 연결하세요



#### Table of Contents

- Dae-Hyun Lee's Knowledge Warehouse
- Notice
- Lectures 2017 Fall
- Previous Lectures
- Links
- Copyright

### Notice

More

### Lectures 2017 Fall

- 2D게임프로그래밍(2학년2학기)
  - 강의 GIT-HUB • 주간 반 프로젝트 • 주간2반 프로젝트 • 야간반 프로젝트
- 중합설계기획 (3학년2학기)
  - 프로젝트보고
- 중합설계(III) (4학년2학기)

# 2DGP Github 저장소

The screenshot shows the GitHub repository page for 'game-lecture/2017-2DGP'. The repository has 3 commits, 1 branch, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted with a red box. A dropdown menu is open, showing the 'Clone with HTTPS' option, which is also highlighted with a red box. The URL 'https://github.com/game-lecture/2017-2DGP' is displayed in the dropdown. The repository contains files such as 'First Lecture PDF', 'Slides', '2017 Fall 2DGP Syllabus.pdf', 'README.md', and 'Initial commit'.

game-lecture / 2017-2DGP

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. Add topics

3 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/game-lecture/2017-2DGP

Open in Desktop Download ZIP

dustindaehyunlee First Lecture PDF

Slides First Lecture PDF

2017 Fall 2DGP Syllabus.pdf New syllabus

README.md Initial commit

2017-2DGP

# Git 의 설치

The screenshot shows the Git website homepage. The browser window has two tabs: 'Git' and '새 탭'. The address bar shows 'https://git-scm.com'. The page features the Git logo and the tagline '--local-branching-on-the-cheap'. A search bar is located in the top right. The main content area describes Git as a free and open source distributed version control system. It highlights features like easy learning, tiny footprint, lightning fast performance, and features like cheap local branching, convenient staging areas, and multiple workflows. A 'Try Git' button is present. Below this, there are four sections: 'About' (advantages compared to other systems), 'Documentation' (command reference, Pro Git book, videos), 'Downloads' (GUI clients, binary releases), and 'Community' (bug reporting, mailing list, chat). On the right, a monitor displays the 'Latest source Release 2.9.3' with a 'Downloads for Windows' button.

Git --local-branching-on-the-cheap

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.

Learn Git in your browser for free with **Try Git**.

**About**  
The advantages of Git compared to other source control systems.

**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases for all major platforms.


**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release  
**2.9.3**  
Release Notes (2016-08-12)  
Downloads for Windows

# TortoiseGit 의 설치

TortoiseGit – Windows 새 탭

← → ↻ <https://tortoisegit.org>

 **TortoiseGit**  
Windows Shell Interface to Git

About Download Support

이 페이지를 번역하시겠습니까? [음션](#)

번역 안함 ▼

## The Power of Git – in a Windows Shell

TortoiseGit provides overlay icons showing the file status, a powerful context menu for Git and much more!

Learn more [about TortoiseGit](#).

Download

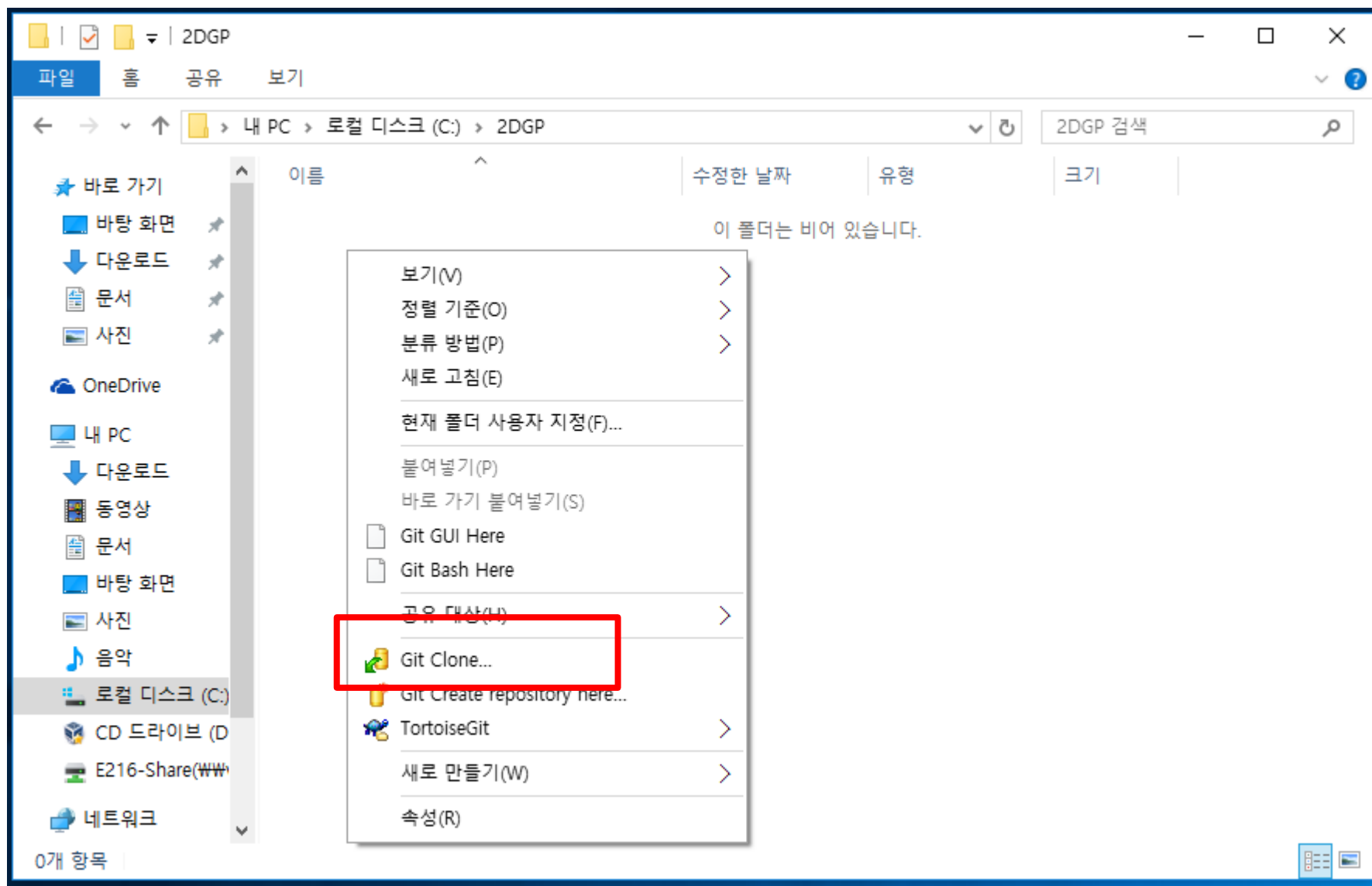
### News

- 2016-07-02 | Released TortoiseGit 2.2.0.0
- 2016-03-26 | Released TortoiseGit 2.1.0.0
- 2016-02-29 | Released TortoiseGit 2.0.0.0

### Releases

- Stable: **2.2.0** | Released on 2016-07-02
- Preview: **2.2.3.0** | Released on 2016-08-19

# Git Clone



URL: <https://github.com/game-lecture/2017-2DGP.git>

Git clone - TortoiseGit

Clone Existing Repository

URL:  ☒

Directory:

☐ Depth  ☐ Recursive ☐ Clone into Bare Repo ☐ No Checkout

☐ Branch  ☐ Origin Name

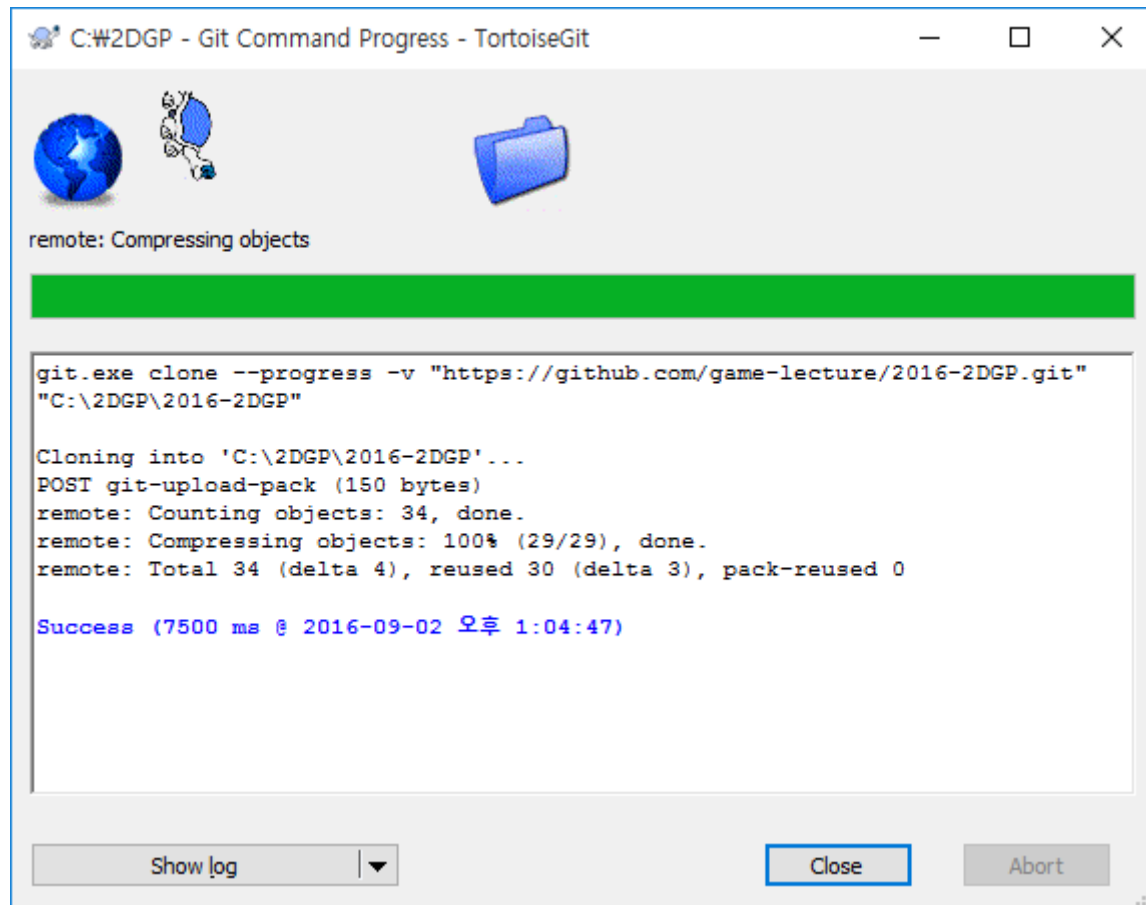
☐ Load Putty Key

From SVN Repository

☐ From SVN Repository

☐ Trunk:  ☐ Tags:  ☐ Branch:

