

Lecture #1. 파이썬 기초 (1)

2D 게임 프로그래밍

이대현 교수

github - Google Search

python - Google Search

start [이대현의 게임프로

← → ↻

www.dae Hyunlee.com/doku.php

☆

start

blog

personal

projects

incognitek

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

 종합설계기획 (3학년2학기)

프로젝트보고

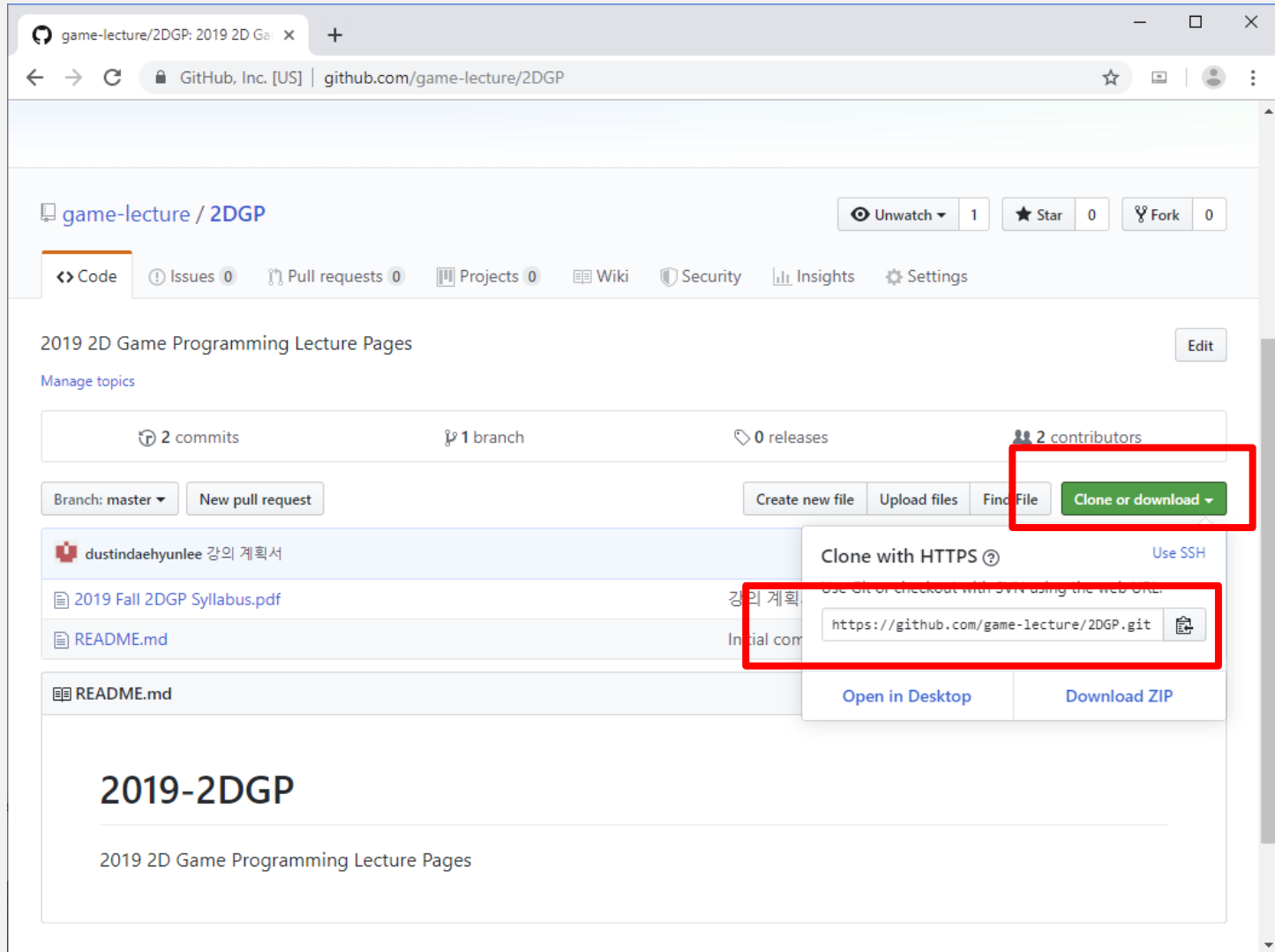
 종합설계(III) (4학년2학기)

2D 게임 프로그래밍

Copyright by 이대현

2DGP Github 저장소

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



Git 의 설치

The screenshot shows the Git website (git-scm.com) in a web browser. The page features the Git logo and the tagline "--local-branching-on-the-cheap". A search bar is located in the top right corner. The main content area describes Git as a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. It also mentions that Git is easy to learn and has a tiny footprint with lightning fast performance, outclassing SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows. A section titled "Learn Git in your browser for free with Try Git." is also present. Below this, there are four sections: "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.), and "Community" (Get involved! Bug reporting, mailing list, chat, development and more.). On the right side, there is a section for the "Latest source Release 2.9.3" with "Release Notes (2016-08-12)" and a button for "Downloads for Windows". A diagram on the right side of the page illustrates the Git branching model with stacks of code and colored lines representing branches and merges.

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.

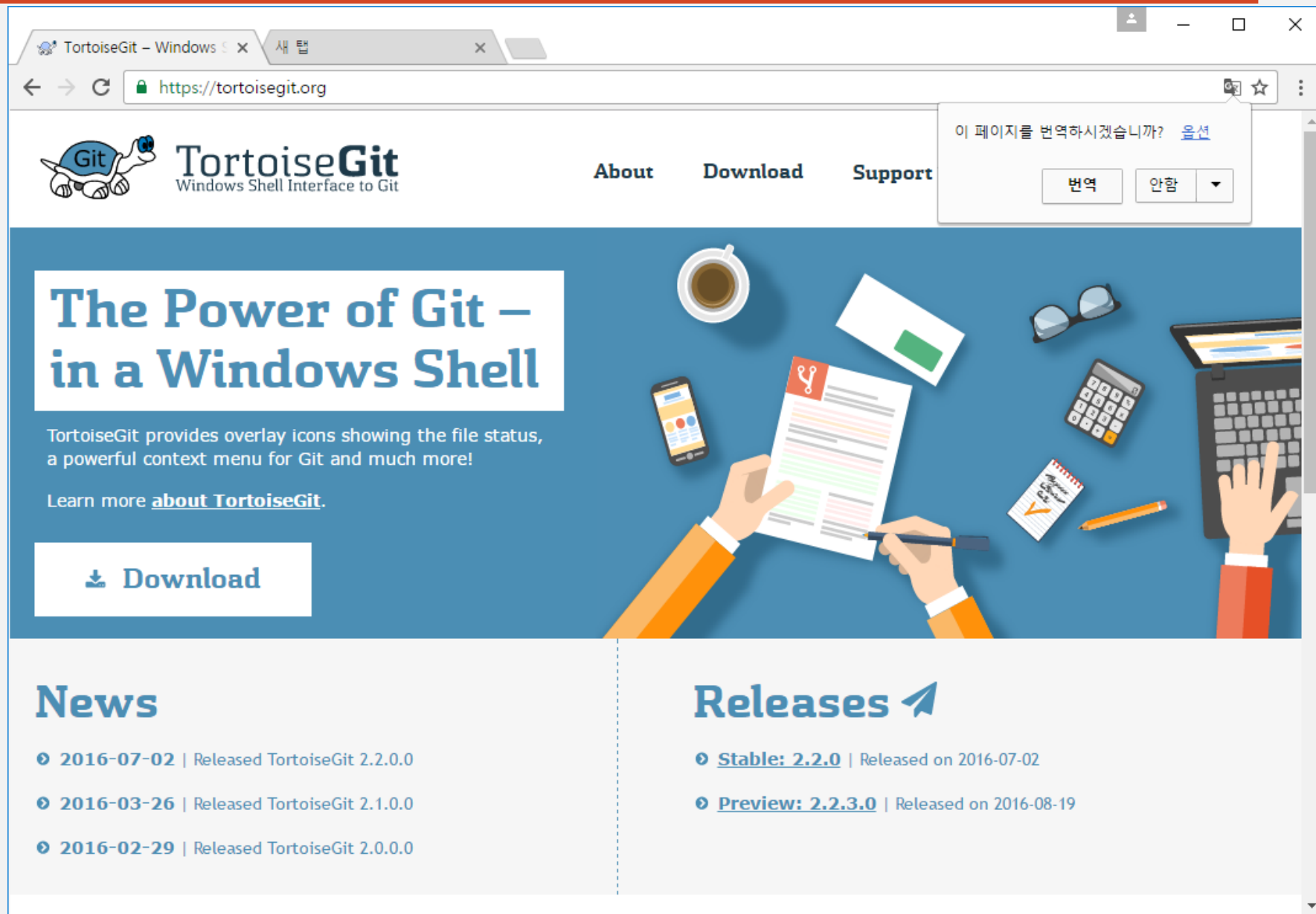
Documentation
Command reference pages, Pro Git book content, videos and other material.

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](#)

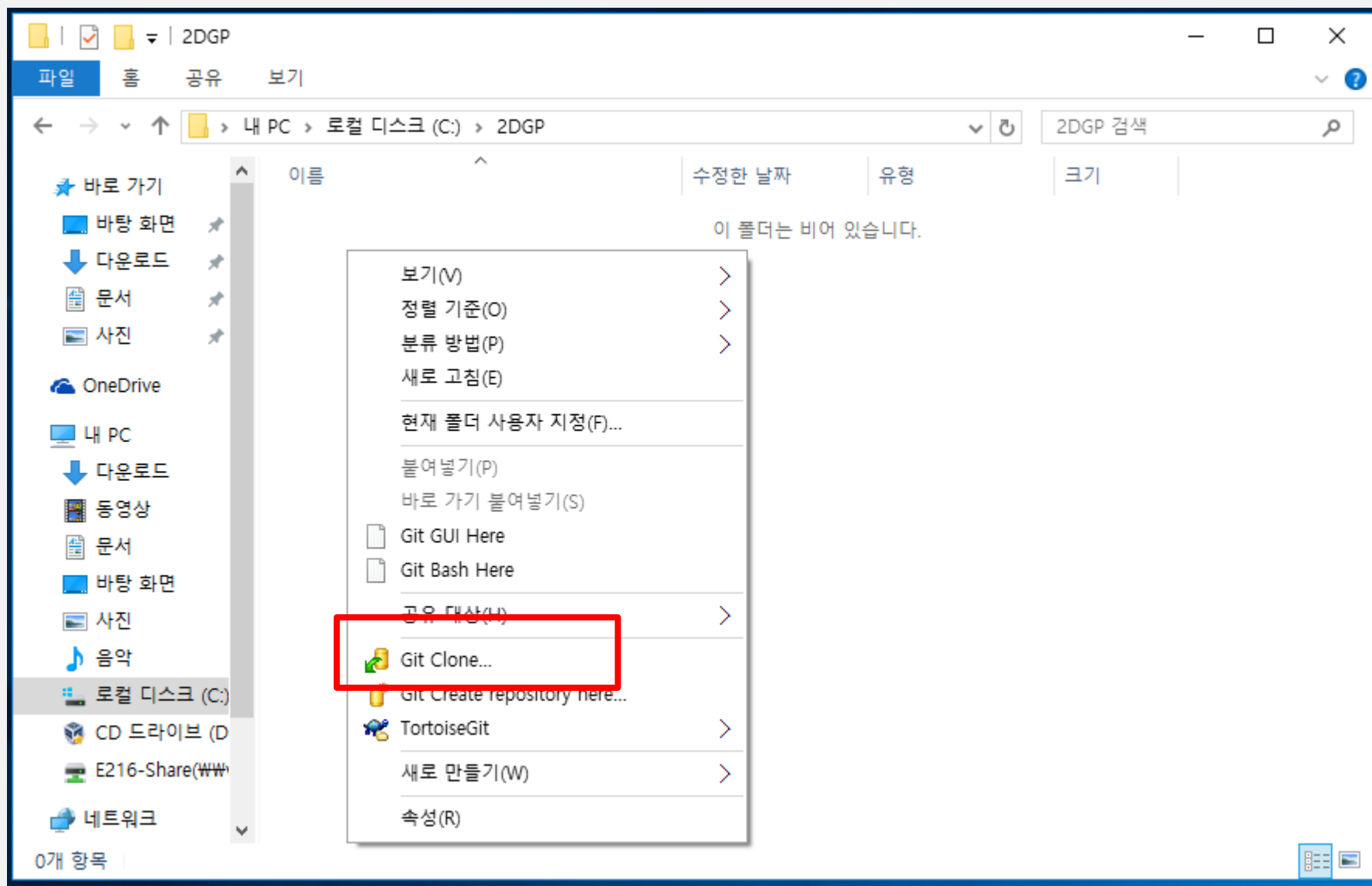
News

- 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 – 서버에 있는 코드를 최초로 내려받을 때



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

Git clone - TortoiseGit

Clone Existing Repository

URL: Browse...

