

# Lecture #1. 파이썬 기초 (1)

2D 게임 프로그래밍

이대현 교수



한국공학대학교  
TECH UNIVERSITY OF KOREA

# 파이썬

---

- 1991년, Guido van Rossum 이 개발
- VM 기반 인터프리터 언어



# 특징

---

- **다양한 프로그래밍 패러다임 제공**

- 구조적 프로그래밍
- 절차지향 프로그래밍
- 객체지향 프로그래밍
- 함수형 프로그래밍

- **동적 자료형**

- **풍부한 기본 라이브러리 함수**

# 장점

---

- 쉽다.
- 간결하다.
- 빠르게 개발할 수 있다.

# 파이썬의 인기

- 지난 30년간 지속적으로 순위가 상승하면서, 드디어 2022년 1위로 등극 !

Programming Language	2022	2017	2012	2007	2002	1997	1992	1987
Python	1	5	8	7	12	28	-	-
C	2	2	1	2	2	1	1	1
Java	3	1	2	1	1	16	-	-
C++	4	3	3	3	3	2	2	6
C#	5	4	4	8	14	-	-	-
Visual Basic	6	14	-	-	-	-	-	-
JavaScript	7	8	10	9	8	24	-	-
Assembly language	8	10	-	-	-	-	-	-
SQL	9	-	-	-	7	-	-	-
PHP	10	7	6	5	6	-	-	-
Prolog	24	32	33	27	17	21	12	3
Lisp	33	31	13	16	13	10	4	2
Pascal	270	114	16	22	99	9	3	5
(Visual) Basic	-	-	7	4	4	3	6	4

프로그래밍 언어 순위 변화 ( Source : <https://www.tiobe.com/tiobe-index/> )

# 단점

---

- 느리다.
- 너무 유연하다.
- 진입 장벽이 낮다.

## Python Key Words

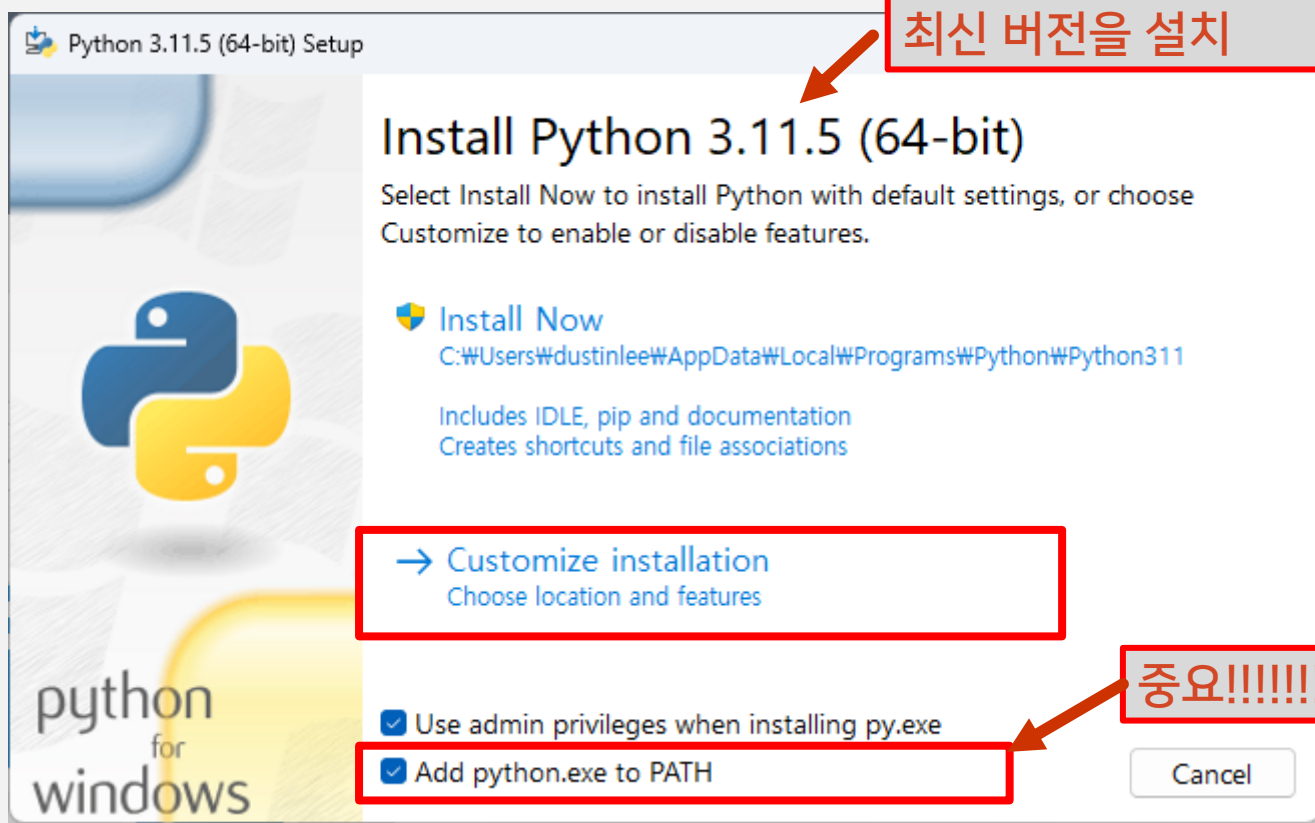
<b>False</b>	<b>class</b>	<b>return</b>	<b>is</b>	<b>finally</b>
<b>None</b>	<b>if</b>	<b>for</b>	<b>lambda</b>	<b>continue</b>
<b>True</b>	<b>def</b>	<b>from</b>	<b>while</b>	<b>nonlocal</b>
<b>and</b>	<b>del</b>	<b>global</b>	<b>not</b>	<b>with</b>
<b>as</b>	<b>elif</b>	<b>try</b>	<b>or</b>	<b>yield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