☐ From:  ☐ Username:





## 인간다운 언어이다

프로그래밍이란 컴퓨터에 인간이 생각하는 것을 입력시키는 행위라고 할 수 있다. 앞으로 살펴볼 파이썬 문법들에서도 볼 수 있겠지만 파이썬은 사람이 생각하는 방식을 그대로 표현할 수 있도록 해주는 언어이다. 따라서 프로그래머는 굳이 컴퓨터식 사고 방식으로 프로그래밍을 하려고 애쓸 필요가 없다. 이제 곧 어떤 프로그램을 구상하자마자 생각한대로 쉽게 술술 써내려가는 여러분의 모습에 놀라게 될 것이다. 아래 예문을 보면 이 말이 더 쉽게 이해될 것이다.

```
if 4 in [1,2,3,4]: print ("4가 있습니다")
```

위의 예제는 다음처럼 읽을 수 있다:

"만약 4가 1,2,3,4중에 있으면 "4가 있습니다"를 출력한다."

프로그램을 모르더라도 직관적으로 무엇을 뜻하는지 알 수 있지 않겠는가? 마치 영어문장을 읽는 듯한 착각에 빠져든다.

## 간결하다

파이썬은 간결하다. 이 간결함은 파이썬을 만든 귀도(Guido)의 의도적인 산물이다. 만약 어떤 언어(Perl?)가 하나의 일을 하기 위한 방법이 100가지라면 파이썬은 가장 좋은 방법 1가지를 선호한다. 이 파이썬의 간결함이란 철학은 소스코드에도 그대로 적용되어 파이썬 프로그래밍을 하는 사람들은 잘 정리되어 있는 소스코드를 볼 수 있게 되었다. 다른 사람들의 소스 코드가 한눈에 들어오기 때문에 이 간결함은 공동 작업에 매우 큰 역할을 하게 되었다. 다음은 파이썬 프로그램의 예제이다:

```
# simple.py
languages = ['python', 'perl', 'c', 'java']

for lang in languages:
    if lang in ['python', 'perl']:
        print("%6s need interpreter" % lang)
    elif lang in ['c', 'java']:
        print("%6s need compiler" % lang)
    else:
        print("should not reach here")
```

## 프로그래밍이 재밌다

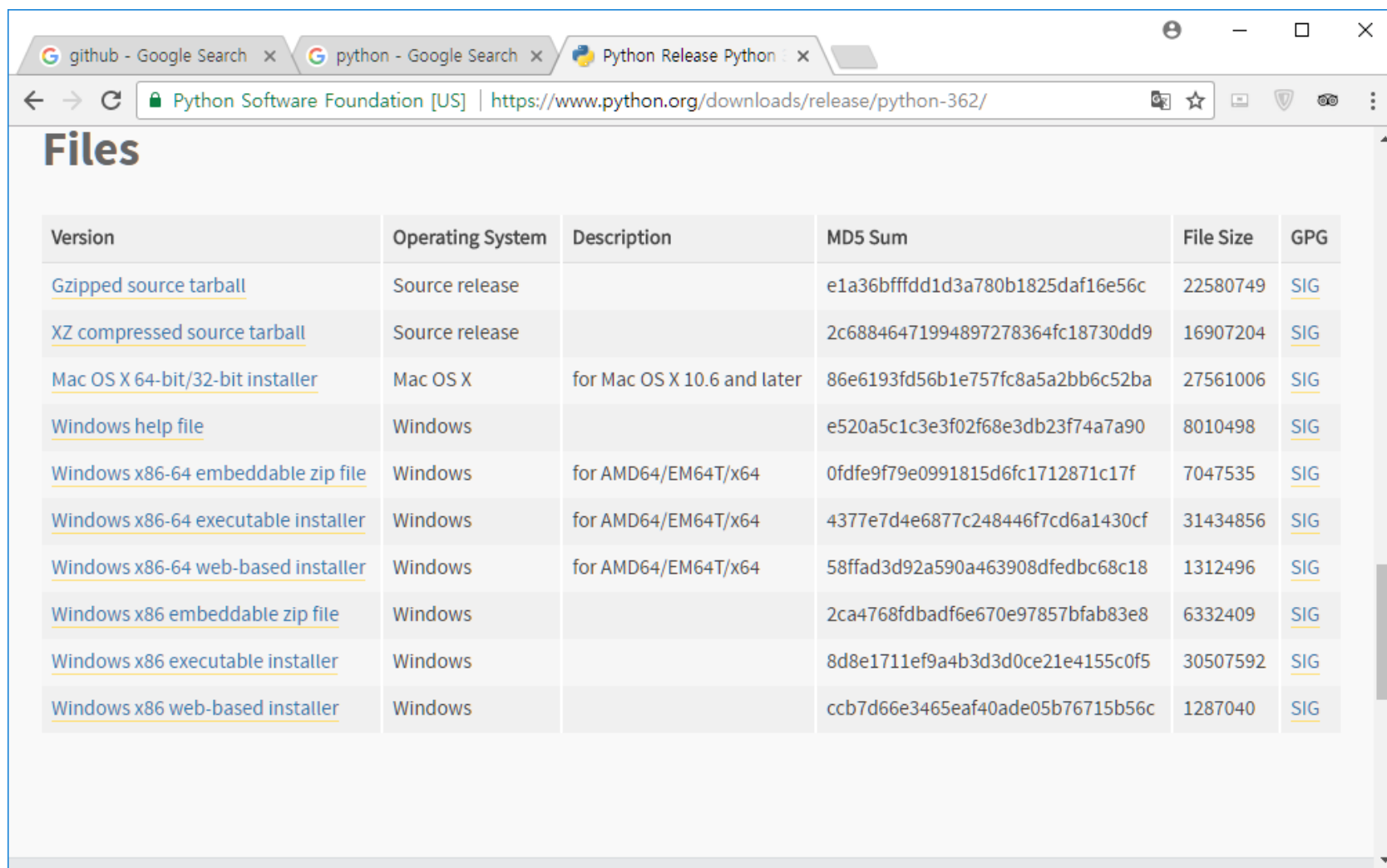
이 부분이 가장 강조하고 싶은 부분이다. 필자에게 파이썬만큼 프로그래밍을 하는 순간을 즐기게 해준 언어는 없었던 것 같다. 파이썬은 다른 것에 신경 쓸 필요 없이 내가 하고자 하는 부분에만 집중할 수 있게 해주기 때문이다. 억지로 만든 프로그램과 즐기면서 만든 프로그램, 과연 어떤 프로그램이 좋을까? 리누즈 토발즈는 재미로 리눅스를 만들었다고 하지 않는가? 파이썬을 배우고 나면 다른 언어로 프로그래밍을 하는 것에 지루함을 느끼게 될 지도 모른다. 조심하자! ^^

## 개발속도가 빠르다

마지막으로 다음의 재미있는 말로 파이썬의 특징을 마무리하려 한다.