Directory: Browse...

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

☐ Branch ☐ Origin Name ☐ LFS

☐ Load Putty Key ...

From SVN Repository

☐ From SVN Repository

☐ Trunk: ☐ Tags: ☐ Branch:

☐ From: ☐ Username:

OK Cancel Help



remote: Compressing objects



```
git.exe clone --progress -v "https://github.com/game-lecture/2DGP.git" "D:\2DGP"
```

Cloning into 'D:\2DGP'...

POST git-upload-pack (175 bytes)

remote: Enumerating objects: 6, done.

remote: Counting objects: 100% (6/6), done.

remote: Compressing objects: 100% (5/5), done.

remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0

Success (3093 ms @ 2019-09-10 오후 6:50:31)

Show log



Close

Abort

파이썬의 특징

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

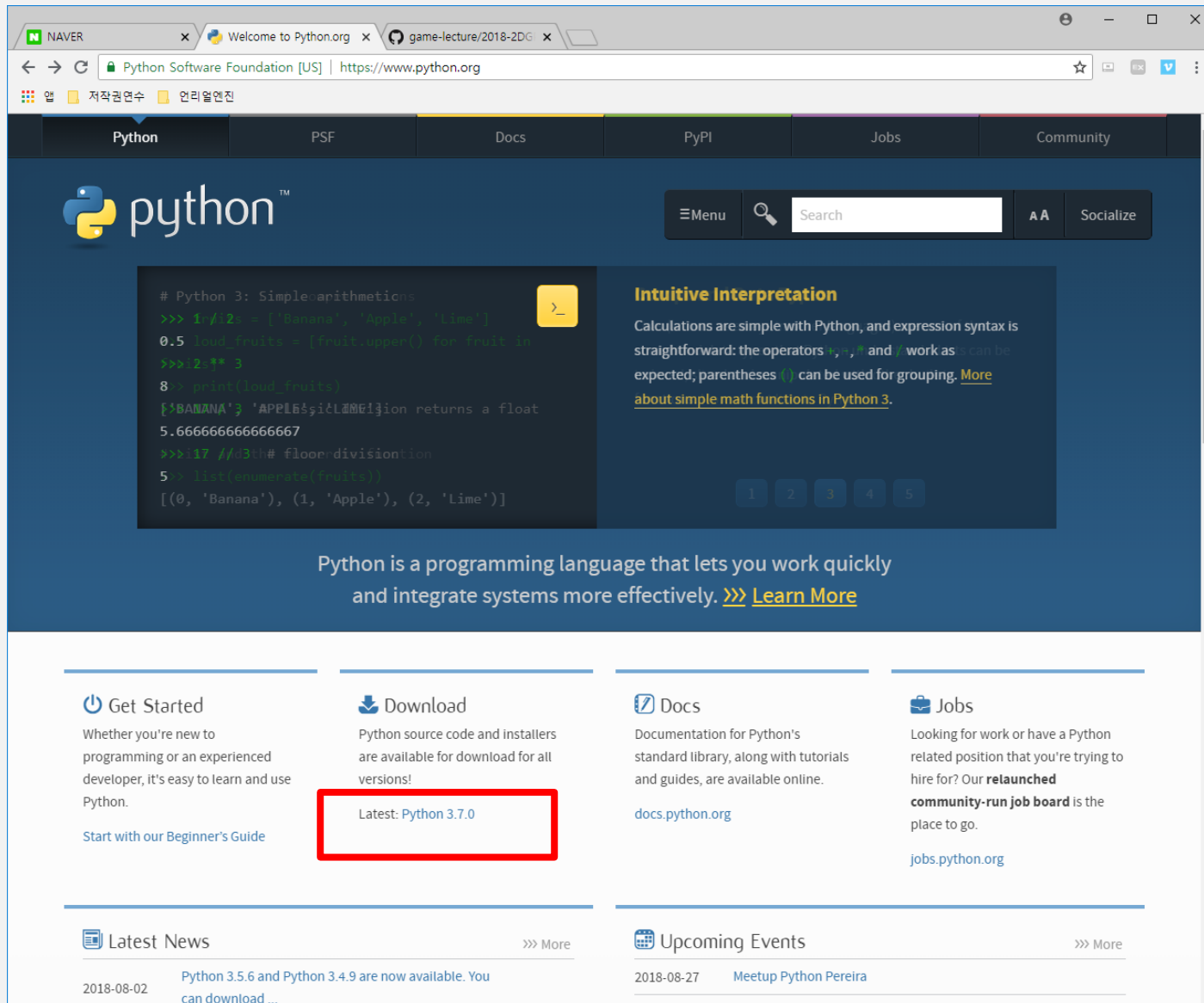
마지막으로 다음의 재미있는 말로 파이썬의 특징을 마무리하려 한다.

Life is too short, You need python.

Python Key Words

False	class	return	is	finally
None	if	for	lambda	continue
True	def	from	while	nonlocal
and	del	global	not	with
as	elif	try	or	yield
assert	else	import	pass	
break	except	in	raise	

파이썬 홈페이지 – <https://www.python.org>



The screenshot shows the Python.org homepage in a web browser. The browser's address bar displays the URL <https://www.python.org>. The page features a dark blue header with the Python logo and navigation links: Python, PSF, Docs, PyPI, Jobs, and Community. A search bar and social media links are also present. The main content area includes a code snippet demonstrating Python 3 arithmetic operations, a section titled "Intuitive Interpretation" explaining basic syntax, and a large blue banner stating "Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)". Below this, four columns provide links to "Get Started", "Download", "Docs", and "Jobs". The "Download" column highlights "Latest: Python 3.7.0" with a red rectangular box. The footer contains "Latest News" and "Upcoming Events" sections.