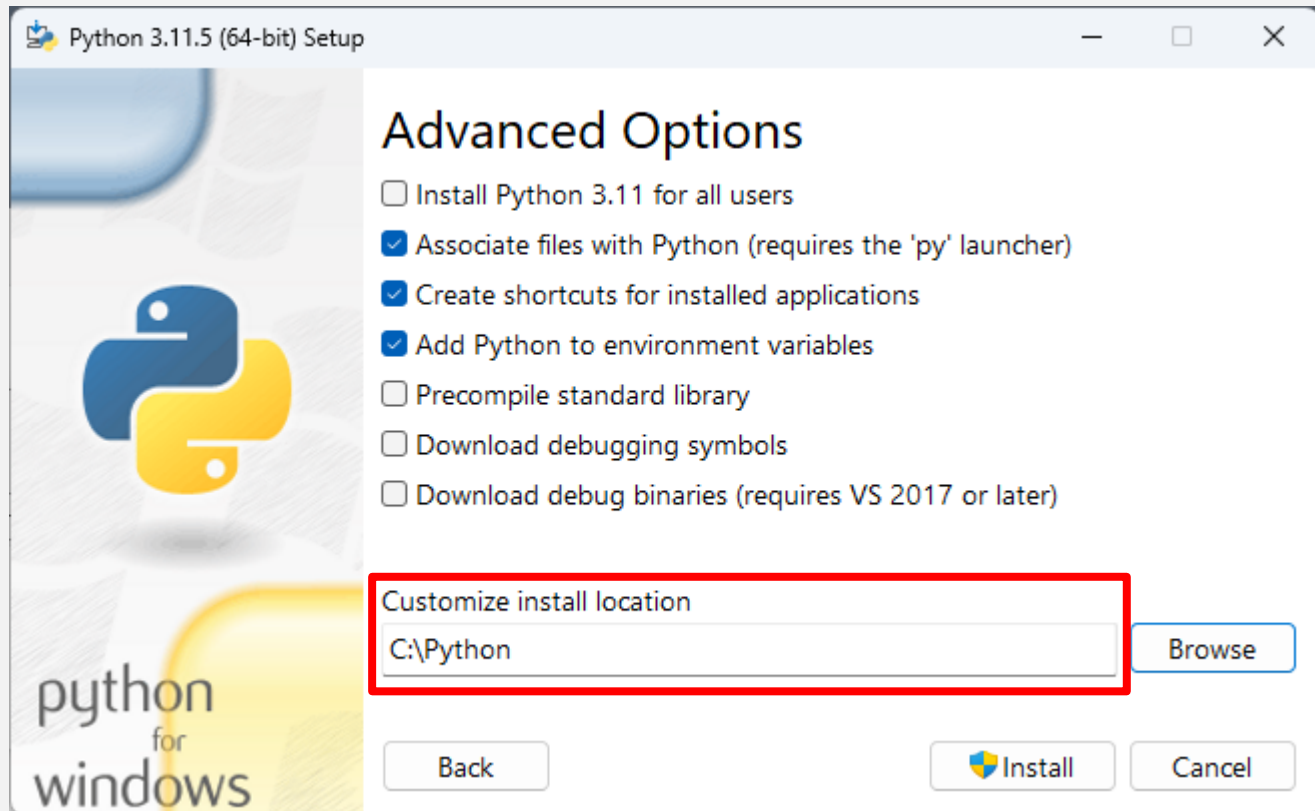
# 파이썬 설치 – <https://www.python.org>

---

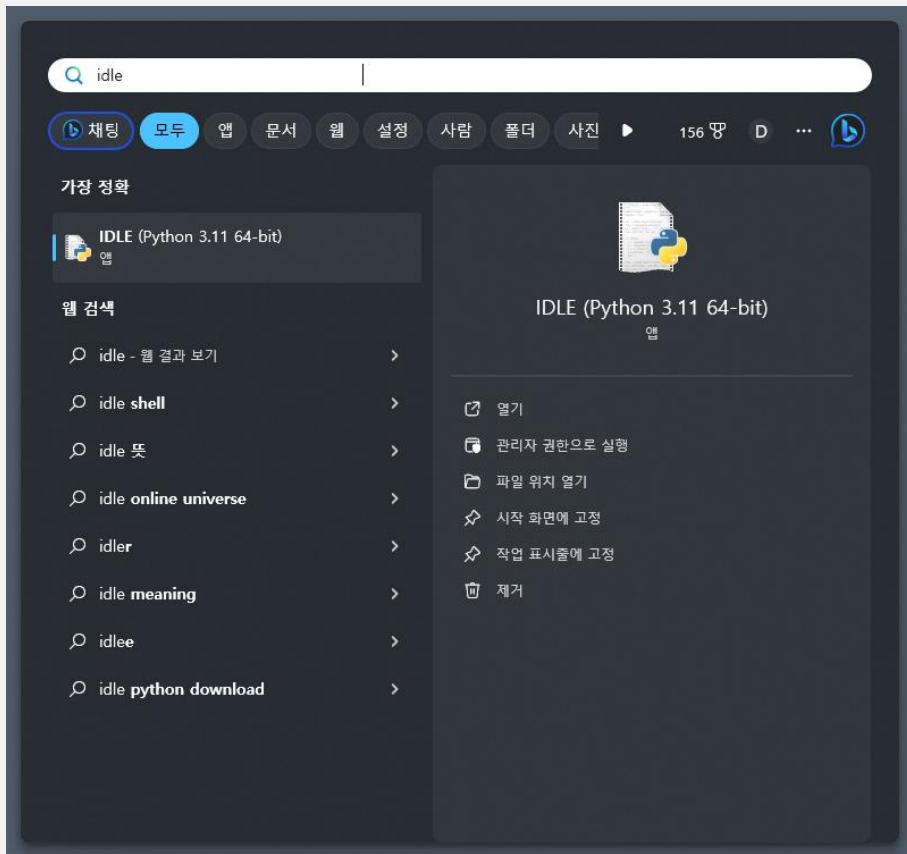
- 64 bit version 설치
- 설치 경로 변경
  - C:\Python 또는 D:\Python
- Python을 모든 경로에서 실행가능하도록 설정
  - “add python 3.xx to path” 옵션 ON