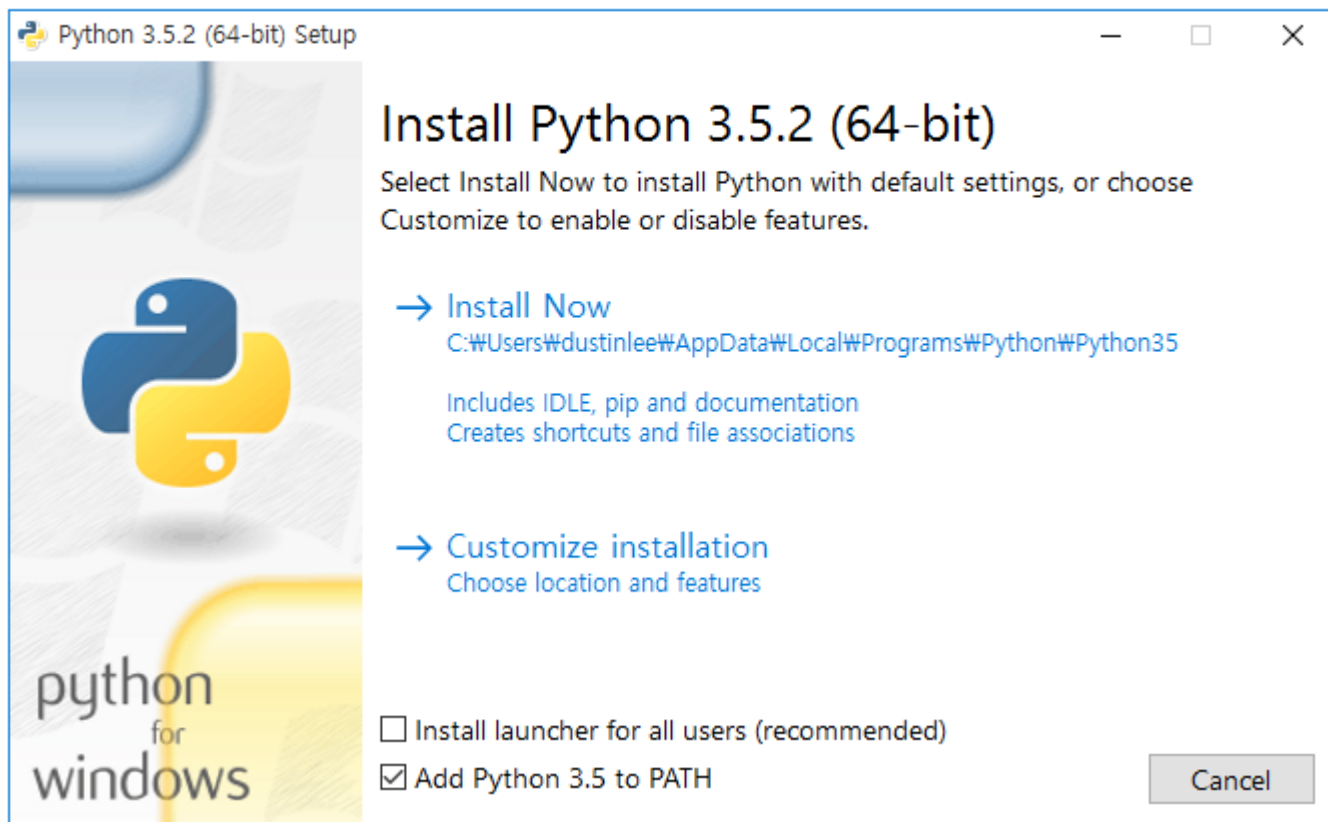
*Life is too short, You need python.*

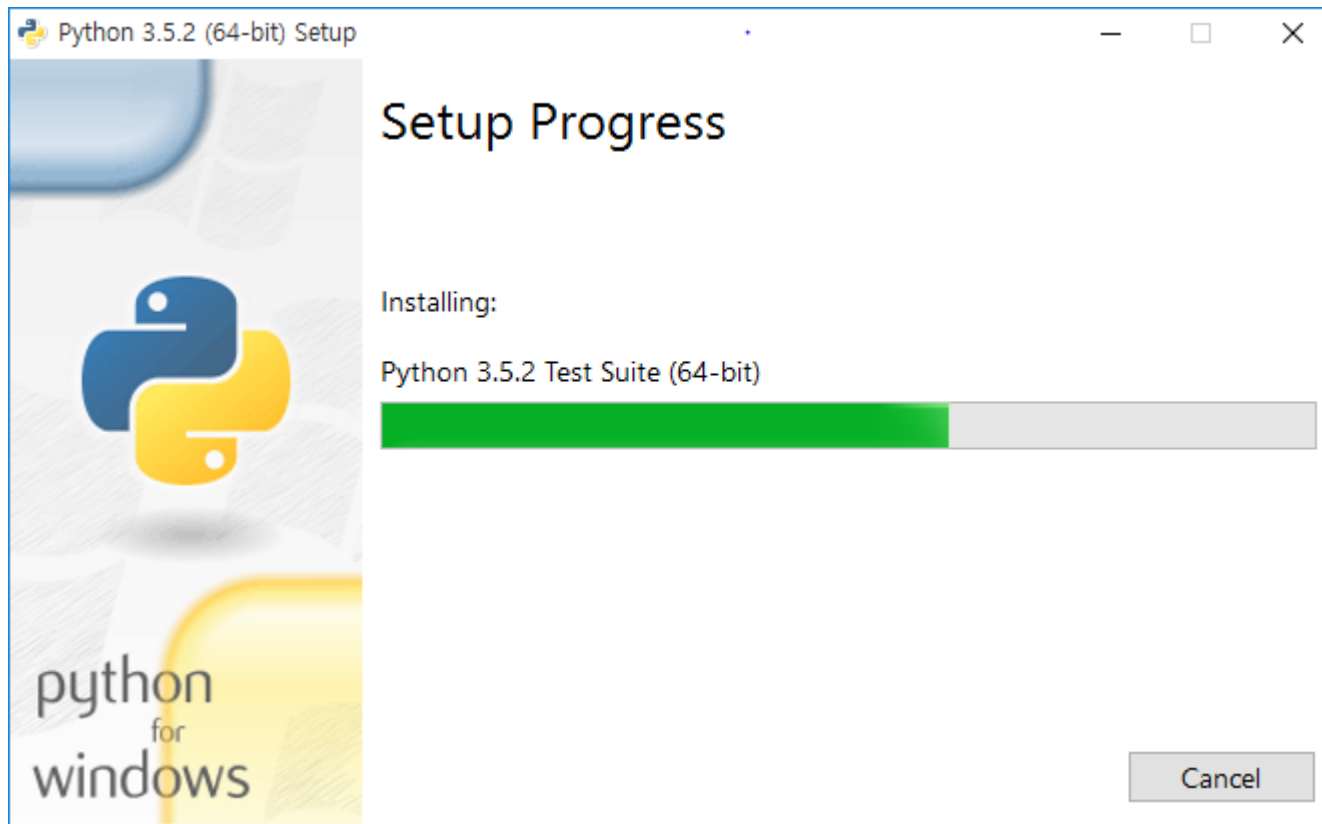
# Python 최신 버전 다운로드(2017년 9월 현재, 3.6.2 가 최신 버전)



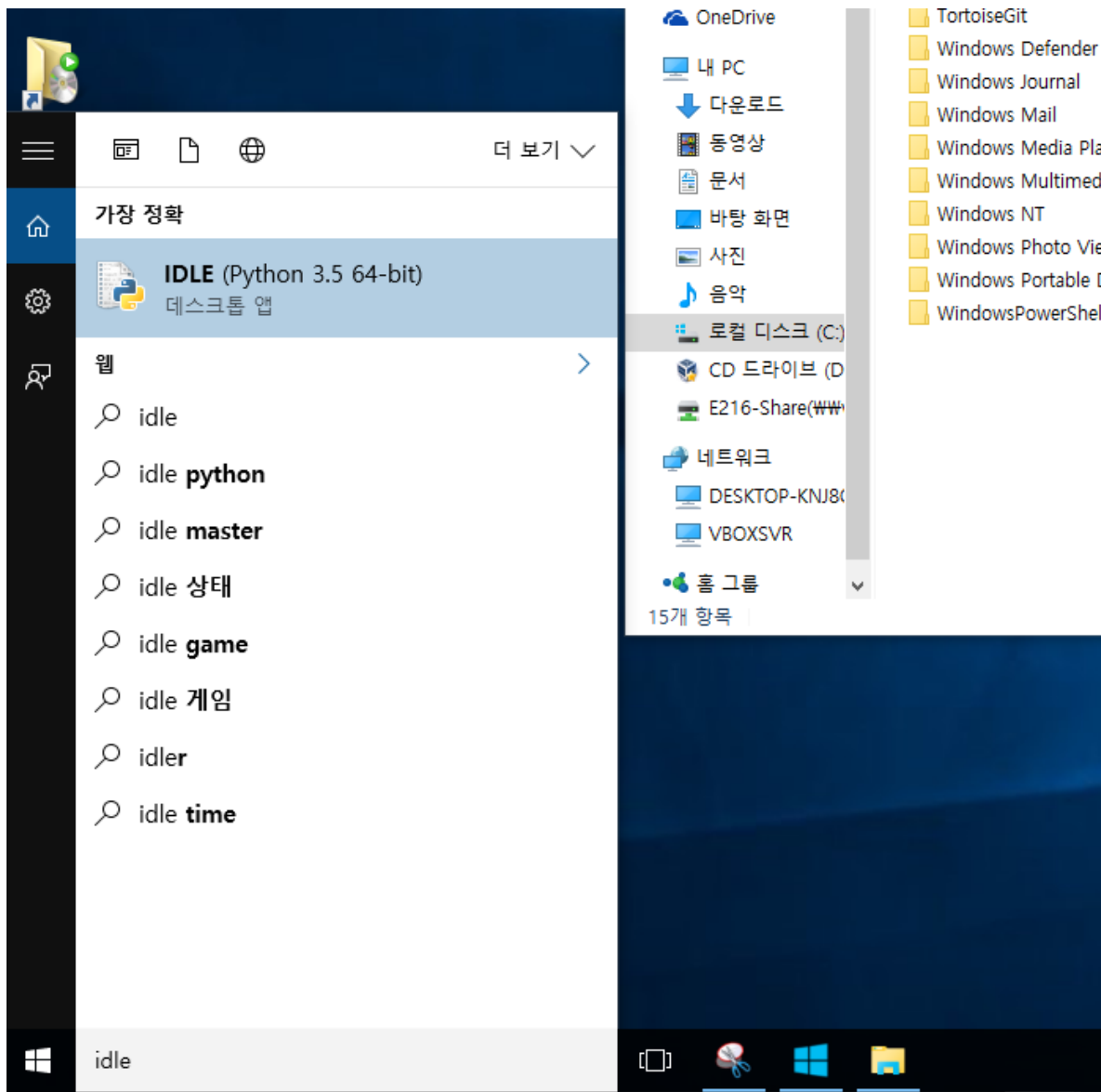
The screenshot shows a web browser window with the URL <https://www.python.org/downloads/release/python-362/>. The page title is "Files". Below the title is a table with 6 columns: Version, Operating System, Description, MD5 Sum, File Size, and GPG. The table lists various download options for Python 3.6.2, including source tarballs, installers, and executables for different operating systems (Mac OS X, Windows).

Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		e1a36bffd1d3a780b1825daf16e56c	22580749	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		2c68846471994897278364fc18730dd9	16907204	<a href="#">SIG</a>
<a href="#">Mac OS X 64-bit/32-bit installer</a>	Mac OS X	for Mac OS X 10.6 and later	86e6193fd56b1e757fc8a5a2bb6c52ba	27561006	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		e520a5c1c3e3f02f68e3db23f74a7a90	8010498	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	0fdfe9f79e0991815d6fc1712871c17f	7047535	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	4377e7d4e6877c248446f7cd6a1430cf	31434856	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	58ffad3d92a590a463908dfedbc68c18	1312496	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		2ca4768fdbadf6e670e97857bfab83e8	6332409	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		8d8e1711ef9a4b3d3d0ce21e4155c0f5	30507592	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		ccb7d66e3465eaf40ade05b76715b56c	1287040	<a href="#">SIG</a>





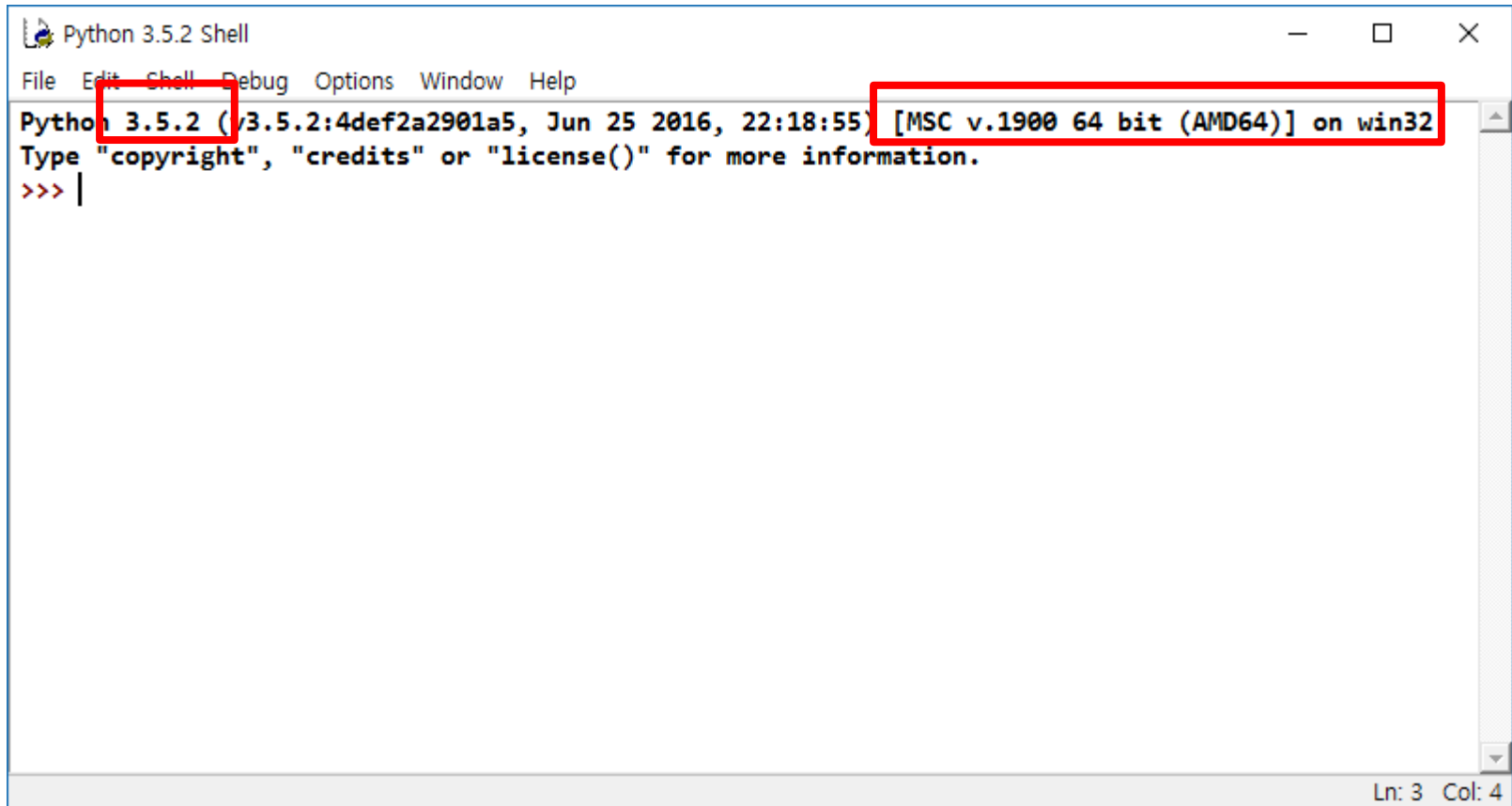
# IDLE의 실행





## 설치 확인 포인트 (1)

- 32bit 로 설치했으면, IDLE 실행될 때, 설치한 버전명 그리고 32bit 로 표시
- 64bit 로 설치했으면, IDLE 실행될 때, 설치한 버전명 그리고 64bit(AMD64)로 표시