Python Software Foundation [US] | <https://www.python.org>

Python PSF Docs PyPI Jobs Community

python™

Menu Search AA Socialize

```
# Python 3: Simple arithmetics
>>> 1/2
0.5
load_fruits = [fruit.upper() for fruit in
>>> 2.1*3
8
>>> print(load_fruits)
['BANANA', 'APPLE', 'LIME']
5.666666666666667
>>> 17//3
5
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Intuitive Interpretation

Calculations are simple with Python, and expression syntax is straightforward: the operators `+`, `*`, `+` and `*` work as expected; parentheses `()` can be used for grouping. [More about simple math functions in Python 3.](#)

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 3.7.0

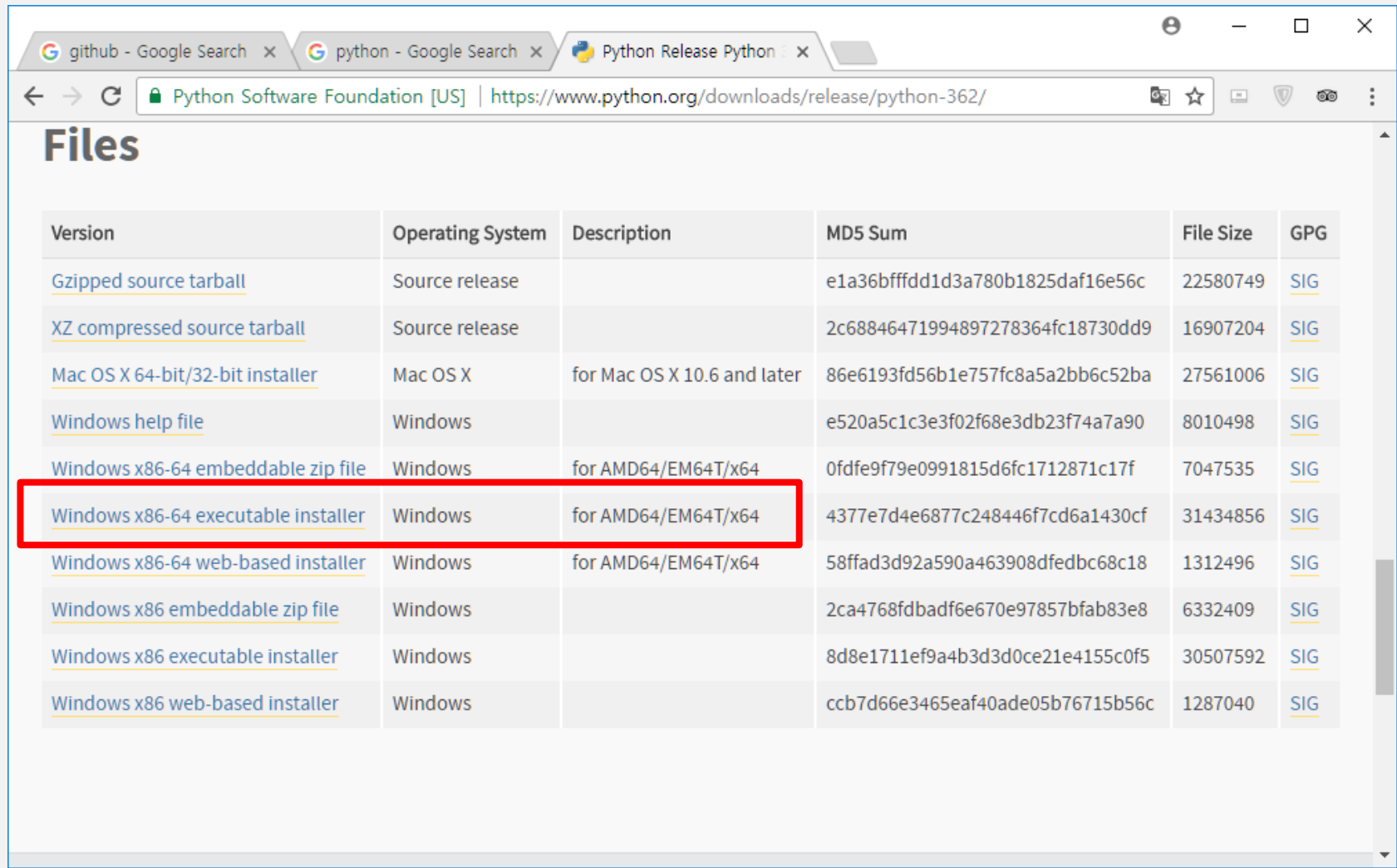
Docs
Documentation for Python's standard library, along with tutorials and guides, are available online.
docs.python.org

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

Latest News >>> More
2018-08-02 [Python 3.5.6 and Python 3.4.9 are now available. You can download ...](#)

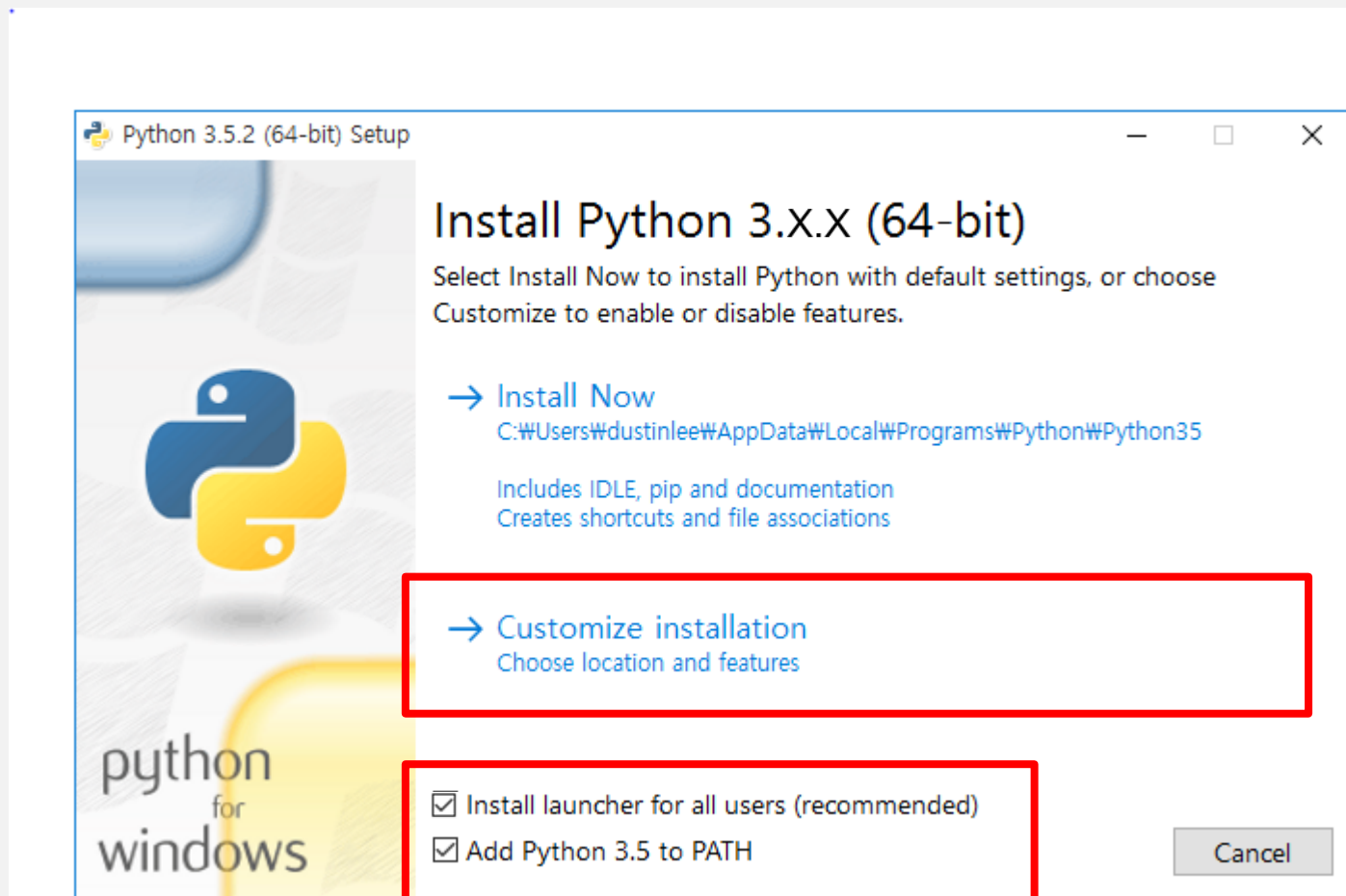
Upcoming Events >>> More
2018-08-27 [Meetup Python Pereira](#)

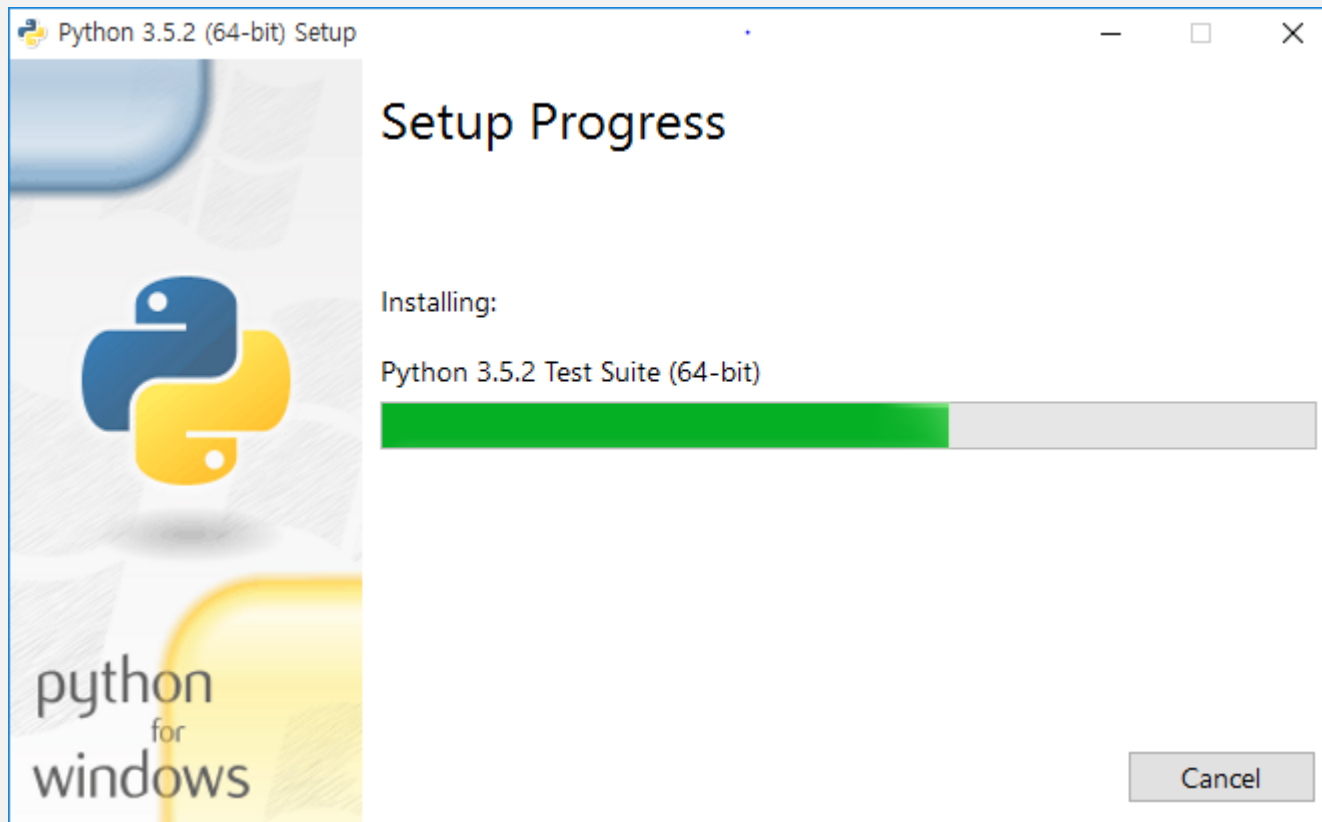
Python 최신 버전 다운로드 - 중요 ! 반드시 64bit version 을 다운로드