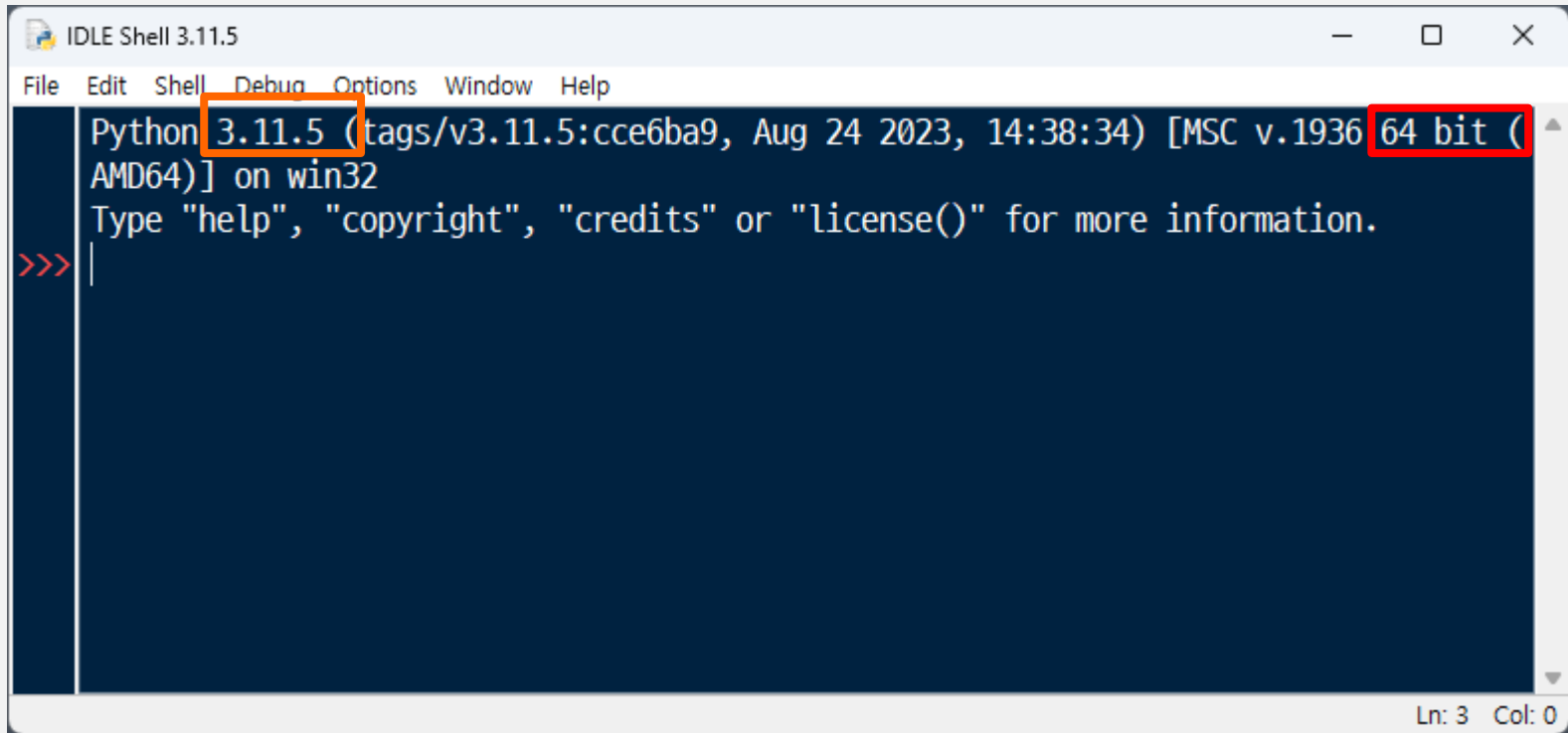


# IDLE의 실행



# 설치 확인 포인트

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



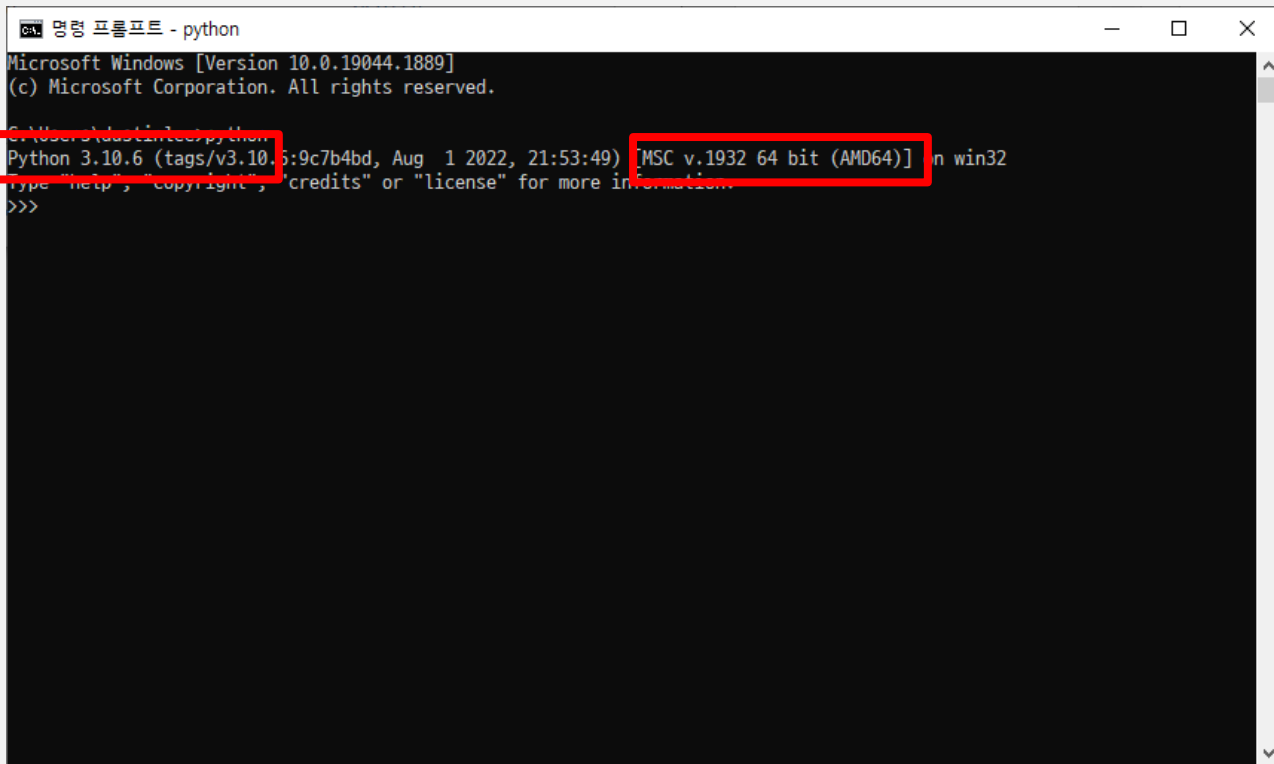
The screenshot shows the IDLE Shell 3.11.5 window. The title bar reads "IDLE Shell 3.11.5". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following information: "Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32". The version number "3.11.5" is highlighted with an orange box, and "64 bit (AMD64)" is highlighted with a red box. Below this, it says "Type 'help', 'copyright', 'credits' or 'license()' for more information." and a red prompt ">>>" is visible. The status bar at the bottom right shows "Ln: 3 Col: 0".

```
IDLE Shell 3.11.5
File Edit Shell Debug Options Window Help
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

Ln: 3 Col: 0

## 설치 확인 포인트 (2)

- cmd 창을 열어서, python 이라고 치면, 앞서 IDLE 을 실행했을 때와 동일하게 파이썬 정보가 표시되어야 함. 중요! 설치한 최신 버전과 동일한 버전으로 표시되어야 함.

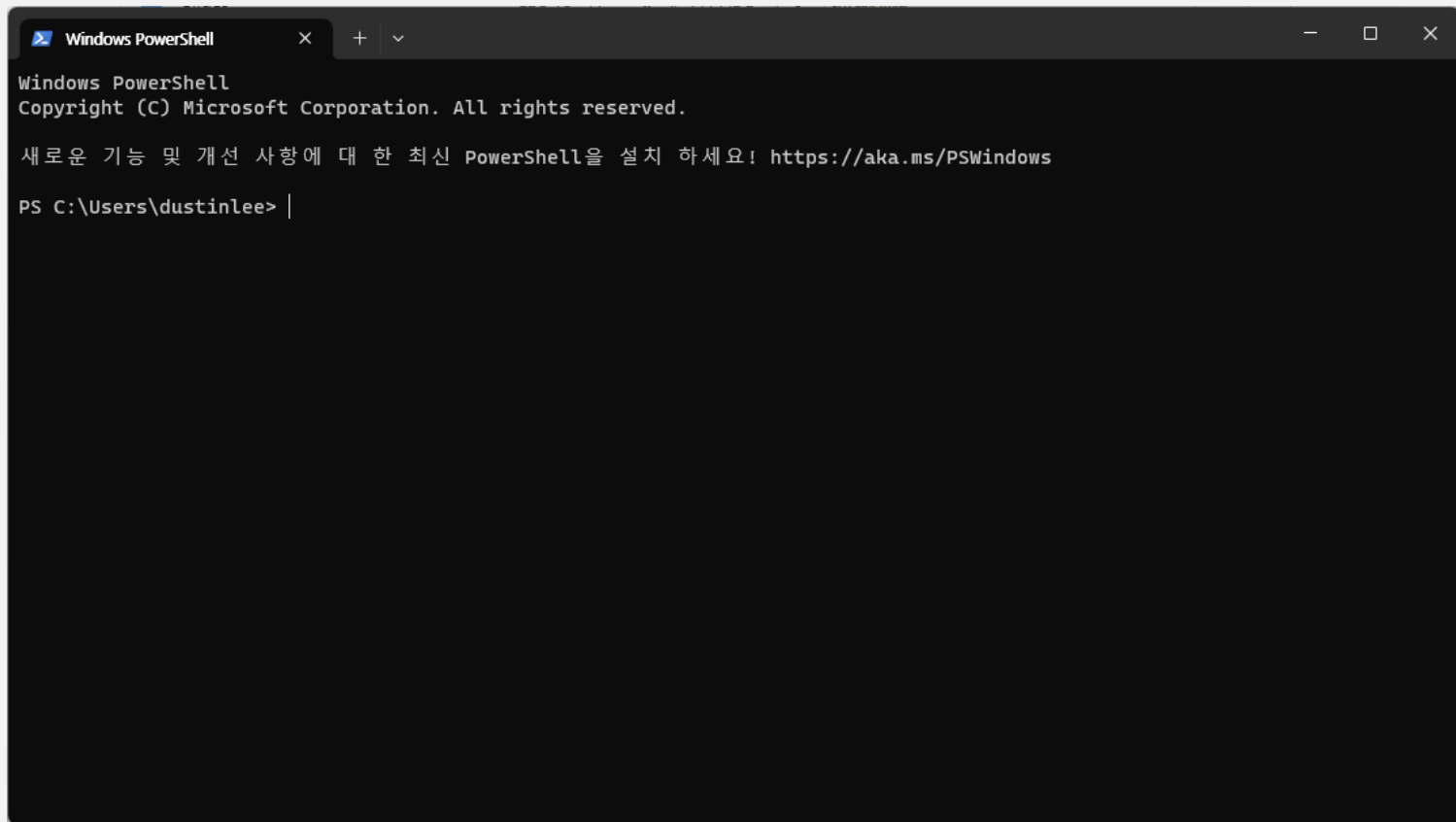