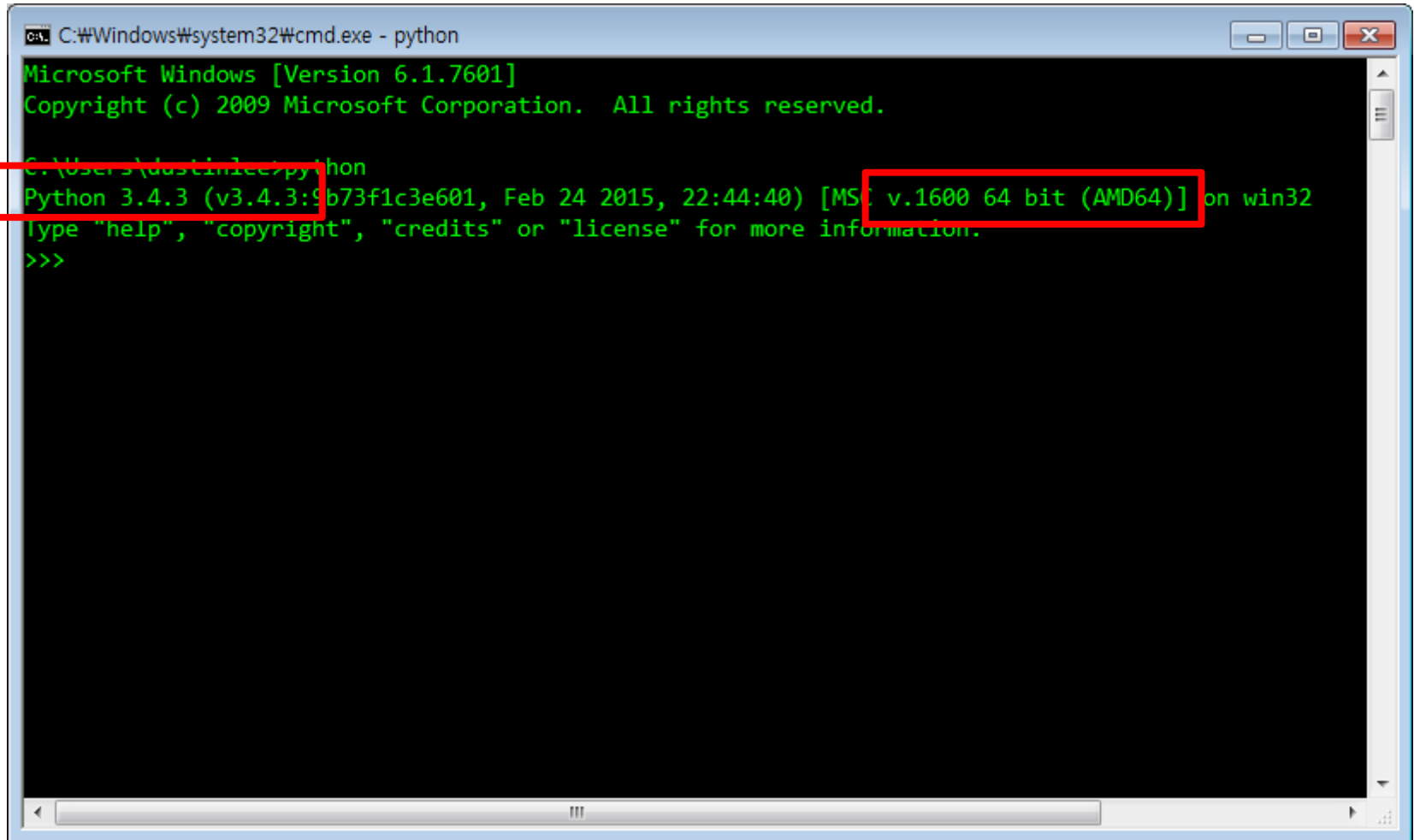


The screenshot shows a 'Python 3.5.2 Shell' window. The title bar is 'Python 3.5.2 Shell'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area displays the following information: 'Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32'. The version '3.5.2' and the architecture '[MSC v.1900 64 bit (AMD64)]' are highlighted with red boxes. Below this, it says 'Type "copyright", "credits" or "license()" for more information.' and the prompt '>>> |' is visible. The status bar at the bottom right shows 'Ln: 3 Col: 4'.

```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

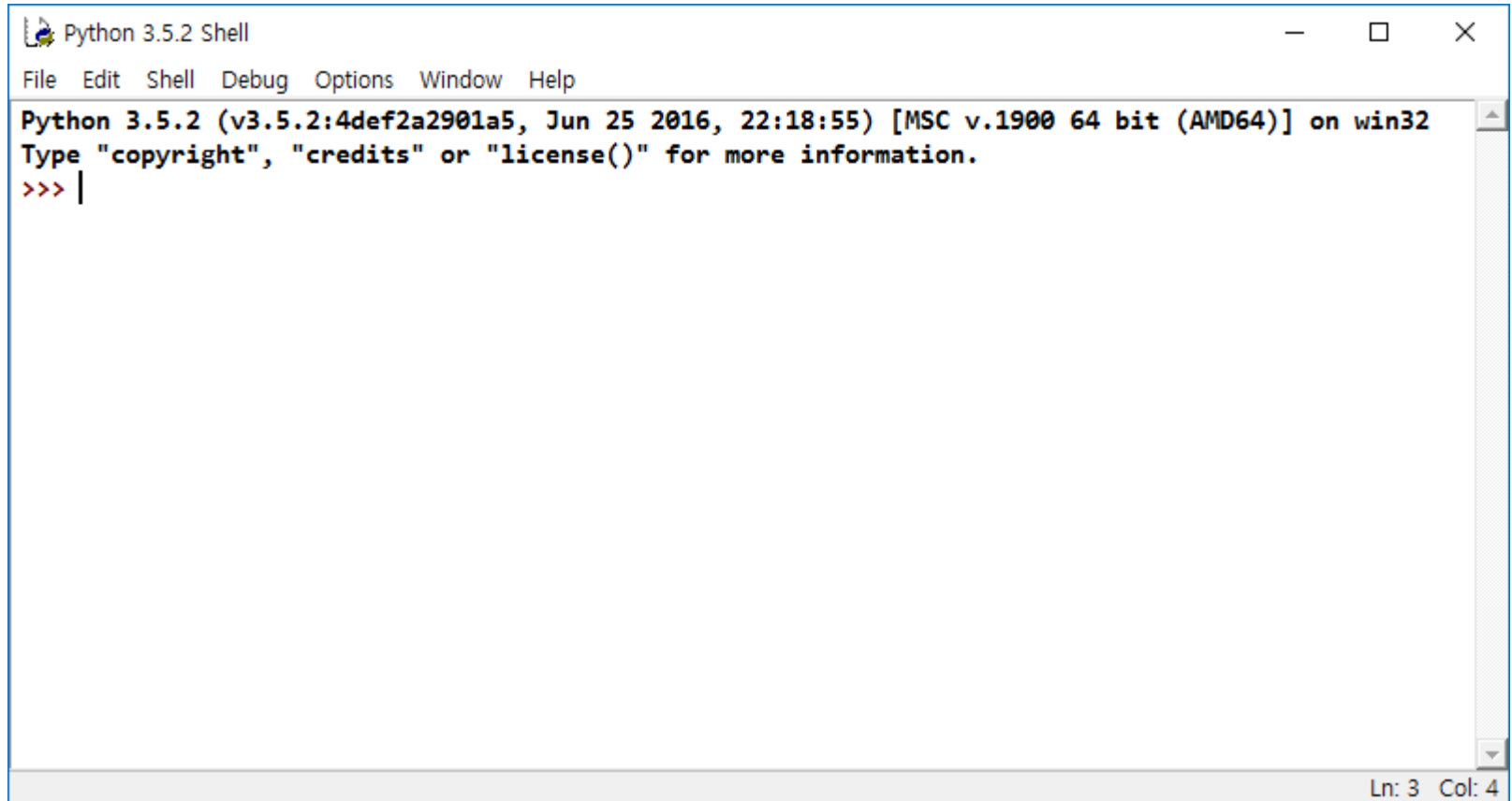
## 설치 확인 포인트 (2)

- cmd 창을 열어서, python 이라고 치면, 앞서 IDLE 을 실행했을 때와 동일하게 32bit 또는 64 bit가 나와야 함.

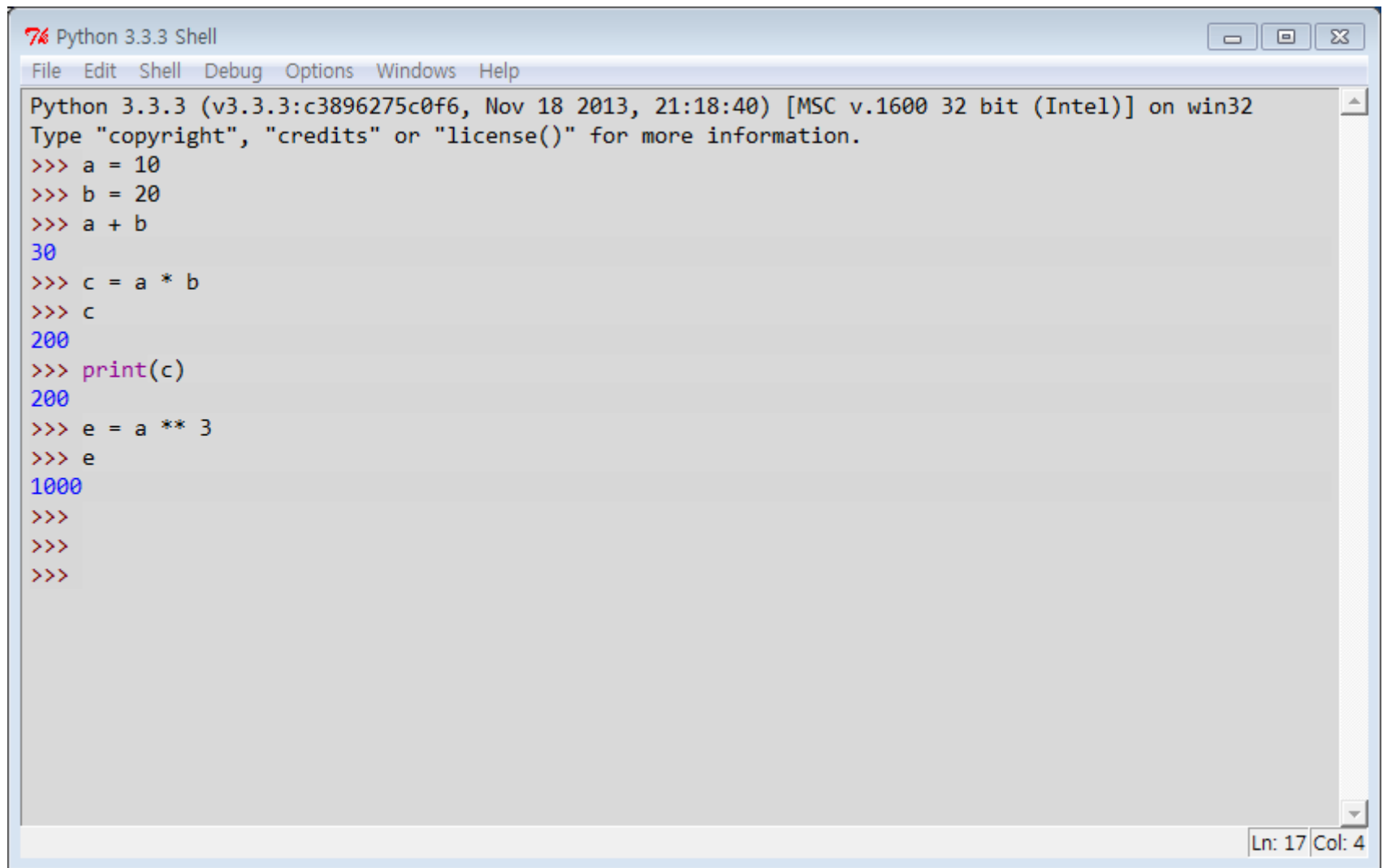


```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\dustinlee>python
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:44:40) [MSI v.1600 64 bit (AMD64)] on win32
type "help", "copyright", "credits" or "license" for more information.
>>>
```



```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:18:55) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
Ln: 3 Col: 4
```

A screenshot of a Python 3.3.3 Shell window. The window has a title bar with a red icon and the text 'Python 3.3.3 Shell'. Below the title bar is a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Windows', and 'Help'. The main area is a text editor with a light gray background. It contains the following text: 'Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32', 'Type "copyright", "credits" or "license()" for more information.', and several lines of Python code: '>>> a = 10', '>>> b = 20', '>>> a + b', '30', '>>> c = a \* b', '>>> c', '200', '>>> print(c)', '200', '>>> e = a \*\* 3', '>>> e', '1000', and three empty lines '>>>'. The status bar at the bottom right shows 'Ln: 17 Col: 4'.

```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a = 10
>>> b = 20
>>> a + b
30
>>> c = a * b
>>> c
200
>>> print(c)
200
>>> e = a ** 3
>>> e
1000
>>>
>>>
>>>
Ln: 17 Col: 4
```

Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32

Type "copyright", "credits" or "license()" for more information.

```
>>> first = 'Dae Hyun'
>>> space = " "
>>> last = "Lee"
>>> name = first + space + last
>>> name
'Dae Hyun Lee'
>>> print(name)
Dae Hyun Lee
>>>
>>> name * 2
'Dae Hyun LeeDae Hyun Lee'
>>> name * 3
'Dae Hyun LeeDae Hyun LeeDae Hyun Lee'
>>>
>>>
>>> name[0]
'D'
>>> name[2]
'e'
>>> name[3:8]
' Hyun'
>>> name[-1]
'e'
>>> name[0:5:2]
'DeH'
>>> name[-1::-1]
'eeL nuyH eaD'
>>> |
```

# List

```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>> gg = ['윤아', '태연', '써니', '티파니']
>>> gg
['윤아', '태연', '써니', '티파니']
>>> gg[0]
'윤아'
>>> gg[0:2]
['윤아', '태연']
>>> gg = gg + ['서현', '제시카']
>>> gg
['윤아', '태연', '써니', '티파니', '서현', '제시카']
>>> gg.append('유리')
>>> gg
['윤아', '태연', '써니', '티파니', '서현', '제시카', '유리']
>>> gg.sort()
>>> gg
['서현', '써니', '유리', '윤아', '제시카', '태연', '티파니']
>>> len(gg)
7
>>>|
```

Ln: 22 Col: 4

# Dictionary

```
>>> phone = {'이대현': '01097881745', '박근혜' : '01012345678', '이건희' : '01077774949'}

>>> print(phone)
{'이대현': '01097881745', '이건희': '01112341234', '박근혜': '01012345678'}

>>> print(phone['이대현'])
'01097881745'

>>> print(phone.keys())
dict_keys(['이대현', '이건희', '박근혜'])

>>> print(phone.values())
dict_values(['01097881745', '01112341234', '01012345678'])
```

## 해당 Key가 있는지 조사 (in)

```
>>> a = {'name': 'pey', 'phone': '0119993323', 'birth': '1118'}  
>>> 'name' in a  
True  
>>> 'email' in a  
False
```