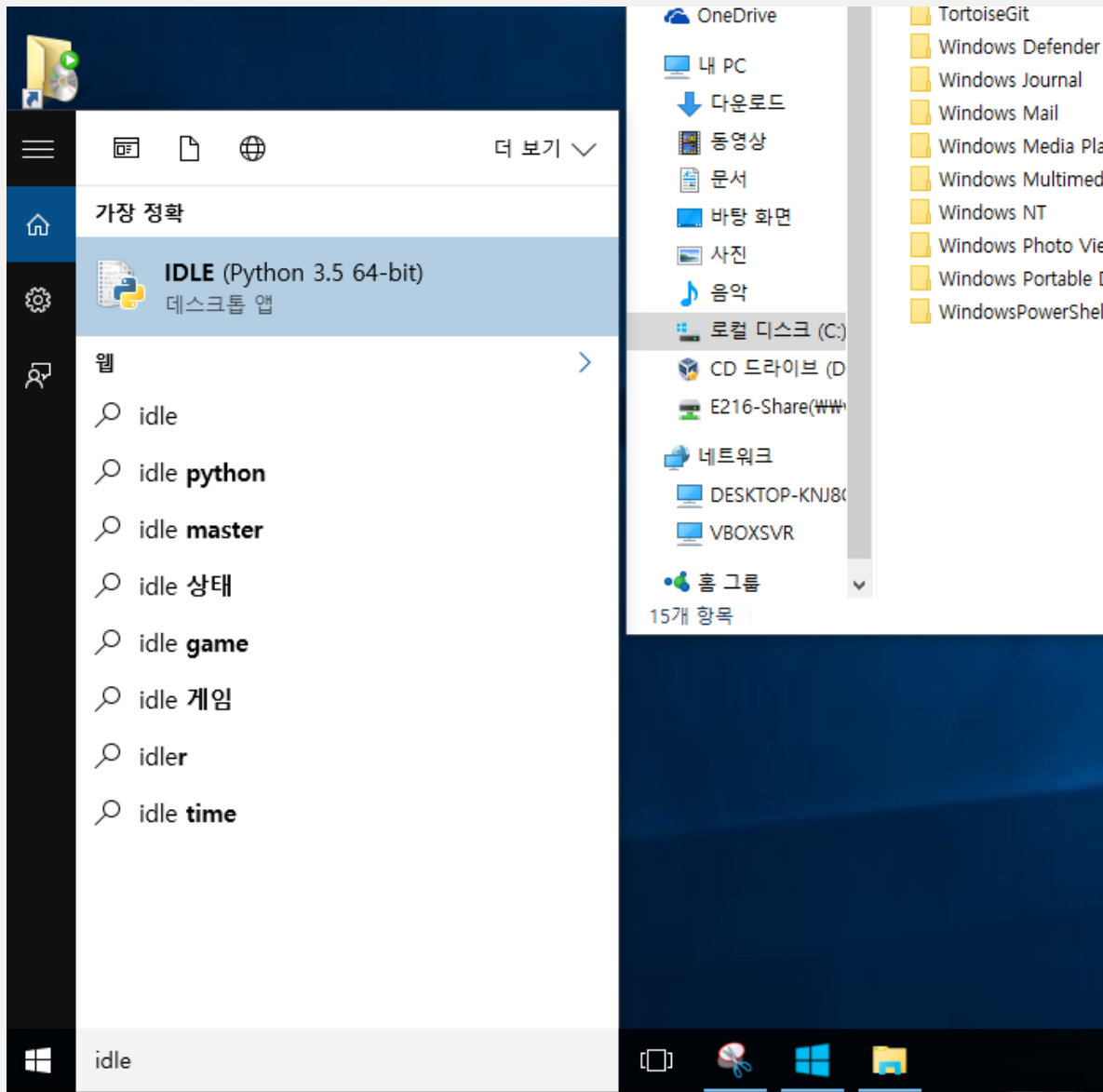
The screenshot shows the Python 3.6.2 download page. The browser tabs include 'github - Google Search', 'python - Google Search', and 'Python Release Python'. The address bar shows the URL 'https://www.python.org/downloads/release/python-362/'. The page title is 'Files'. The table lists various download options, with the 'Windows x86-64 executable installer' highlighted by a red box.

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



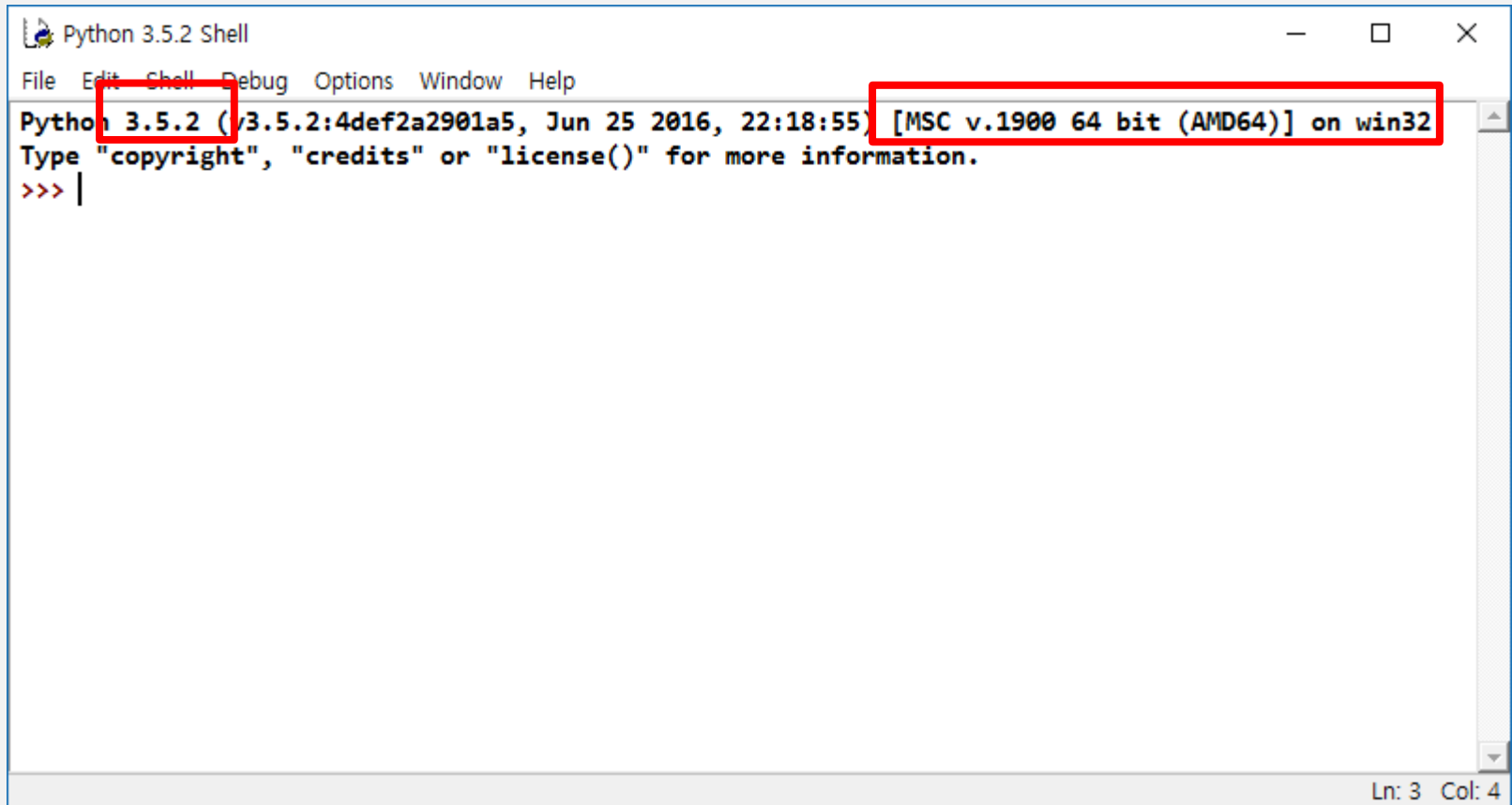


IDLE의 실행



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



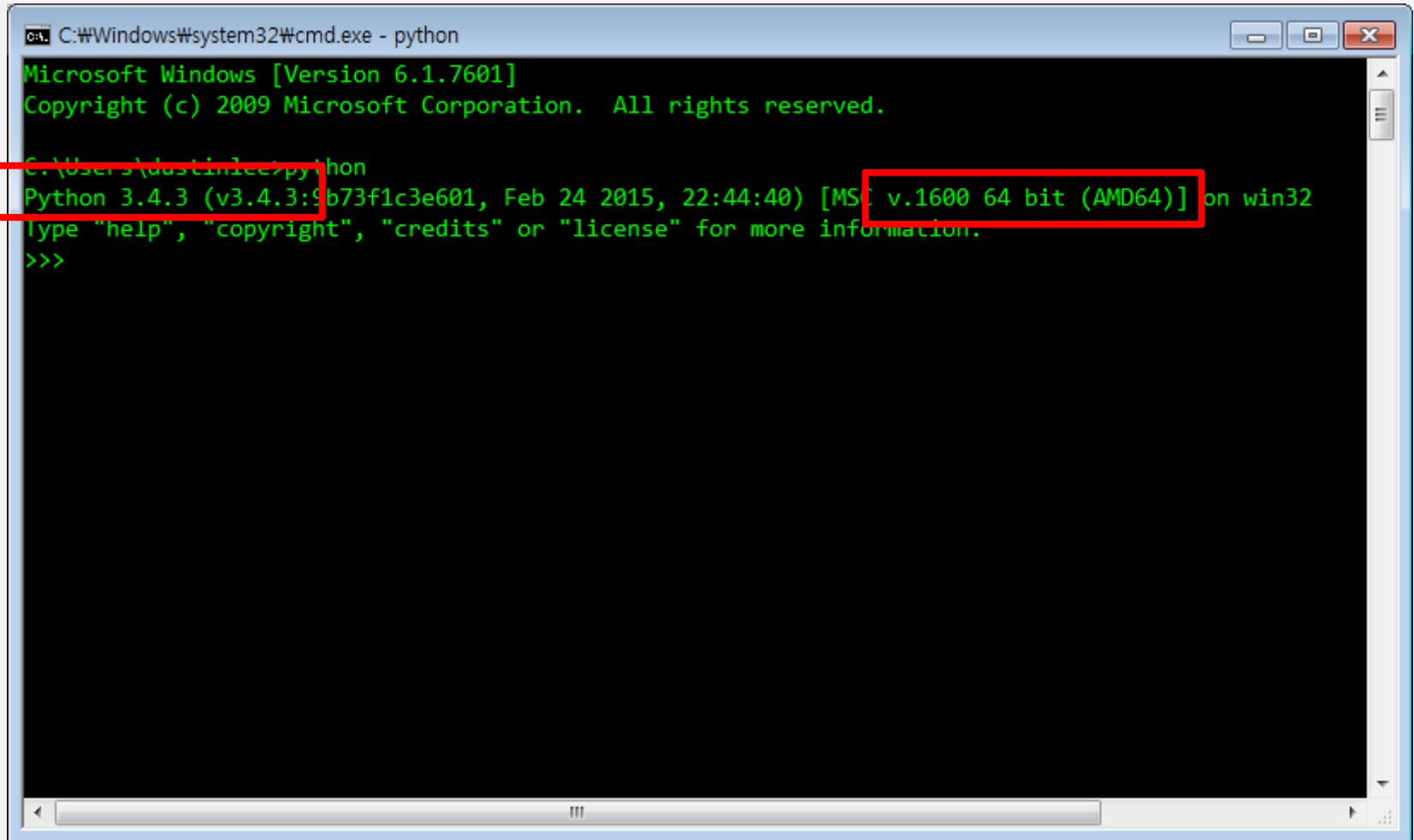
The screenshot shows the Python 3.5.2 Shell window. The title bar reads "Python 3.5.2 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following information:

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

Red boxes highlight the version number "3.5.2" and the architecture "[MSC v.1900 64 bit (AMD64)]". The status bar at the bottom right shows "Ln: 3 Col: 4".

설치 확인 포인트 (2)

- cmd 창을 열어서, python 이라고 치면, 앞서 IDLE 을 실행했을 때와 동일하게 32bit 또는 64 bit가 나와야 함. **중요! 설치한 최신 버전과 동일한 버전으로 표시되어야 함.**

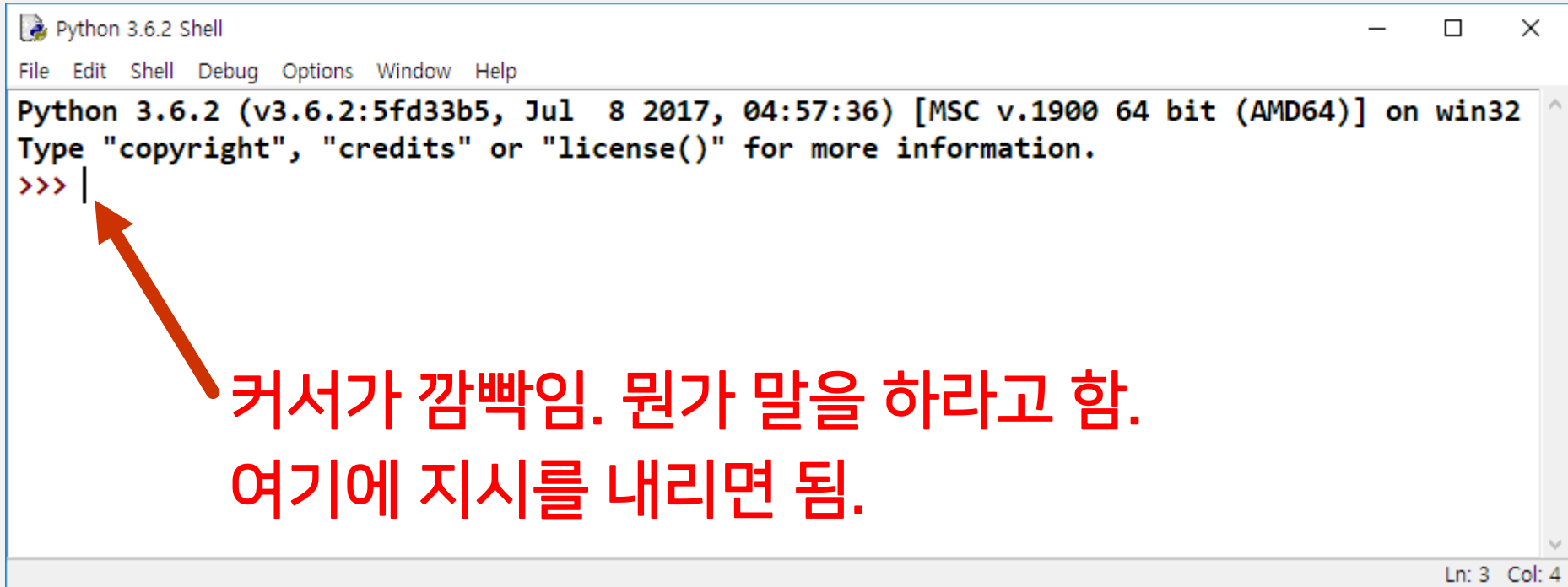


```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

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.
>>>
```

IDLE 실행 화면

- Python 언어로 지시하면, 이를 해석해서 일을 함.

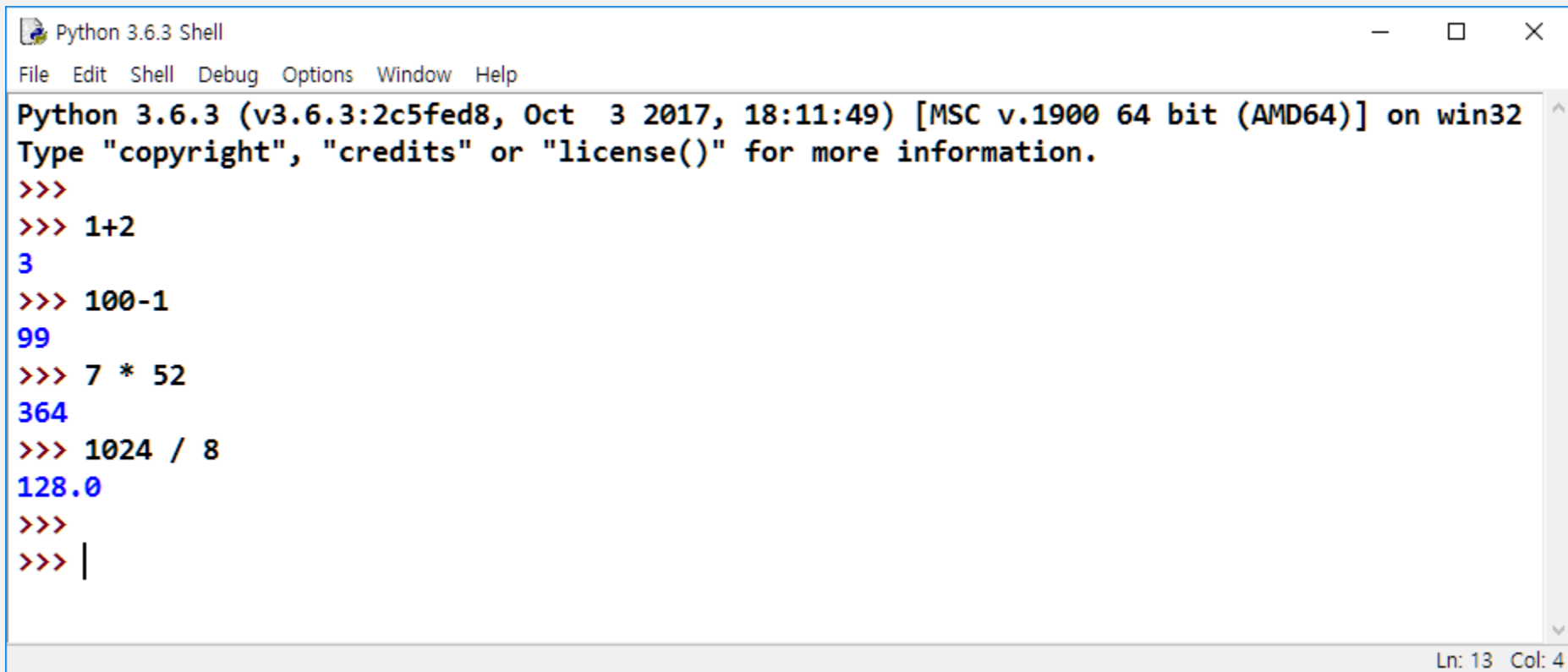


```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

커서가 깜빡임. 뭔가 말을 하라고 함.
여기에 지시를 내리면 됨.

Ln: 3 Col: 4

계산을 시켜보자.

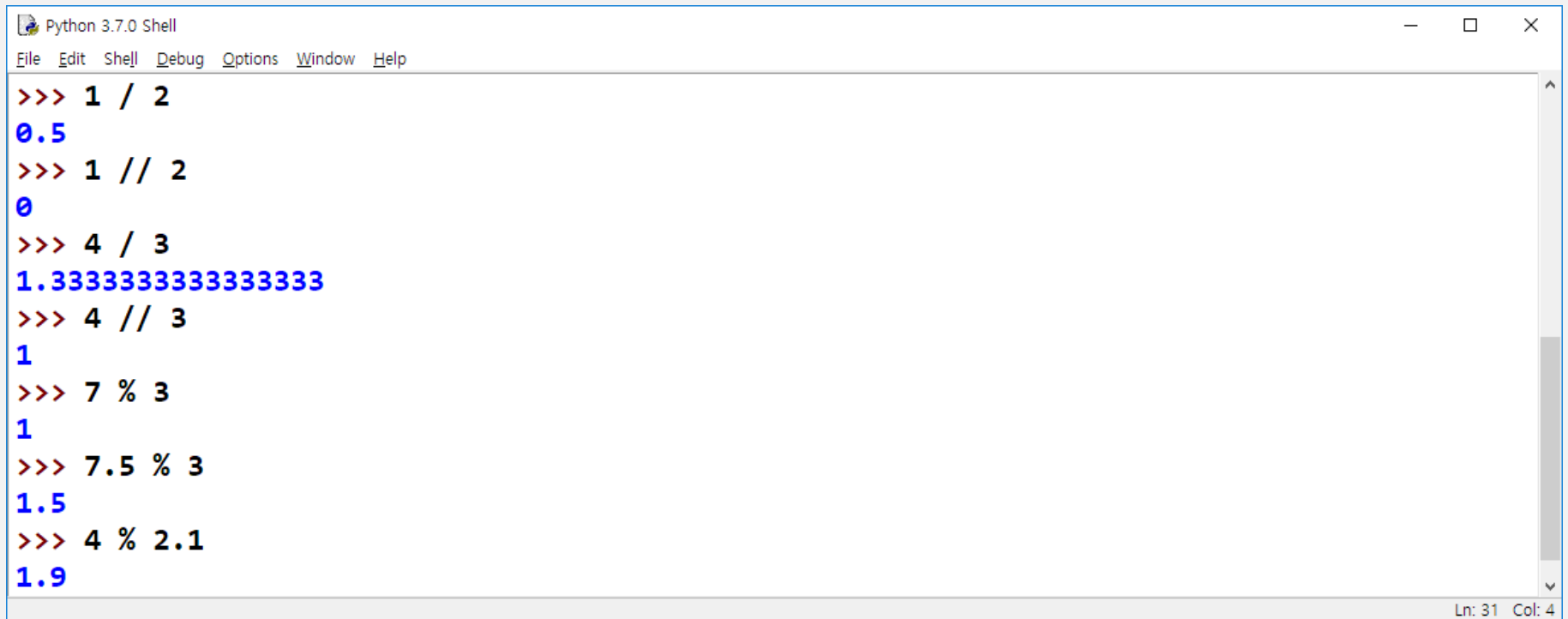


A screenshot of a Python 3.6.3 Shell window. The window has a title bar with the text 'Python 3.6.3 Shell' and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area shows the following content:

```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 18:11:49) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>> 1+2
3
>>> 100-1
99
>>> 7 * 52
364
>>> 1024 / 8
128.0
>>>
>>> |
```

The status bar at the bottom right of the window displays 'Ln: 13 Col: 4'.

나누기와 나머지 연산



A screenshot of a Python 3.7.0 Shell window. The window has a title bar with the text 'Python 3.7.0 Shell' and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains a series of Python commands and their outputs, each on a new line. The commands are: 1 / 2, 1 // 2, 4 / 3, 4 // 3, 7 % 3, 7.5 % 3, and 4 % 2.1. The outputs are: 0.5, 0, 1.3333333333333333, 1, 1, 1.5, and 1.9. The status bar at the bottom right of the window shows 'Ln: 31 Col: 4'.

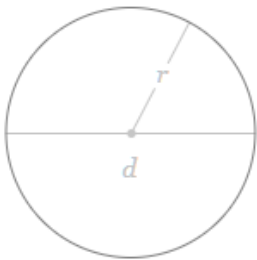
```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>> 1 / 2
0.5
>>> 1 // 2
0
>>> 4 / 3
1.3333333333333333
>>> 4 // 3
1
>>> 7 % 3
1
>>> 7.5 % 3
1.5
>>> 4 % 2.1
1.9
Ln: 31 Col: 4
```

원의 넓이를 구해보자. 반지름이 3미터 이면?

Circle
Solve for area ▾

$$A = \pi r^2$$

r Radius



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>>
>>>
>>>
>>>
>>> 3.141592653589793 * (3 * 3)
28.274333882308138
>>>
>>> 3.141592653589793 * 3 ** 2
28.274333882308138
>>>
>>> |
```

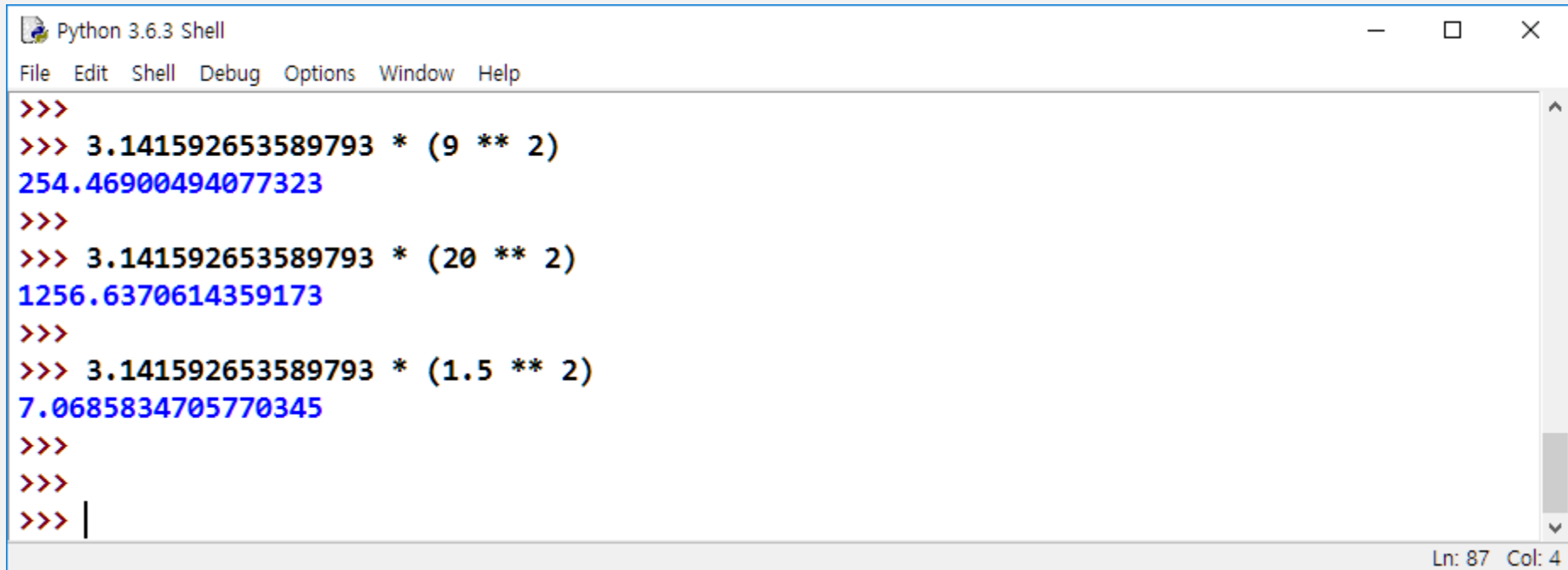
Ln: 77 Col: 4

28.274328 평방미터

IDLE 에디팅 팁

- 라인 단위로의 입력이 기본
- 이미 입력한 라인은 편집이 불가능
- 입력한 라인들은 내부 버퍼에 저장되어 있음.
- Alt + p 와 Alt + n 을 이용해서, 앞서 입력했던 라인들을 꺼내올 수 있음.

반지름이 9미터이면? 20 미터면? 1.5미터면?



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> 3.141592653589793 * (9 ** 2)
254.46900494077323
>>>
>>> 3.141592653589793 * (20 ** 2)
1256.6370614359173
>>>
>>> 3.141592653589793 * (1.5 ** 2)
7.0685834705770345
>>>
>>>
>>> |
```