```
If <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
elif <조건문>:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...  
...  
else:  
    <수행할 문장1>  
    <수행할 문장2>  
    ...
```

비교연산자	설명
$x < y$	x가 y보다 작다
$x > y$	x가 y보다 크다
$x == y$	x와 y가 같다
$x != y$	x와 y가 같지 않다
$x \geq y$	x가 y보다 크거나 같다
$x \leq y$	x가 y보다 작거나 같다

연산자	설명
x or y	x와 y 둘중에 하나만 참이면 참이다
x and y	x와 y 모두 참이어야 참이다
not x	x가 거짓이면 참이다

```
>>> money = 2000
>>> watch = 1
>>> if money >= 3000 or watch:
...     print("택시를 타고 가라")
... else:
...     print("걸어가라")
...
택시를 타고 가라
>>>
```

```
>>> 1 in [1, 2, 3]
True
>>> 1 not in [1, 2, 3]
False
```

```
>>> 'a' in ('a', 'b', 'c')
True
>>> 'j' not in 'python'
True
```

# while

```
while <조건문>:
```

```
    <수행할 문장1>
```

```
    <수행할 문장2>
```

```
    <수행할 문장3>
```

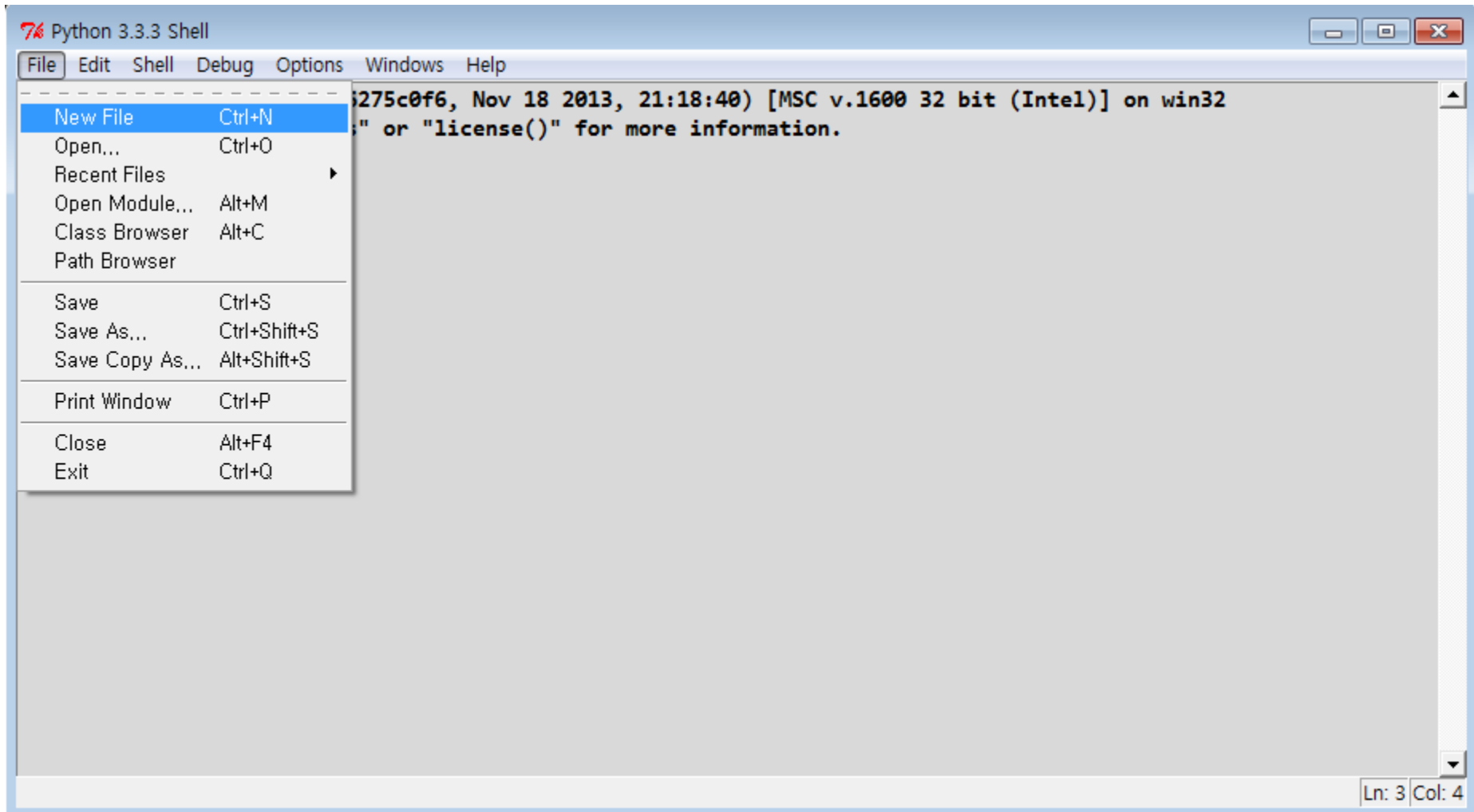
```
    ...
```

```

coffee = 10
while 1:
    money = int(input("돈을 넣어 주세요: "))
    if money == 300:
        print("커피를 줍니다.")
        coffee = coffee - 1
    elif money > 300:
        print("거스름돈 %d를 주고 커피를 줍니다." % (money - 300))
        coffee = coffee - 1
    else:
        print("돈을 다시 돌려주고 커피를 주지 않습니다.")
        print("남은 커피의 양은 %d개 입니다." % coffee)
    if not coffee:
        print("커피가 다 떨어졌습니다. 판매를 중지 합니다.")
        break

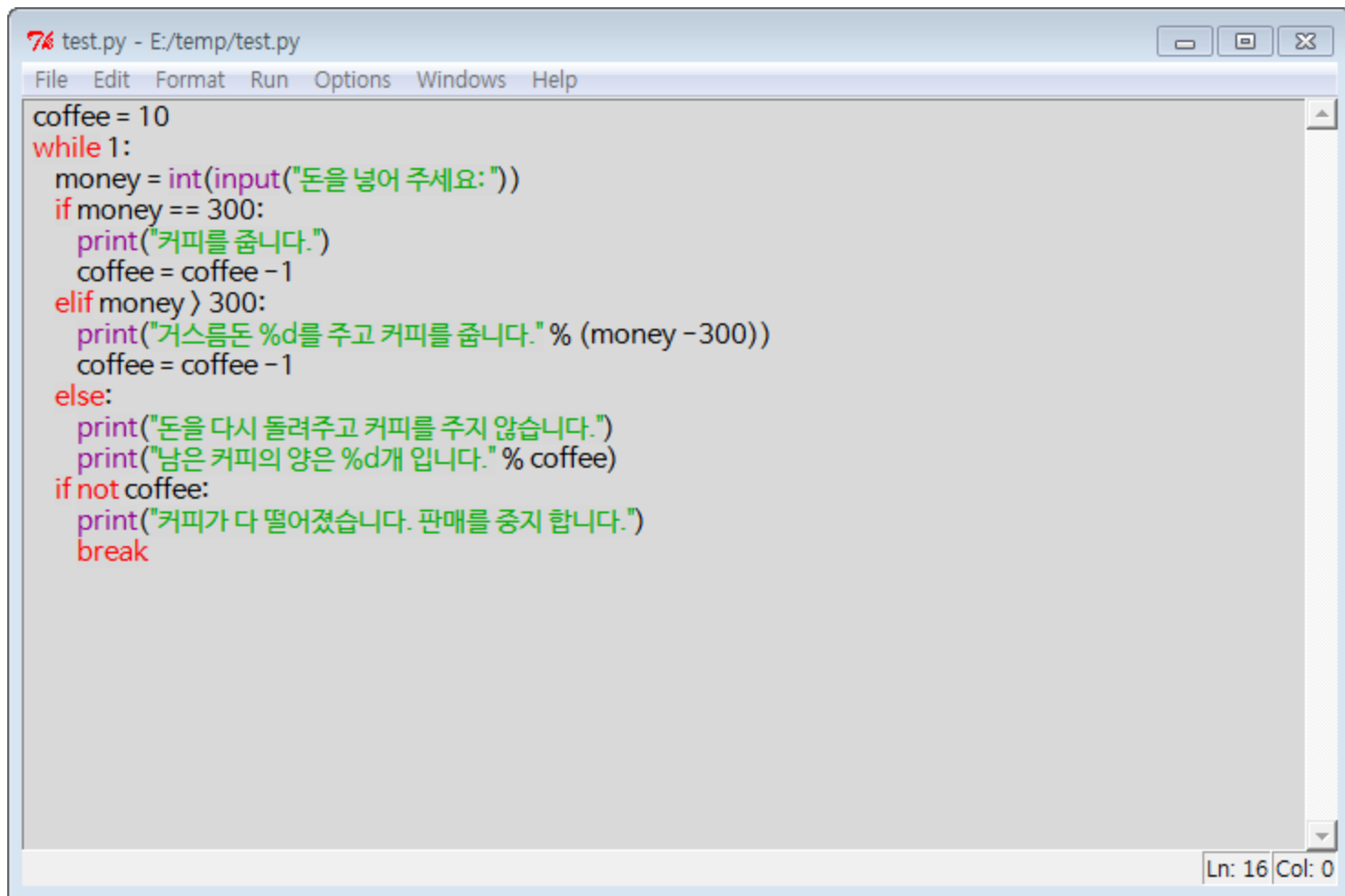
```

# Python code를 file로 생성





# test.py로 저장

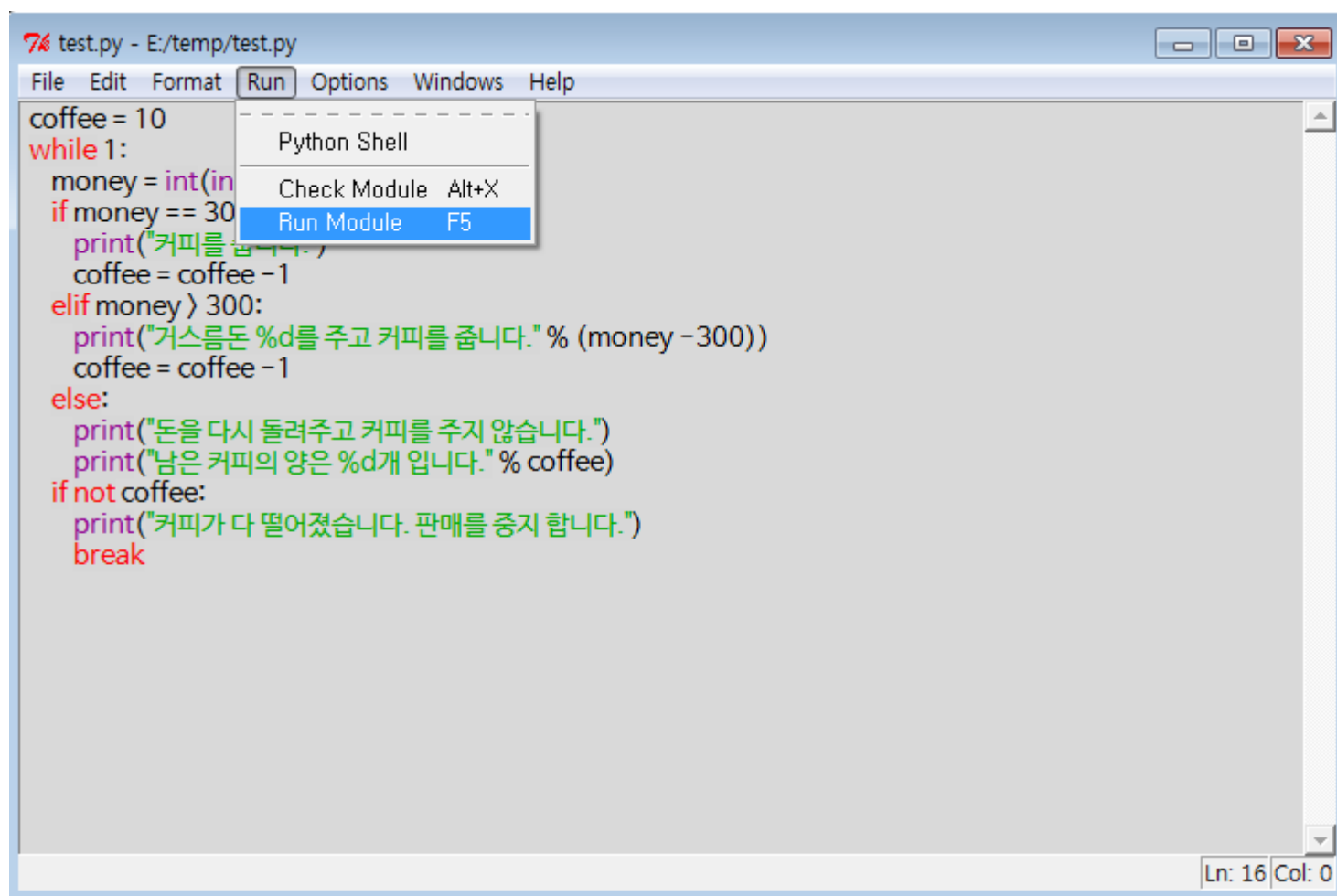


The screenshot shows a Python IDE window titled "test.py - E:/temp/test.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code is a Python script for a coffee vending machine simulation. It starts with a variable 'coffee' set to 10. A 'while' loop runs indefinitely. Inside the loop, it prompts the user to enter money. If the entered money is exactly 300, it prints "커피를 줍니다.", decrements 'coffee' by 1, and continues. If the entered money is greater than 300, it prints a message about giving change and the coffee, decrements 'coffee' by 1, and continues. Otherwise, it prints a message about returning the money and not giving coffee. If 'coffee' reaches 0, it prints a message about the coffee being sold out and breaks the loop. The status bar at the bottom right shows "Ln: 16 Col: 0".