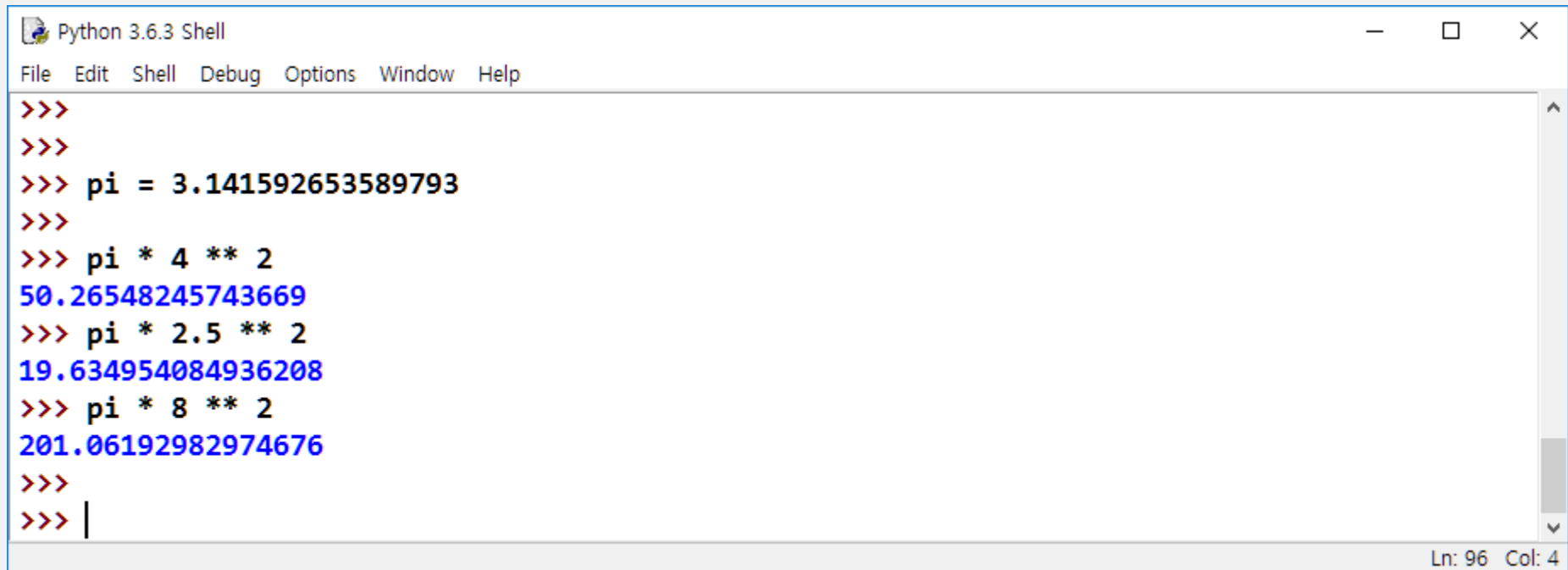
Ln: 87 Col: 4

슬슬 귀찮아지기 시작한다.

3.141592... 를 어디엔가 기록해놓고, 이걸 재사용하면 좋을 것 같은데...

변수(variable)

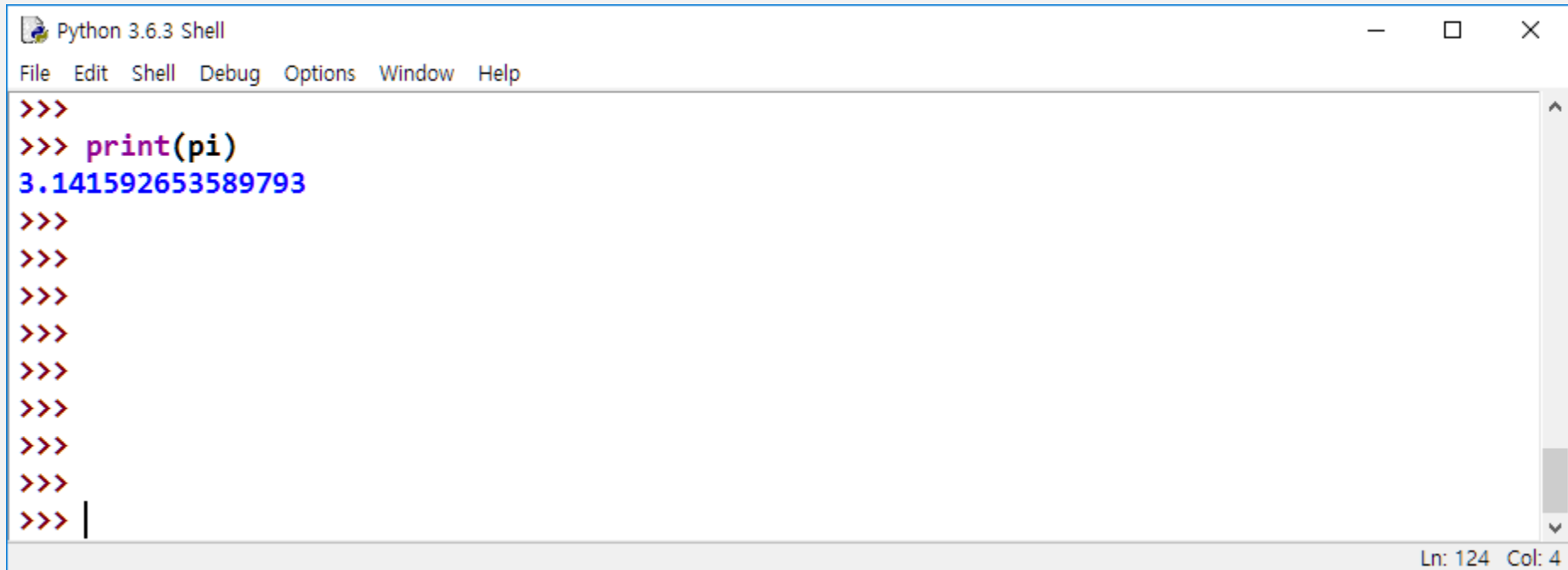
- 변수: 값을 저장해놓는 컴퓨터 메모리 안의 공간
- 변수는 해당되는 이름이 있다. 프로그래머가 이름을 지어야 함.
- 이름은 영문자와 숫자를 조합해서 씀. 단, 파이썬의 기본 단어는 쓰면 안됨.
- 사실, 변수의 값은 맘대로 언제든지 바꿀 수 있다.



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>>
>>> pi = 3.141592653589793
>>>
>>> pi * 4 ** 2
50.26548245743669
>>> pi * 2.5 ** 2
19.634954084936208
>>> pi * 8 ** 2
201.06192982974676
>>>
>>> |
```

Ln: 96 Col: 4

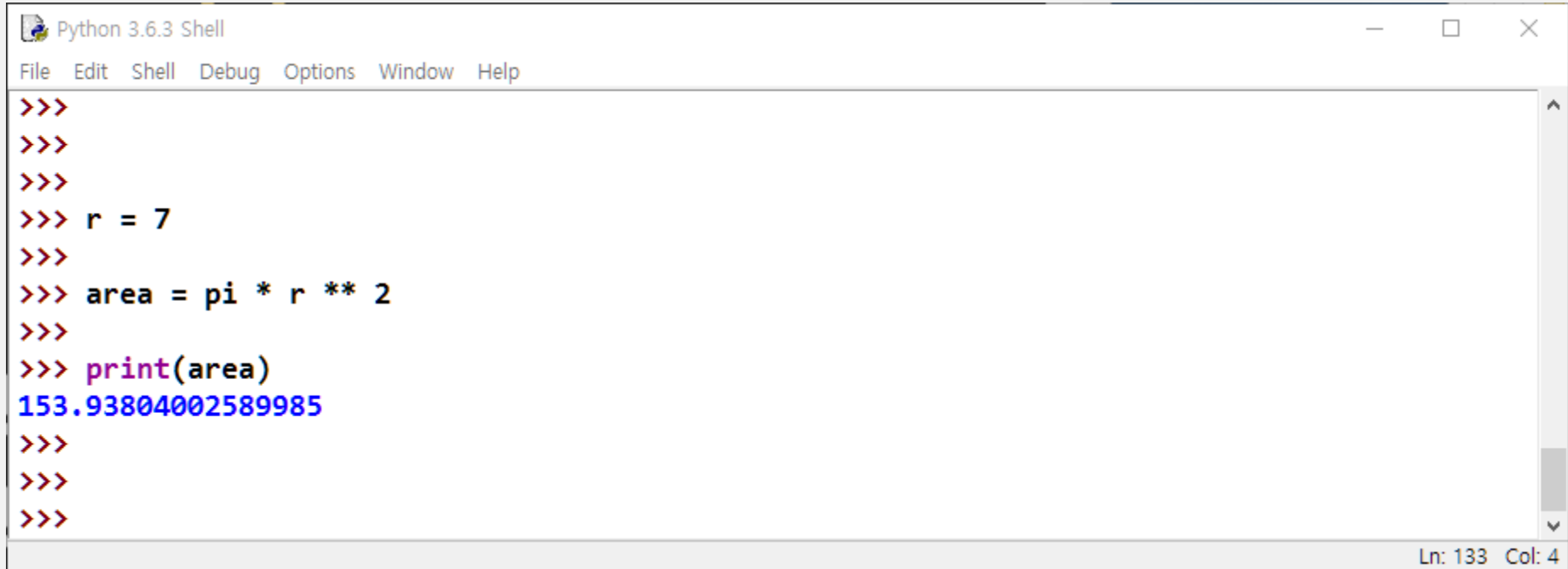
print 함수를 이용하면, 변수의 값을 볼 수 있다.



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>> print(pi)
3.141592653589793
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> |
```