```
test.py - E:/temp/test.py
File Edit Format Run Options Windows Help

coffee = 10
while 1:
    money = int(input("돈을 넣어 주세요: "))
    if money == 300:
        print("커피를 줍니다.")
        coffee = coffee - 1
    elif money > 300:
        print("거스름돈 %d를 주고 커피를 줍니다." % (money - 300))
        coffee = coffee - 1
    else:
        print("돈을 다시 돌려주고 커피를 주지 않습니다.")
        print("남은 커피의 양은 %d개 입니다." % coffee)
    if not coffee:
        print("커피가 다 떨어졌습니다. 판매를 중지 합니다.")
        break

Ln: 16 Col: 0
```



test.py - E:/temp/test.py

File Edit Format Run Options Windows Help

```
coffee = 10
while 1:
    money = int(input("돈을 넣어주세요: "))
    if money == 300:
        print("커피를 드립니다.")
        coffee = coffee - 1
    elif money > 300:
        print("거스름돈 %d를 주고 커피를 줍니다." % (money - 300))
        coffee = coffee - 1
    else:
        print("돈을 다시 돌려주고 커피를 주지 않습니다.")
        print("남은 커피의 양은 %d개 입니다." % coffee)
    if not coffee:
        print("커피가 다 떨어졌습니다. 판매를 중지 합니다.")
        break
```

Python Shell  
Check Module Alt+X  
Run Module F5

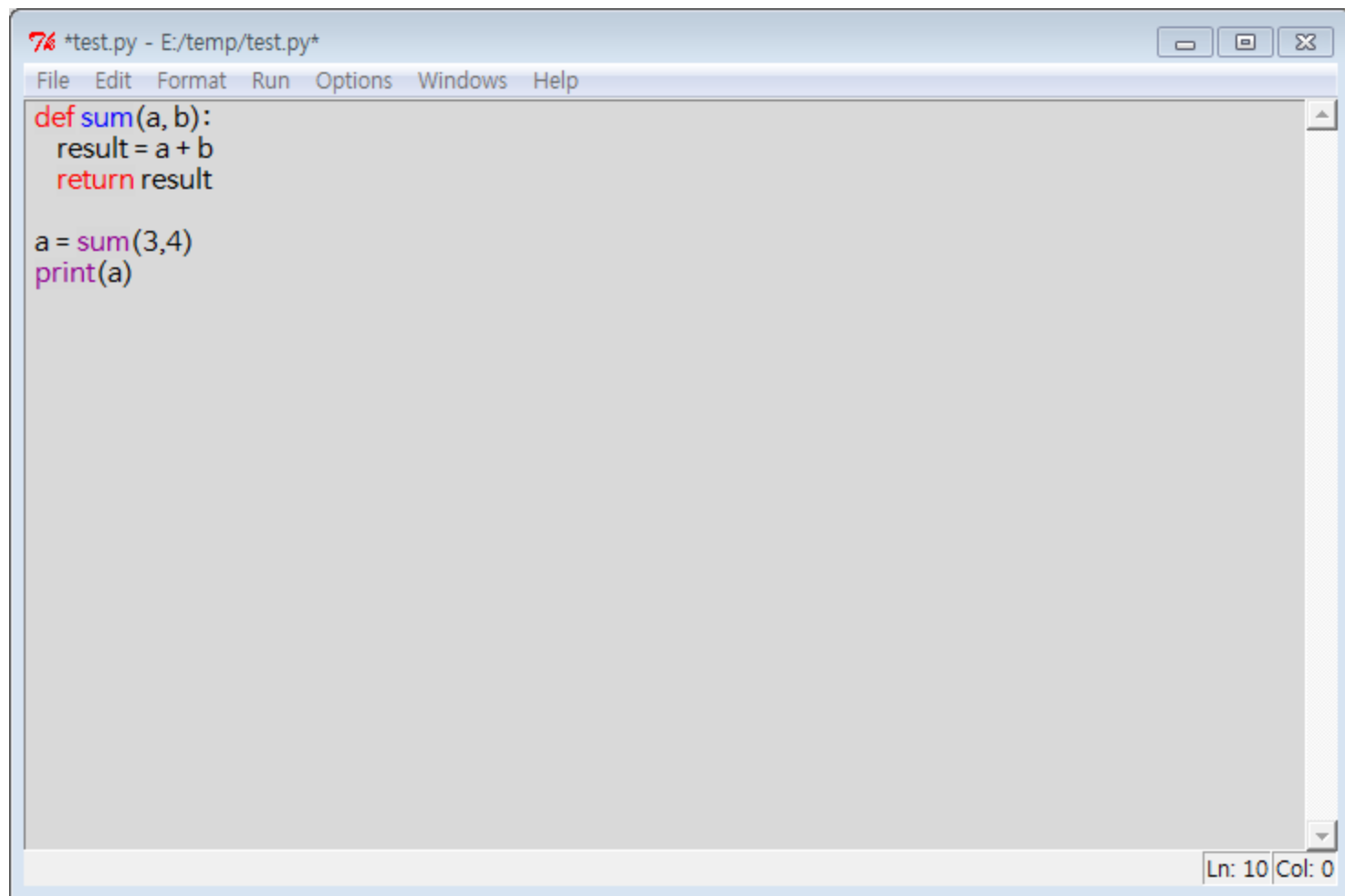
Ln: 16 Col: 0

```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>>
>>> ===== RESTART =====
>>>
돈을 넣어 주세요: 400
거스름돈 100를 주고 커피를 줍니다.
돈을 넣어 주세요: 200
돈을 다시 돌려주고 커피를 주지 않습니다.
남은 커피의 양은 9개 입니다.
돈을 넣어 주세요: 300
커피를 줍니다.
돈을 넣어 주세요: 300
커피를 줍니다.
돈을 넣어 주세요: 300
커피를 줍니다.
돈을 넣어 주세요: 300
커피를 줍니다.
돈을 넣어 주세요: 300
커피를 줍니다.
돈을 넣어 주세요: 300
커피를 줍니다.
돈을 넣어 주세요: 300
커피를 줍니다.
돈을 넣어 주세요: 300
커피를 줍니다.
커피가 다 떨어졌습니다. 판매를 중지 합니다.
>>>
>>>
```

Ln: 17 Col: 0

# 함수(function)

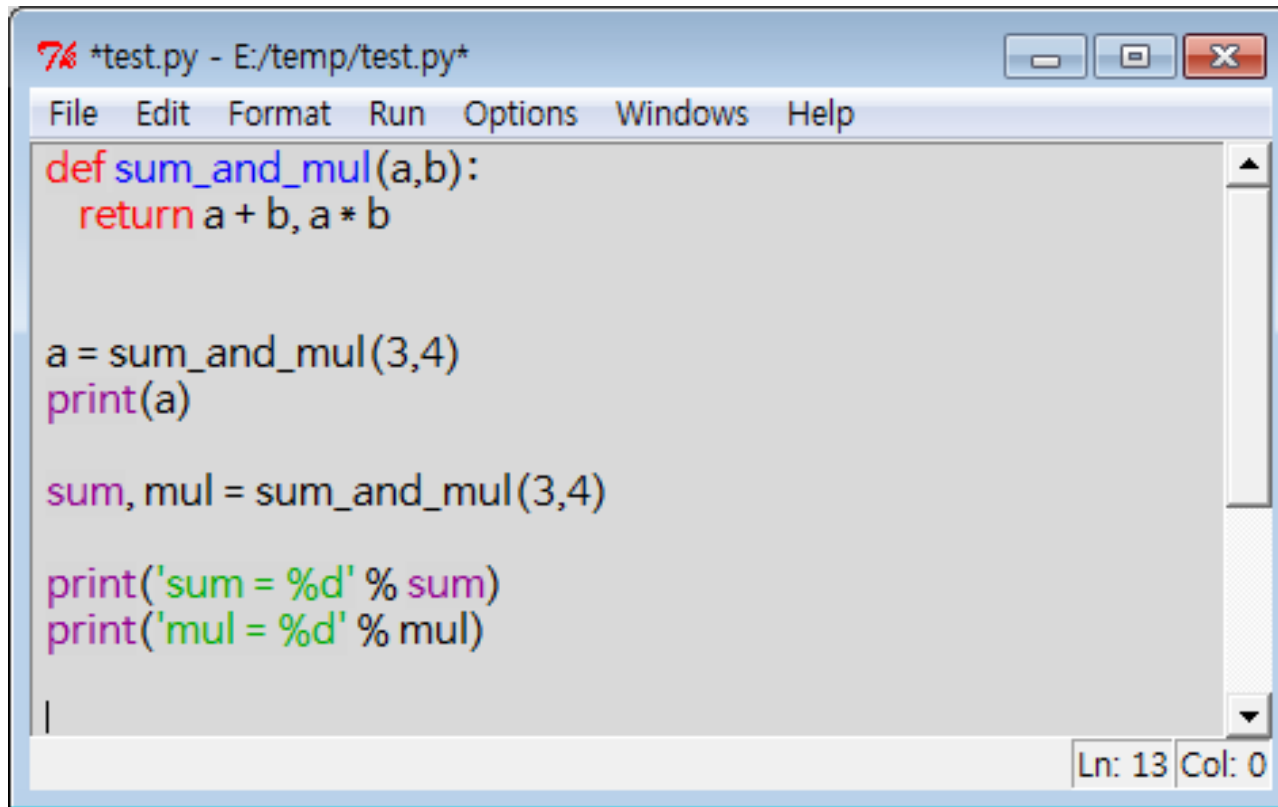
```
def 함수명(입력 인수):  
    <수행할 문장1>  
    <수행할 문장2>  
    ...
```



The image shows a screenshot of a Python IDE window. The title bar reads "74 \*test.py - E:/temp/test.py\*". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The code editor contains the following Python code:

```
def sum(a, b):  
    result = a + b  
    return result  
  
a = sum(3,4)  
print(a)
```

The status bar at the bottom right indicates "Ln: 10 Col: 0".



```
*test.py - E:/temp/test.py*
File Edit Format Run Options Windows Help

def sum_and_mul(a,b):
    return a + b, a * b

a = sum_and_mul(3,4)
print(a)

sum, mul = sum_and_mul(3,4)

print('sum = %d' % sum)
print('mul = %d' % mul)

Ln: 13 Col: 0
```

```
class Simple:  
    pass
```

```
a = Simple()
```



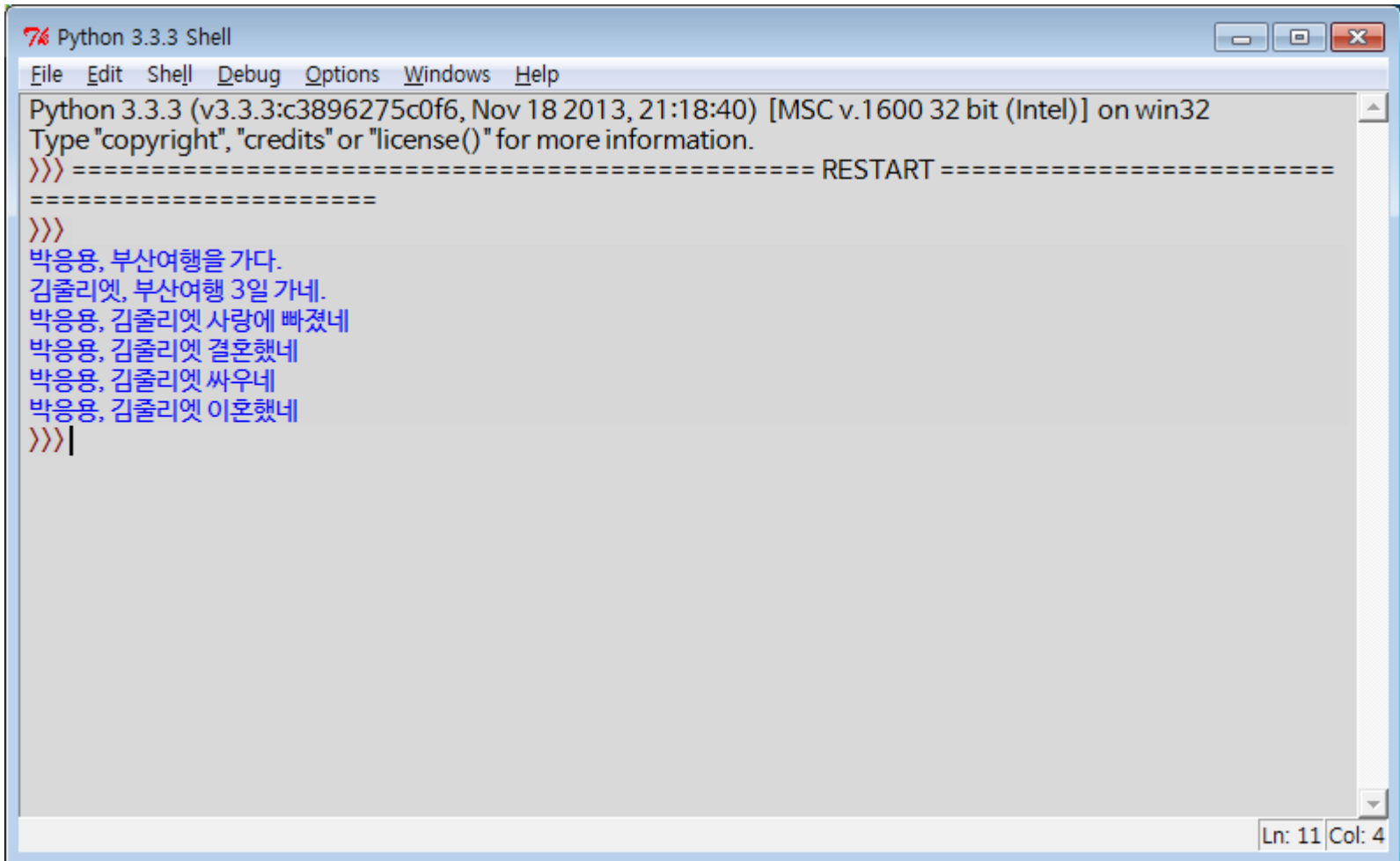
```
*test.py - E:/temp/test.py*
File Edit Format Run Options Windows Help

class HousePark:
    lastname = "박"
    def __init__(self, name):
        self.fullname = self.lastname + name
    def travel(self, where):
        print("%s, %s여행을 가다." % (self.fullname, where))
    def love(self, other):
        print("%s, %s 사랑에 빠졌네" % (self.fullname, other.fullname))
    def fight(self, other):
        print("%s, %s 싸우네" % (self.fullname, other.fullname))
    def __add__(self, other):
        print("%s, %s 결혼했네" % (self.fullname, other.fullname))
    def __sub__(self, other):
        print("%s, %s 이혼했네" % (self.fullname, other.fullname))
    def __del__(self):
        print("%s 죽네" % self.fullname)

class HouseKim(HousePark):
    lastname = "김"
    def travel(self, where, day):
        print("%s, %s여행 %d일 가네." % (self.fullname, where, day))

pey = HousePark("응용")
juliet = HouseKim("줄리엣")
pey.travel("부산")
juliet.travel("부산", 3)
pey.love(juliet)
pey + juliet
pey.fight(juliet)
pey - juliet

Ln: 36 Col: 0
```



A screenshot of a Python 3.3.3 Shell window. The window has a title bar with the text "Python 3.3.3 Shell" and standard Windows window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following content:

```
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>> 박응용, 부산여행을 가다.
>>> 김줄리엣, 부산여행 3일 가네.
>>> 박응용, 김줄리엣 사랑에 빠졌네
>>> 박응용, 김줄리엣 결혼했네
>>> 박응용, 김줄리엣 싸우네
>>> 박응용, 김줄리엣 이혼했네
>>>|
```

The status bar at the bottom right indicates "Ln: 11 Col: 4".