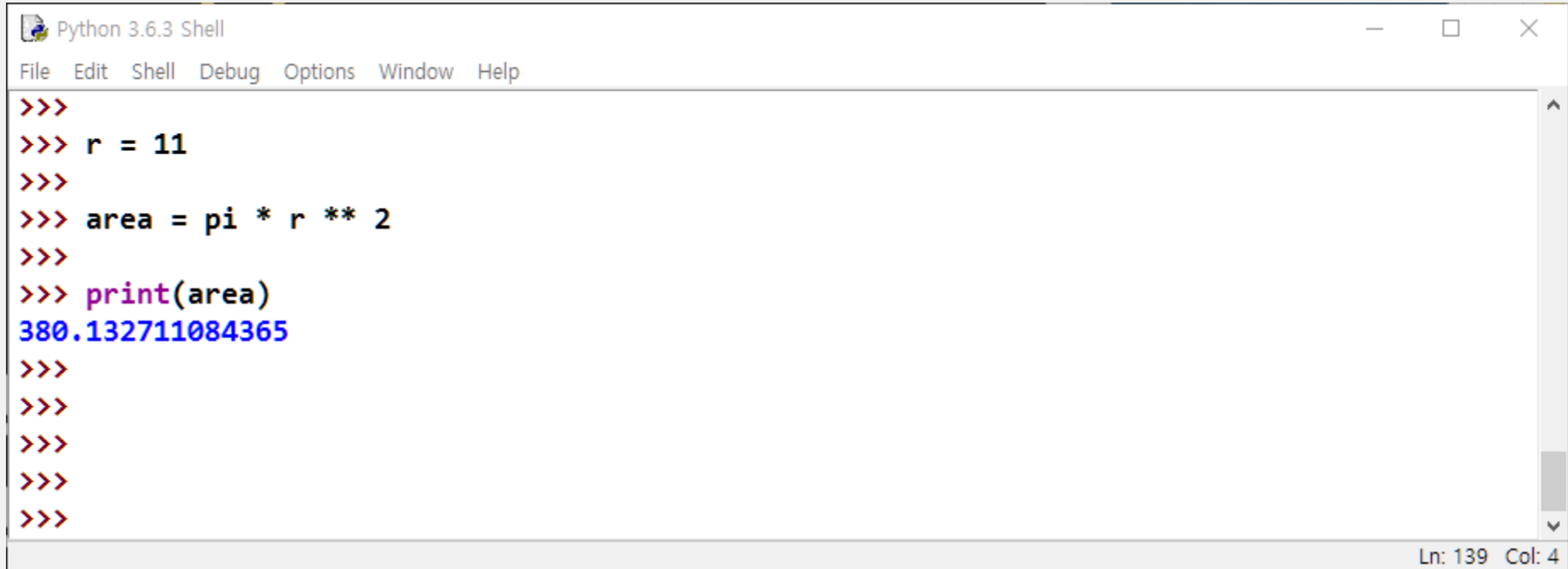
Ln: 124 Col: 4

반지름 변수 r과 면적 변수 area 를 사용한 면적 계산




```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
>>>
>>>
>>>
>>> r = 7
>>>
>>> area = pi * r ** 2
>>>
>>> print(area)
153.93804002589985
>>>
>>>
>>>
Ln: 133 Col: 4
```

변수값을 바꿔서 사용 : r 에 11을 대입

A screenshot of a Python 3.6.3 Shell window. The window has a title bar with the text 'Python 3.6.3 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 area is a text editor showing a Python script. The script consists of several lines of code: four empty lines, followed by 'r = 11', another empty line, 'area = pi * r ** 2', another empty line, 'print(area)', and the output '380.132711084365'. The output is displayed in blue text. The script ends with four more empty lines. The status bar at the bottom right shows 'Ln: 139 Col: 4'.

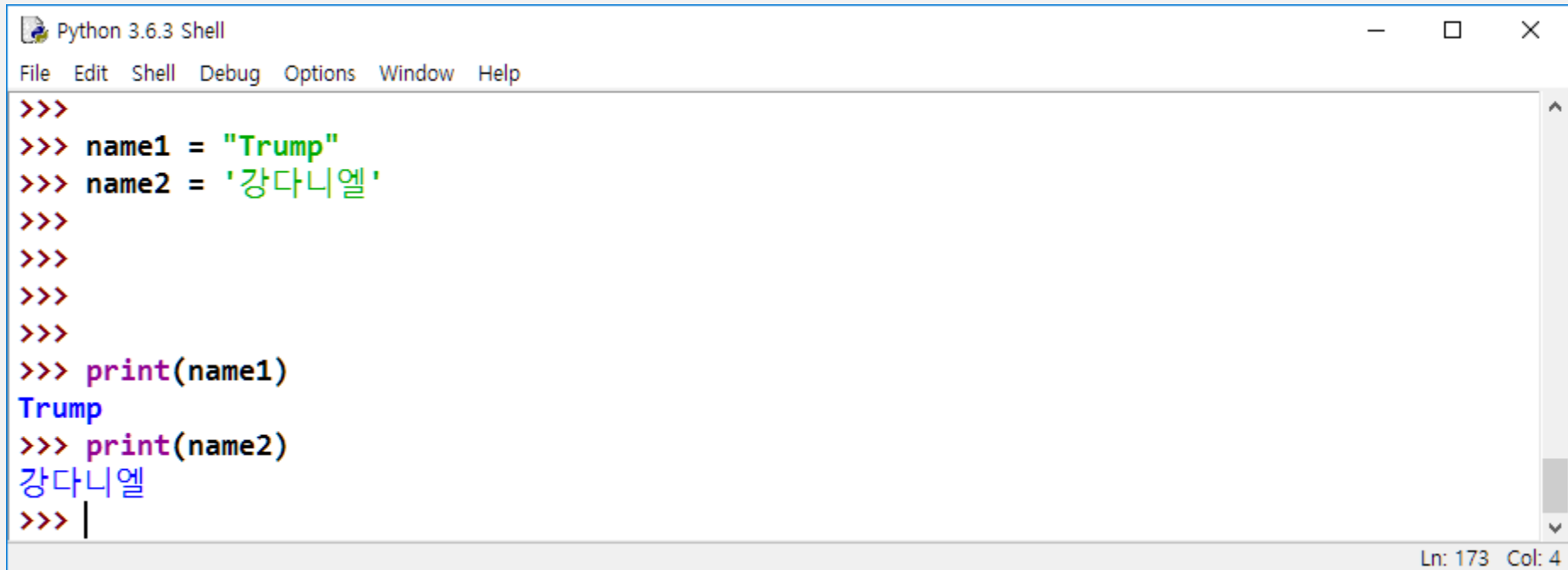
```
>>>
>>> r = 11
>>>
>>> area = pi * r ** 2
>>>
>>> print(area)
380.132711084365
>>>
>>>
>>>
>>>
>>>
```

 + p

연산 기호

연산자	연산
+	덧셈
-	뺄셈
*	곱셈
/ 과 //	나눗셈
**	제곱
%	나머지

변수에는 문자열(string)을 담을 수 있다.

A screenshot of a Python 3.6.3 Shell window. The window has a title bar with the text 'Python 3.6.3 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 area is a text editor with a light blue background. It contains the following code:

```
>>>  
>>> name1 = "Trump"  
>>> name2 = '강다니엘'  
>>>  
>>>  
>>>  
>>> print(name1)  
Trump  
>>> print(name2)  
강다니엘  
>>> |
```

The output of the code is shown in blue text. At the bottom right of the window, the status bar shows 'Ln: 173 Col: 4'.

문자열은 문자들의 집합임. 여러 개의 문자들을 나열한 것. 큰따옴표 또는 작은 따옴표로 감쌌. 한글, 영어 상관없음.

기본 자료형(Type)

- 변수에는 다양한 종류의 정보를 담을 수 있음.
- 어떤 변수의 자료형을 알고 싶으면, type(변수명) 함수를 이용함.

order = 4 정수형 int

pi = 3.141592 실수형 float

name1 = "Trump" 문자열형 str
name2 = 'Daehyun'

result = True 참거짓형 bool



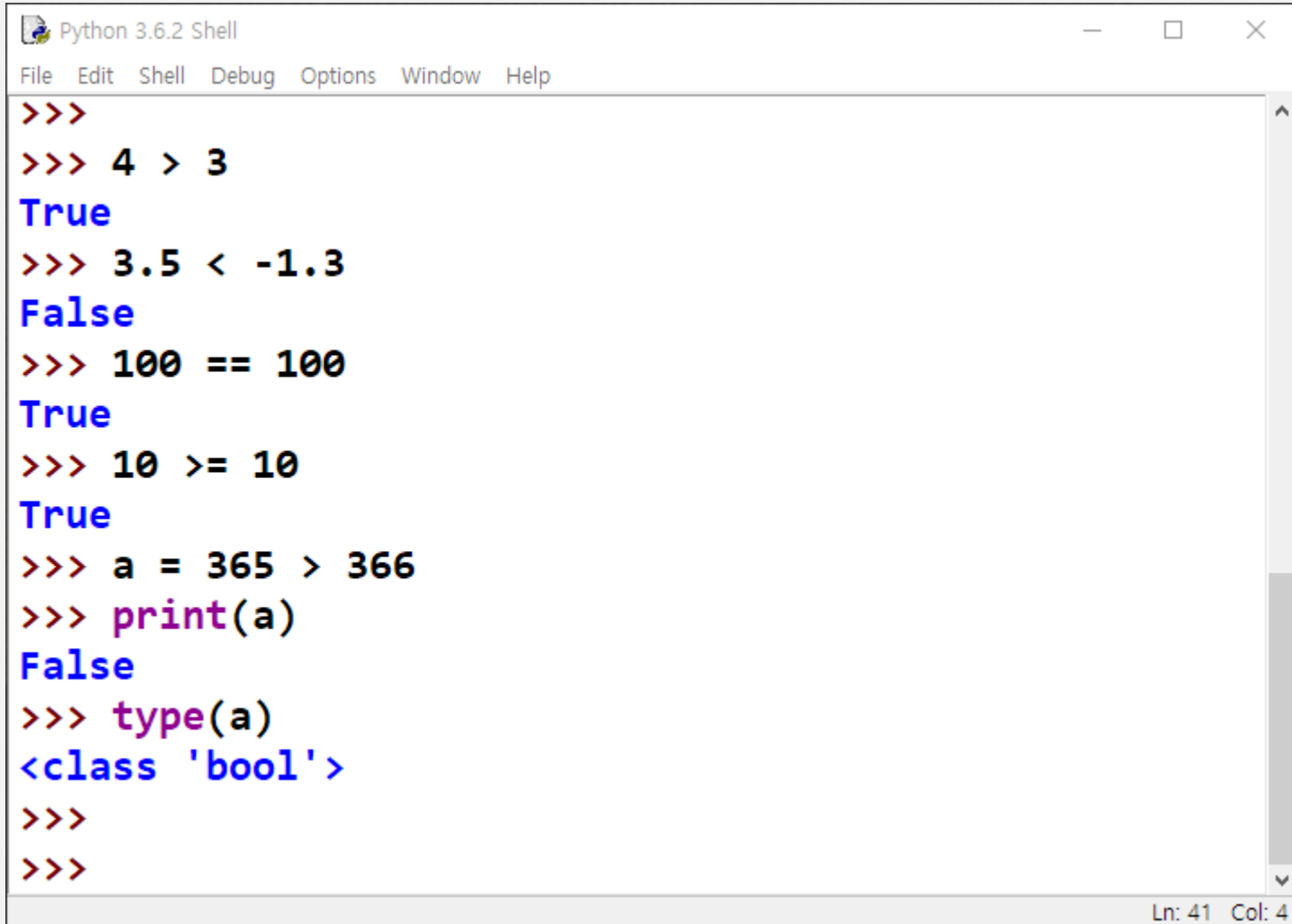
A screenshot of a Python 3.6.3 Shell window. The window has a title bar with the text "Python 3.6.3 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains a series of Python commands and their outputs, color-coded for readability. The commands are: three prompt characters (>>>), an assignment of 4 to the variable 'order', an assignment of 3.141592 to the variable 'pi', an assignment of the string "Trump" to the variable 'name', another set of three prompt characters, and then three calls to the 'type()' function on the variables 'order', 'pi', and 'name' respectively. The outputs are: '<class 'int'>', '<class 'float'>', and '<class 'str'>'. The window also features a vertical scrollbar on the right side and a status bar at the bottom right showing "Ln: 199 Col: 4".

```
>>>
>>> order = 4
>>> pi = 3.141592
>>> name = "Trump"
>>>
>>> type(order)
<class 'int'>
>>> type(pi)
<class 'float'>
>>> type(name)
<class 'str'>
>>>
```

Ln: 199 Col: 4

비교 연산(Comparison Operation)

- 두개의 값의 대소, 동일 등을 확인하는 계산.
- 결과는 참(True) 또는 거짓(False)임.

A screenshot of a Python 3.6.2 Shell window. The window has a title bar with the text 'Python 3.6.2 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 area of the window contains a series of Python commands and their outputs. The commands are: '>>>', '>>> 4 > 3', '>>> 3.5 < -1.3', '>>> 100 == 100', '>>> 10 >= 10', '>>> a = 365 > 366', '>>> print(a)', and '>>> type(a)'. The outputs are: 'True', 'False', 'True', 'True', 'False', and '<class \'bool\'>'. The window also has a status bar at the bottom right showing 'Ln: 41 Col: 4'.

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
>>>
>>> 4 > 3
True
>>> 3.5 < -1.3
False
>>> 100 == 100
True
>>> 10 >= 10
True
>>> a = 365 > 366
>>> print(a)
False
>>> type(a)
<class 'bool'>
>>>
>>>
Ln: 41 Col: 4
```

비교 연산 기호

기호	뜻
<	작다
<=	작거나 같다
==	같다
>=	크거나 같다
>	크다
!=	다르다

Str

```
Python 3.7.0 Shell - C:/Users/dustinlee/Desktop/test.py (3.7.0)
File Edit Shell Debug Options Window Help

>>>
>>> first = "Daehyun"
>>> last = "Lee"
>>> name = first + " " + last
>>> name
'Daehyun Lee'
>>> print(name)
Daehyun Lee
>>>
>>> name * 2
'Daehyun LeeDaehyun Lee'
>>> name * 3
'Daehyun LeeDaehyun LeeDaehyun Lee'
>>>
>>> name[0]
'D'
>>> name[2]
'e'
>>> name[-1]
'e'
>>> name[-2]
'e'
```

Ln: 157 Col: 4

Slice(슬라이스)

■ 문자열의 일부분을 잘라내는 기법

name[start : stop : step]

```
>>> title = "Python 2D Game Programming"
>>> title[0:6]
'Python'
>>> title[7:9]
'2D'
>>> title[10:14]
'Game'
>>> title[:6]
'Python'
>>> title[-11:]
'Programming'
>>> title[::2]
'Pto DGm rgamn'
>>> title[::-1]
'gnimmargorP emaG D2 nohtyP'
```

List

```
>>> twice = ['momo', 'sana', 'zwi', 'nayun', 'dahyun']
>>> black_pink = ['jisu', 'jenni', 'rose', 'risa']
>>> twice
['momo', 'sana', 'zwi', 'nayun', 'dahyun']
>>> twice.append('jihyo')
>>> twice
['momo', 'sana', 'zwi', 'nayun', 'dahyun', 'jihyo']
>>> twice.sort()
>>> twice
['dahyun', 'jihyo', 'momo', 'nayun', 'sana', 'zwi']
>>> len(twice)
6
>>> unite = twice + black_pink
>>> unite
['dahyun', 'jihyo', 'momo', 'nayun', 'sana', 'zwi', 'jisu', 'jenni', 'rose', 'risa']
>>> unite.remove('momo')
>>> unite
['dahyun', 'jihyo', 'nayun', 'sana', 'zwi', 'jisu', 'jenni', 'rose', 'risa']
```

List 에서 Slice 가 적용됨.

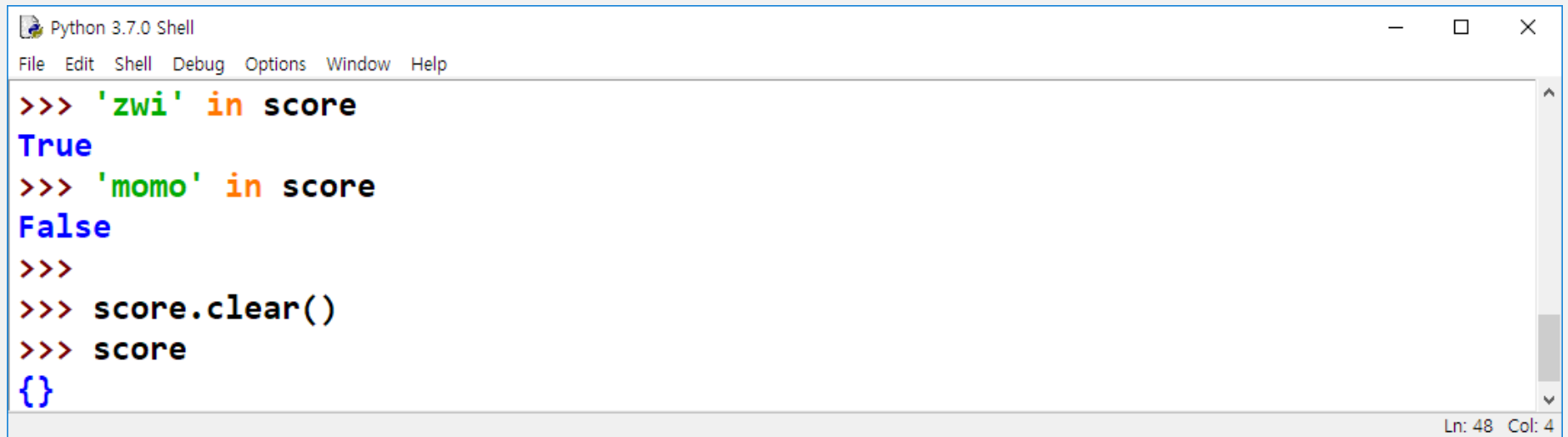
```
>>> unite[0]
'dahyun'
>>> unite[-1]
'risa'
>>> unite[:3]
['dahyun', 'jihyo', 'nayun']
>>> unite[-3:]
['jeni', 'rose', 'risa']
```


Dictionary

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

>>> score = { 'momo' : 80, 'zwi' : 85, 'sana' : 98 }
>>> type(score)
<class 'dict'>
>>> score['momo']
80
>>> score['nayun']
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    score['nayun']
KeyError: 'nayun'
>>> score['nayun'] = 100
>>> score
{'momo': 80, 'zwi': 85, 'sana': 98, 'nayun': 100}
>>> del score['momo']
>>> score
{'zwi': 85, 'sana': 98, 'nayun': 100}
>>> score.keys()
dict_keys(['zwi', 'sana', 'nayun'])
>>> score.values()
dict_values([85, 98, 100])
>>>
```

Ln: 12 Col: 0



A screenshot of a Python 3.7.0 Shell window. The window has a title bar with the text "Python 3.7.0 Shell" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window contains a Python REPL session with the following code and output:

```
>>> 'zwi' in score
True
>>> 'momo' in score
False
>>>
>>> score.clear()
>>> score
{}
Ln: 48 Col: 4
```

The code is color-coded: strings are green, keywords like 'in' are orange, and function calls like 'clear()' are black. The output 'True' and 'False' are blue. The prompt '>>>' is red. The status bar at the bottom right shows 'Ln: 48 Col: 4'.

Tuple

- 여러 개의 값을 동시에 관리. 리스트와 유사.
- 하지만, 기본적으로 값을 바꿀 수는 없음. ==> 프로그램 중 변경이 되지 않는 값들의 모음이 필요할 때 사용하면 됨.

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

>>>
>>> t1 = (1,2,3)
>>> t2 = (1, )
>>> t3 = ()
>>> t4 = 1,2,3,4
>>> t4
(1, 2, 3, 4)
>>> type(t4)
<class 'tuple'>
>>> t5 = (1, 'a', "park", (1, 2))
>>> t1[1:]
(2, 3)
>>> t1 + t5
(1, 2, 3, 1, 'a', 'park', (1, 2))
>>> t4 * t4
Traceback (most recent call last):
  File "<pyshell#15>", line 1, in <module>
    t4 * t4
TypeError: can't multiply sequence by non-int of type 'tuple'
>>> t4 * 2
(1, 2, 3, 4, 1, 2, 3, 4)
>>> |
```

Ln: 37 Col: 4

set

- 집합 자료형
- 리스트와 달리, 중복을 허용하지 않고, 순서가 없음.

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>> s1 = {1,2,3}
>>> type(s1)
<class 'set'>
>>> s1 = {1,2,2,4}
>>> s1
{1, 2, 4}
>>> l1 = [1,2,2,2,2,3,3,3,3,5,5,5,5,5]
>>> s1 = set(l1)
>>> s1
{1, 2, 3, 5}
>>> s2 = {3,5,6,7}
>>> s1 + s2
Traceback (most recent call last):
  File "<pyshell#36>", line 1, in <module>
    s1 + s2
TypeError: unsupported operand type(s) for +: 'set' and 'set'
>>> s1 | s2
{1, 2, 3, 5, 6, 7}
>>> s1 & s2
{3, 5}
>>> s2 - s1
{6, 7}
>>> s1 - s2
{1, 2}
>>> s1.add(8)
>>> s1
{1, 2, 3, 5, 8}
>>> s2.remove(6)
>>> s2
{3, 5, 7}
```

Ln: 86 Col: 4

Complex Data Type

■ List – list

- 순서가 있는, 중복을 허용하는 데이터들의 집합.
- 원하는 데이터를 찾기 위해, 순서 index 를 이용.

[val1, val2, ...]

■ Dictionary – dict

- 검색을 위한 키를 갖는 데이터들의 집합
- key – value 쌍 들의 집합

{ key1: val1, key2: val2, ... }

■ Tuple – tuple

- 순서가 있는, 중복을 허용하는 데이터들의 집합
- 다만, 데이터값을 변경하는 것은 불가

(val1, val2, ...)

■ Set – set

- 중복을 허용하지 않는, 순서에 상관없는 데이터들의 집합

{ val1, val2, ... }