```
명령 프롬프트 - python
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

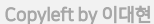
C:\Users\dustinlee>python
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) MSC v.1932 64 bit (AMD64) on win32
Type "help", "copyright", "credits" or "license()" for more information
Python 3.10.6 >>>
```

# powershell

---

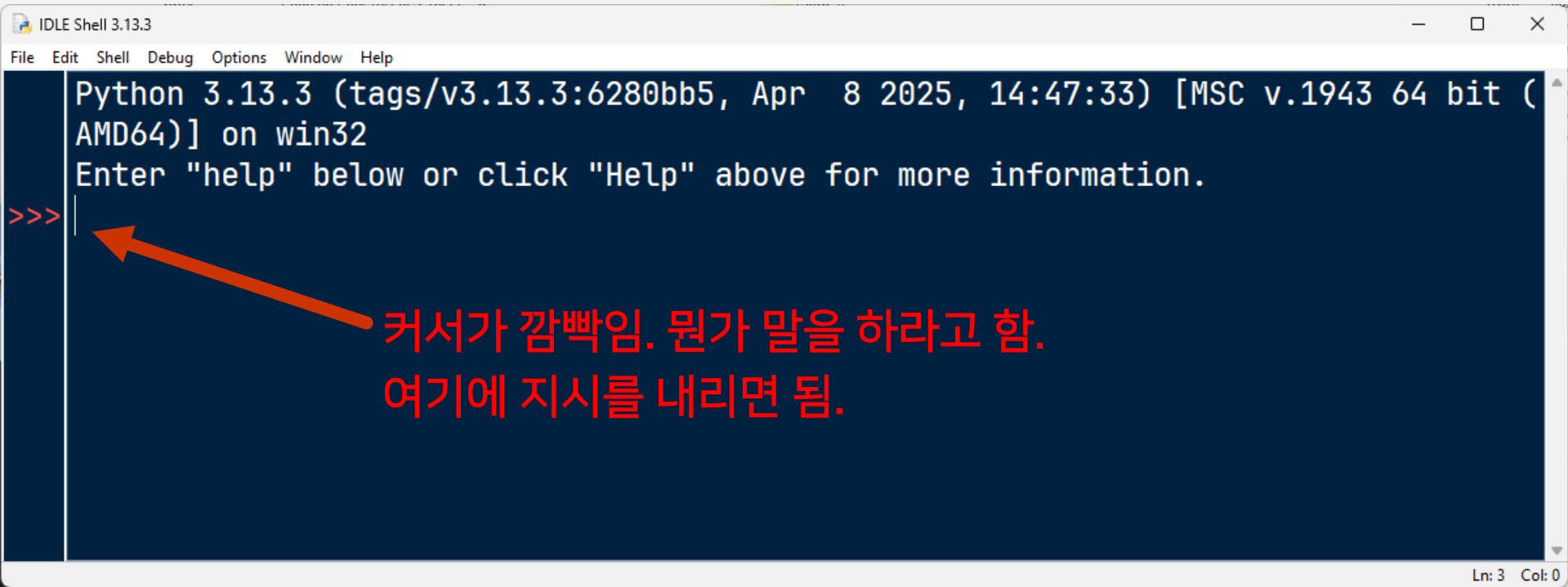


## 2D 게임 프로그래밍



# IDLE 실행 화면

- Python 언어로 지시하면, 이를 해석해서 일을 함.



The screenshot shows the IDLE Shell 3.13.3 window. The title bar reads "IDLE Shell 3.13.3". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area has a dark blue background and displays the following text:

```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr 8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
```

Below the text, the Python prompt `>>>` is visible, followed by a vertical cursor line. A red arrow points from the Korean text annotation to this cursor.

커서가 깜빡임. 뭔가 말을 하라고 함.  
여기에 지시를 내리면 됨.

Ln: 3 Col: 0



# Hello, Python.

---

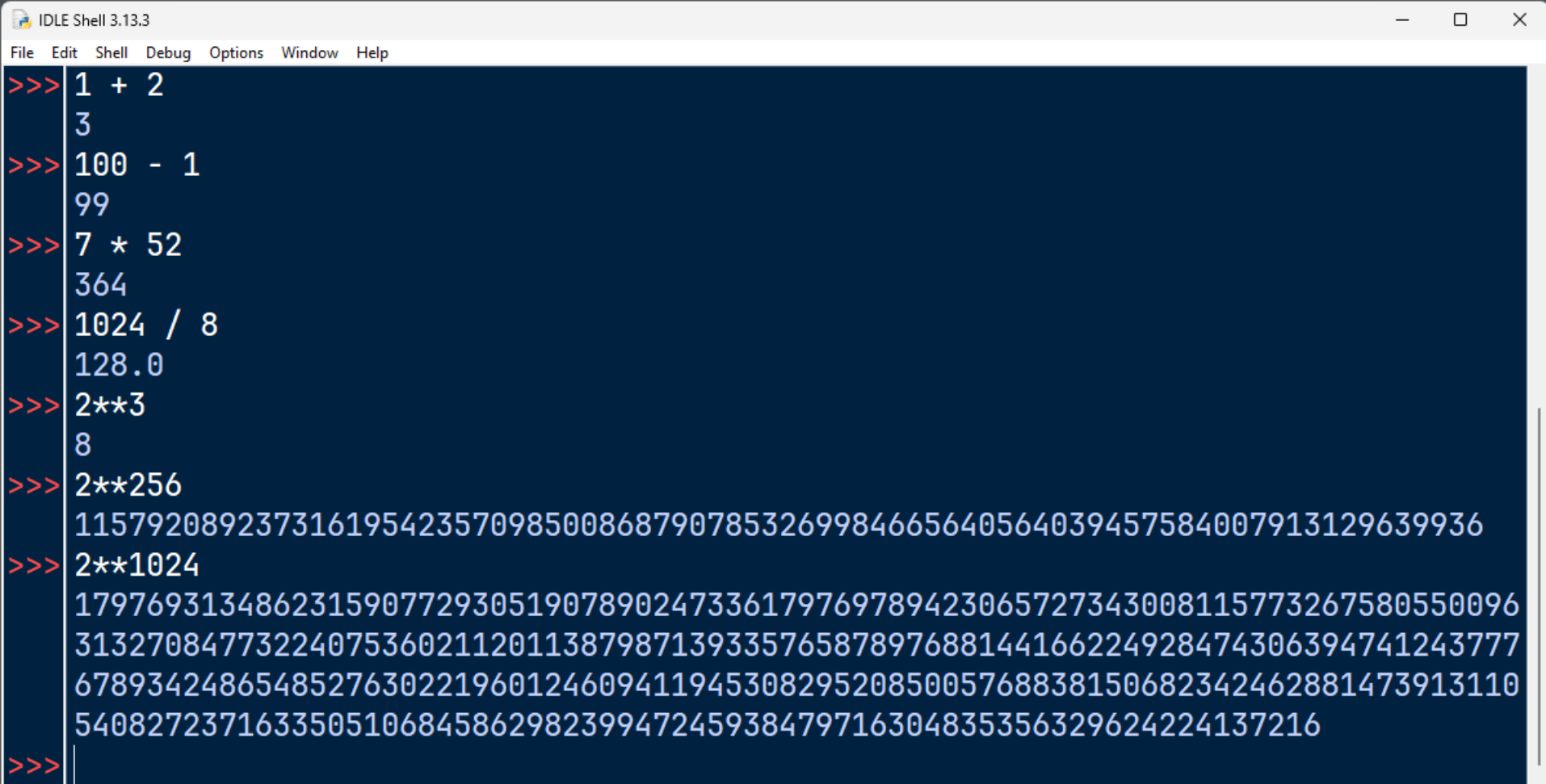


The image shows a screenshot of the IDLE Shell 3.13.3 window. The window has a title bar with the text "IDLE Shell 3.13.3" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window is a dark blue shell with a light blue border. It contains the following text:

```
>>> print("Hello, Python.")  
Hello, Python.  
>>>
```

The text is color-coded: the prompt characters ">>>" are red, the function name "print" is green, the string "Hello, Python." is in quotes and green, and the output "Hello, Python." is white. The cursor is positioned at the end of the second prompt line. In the bottom right corner of the window, the text "Ln: 49 Col: 0" is displayed.

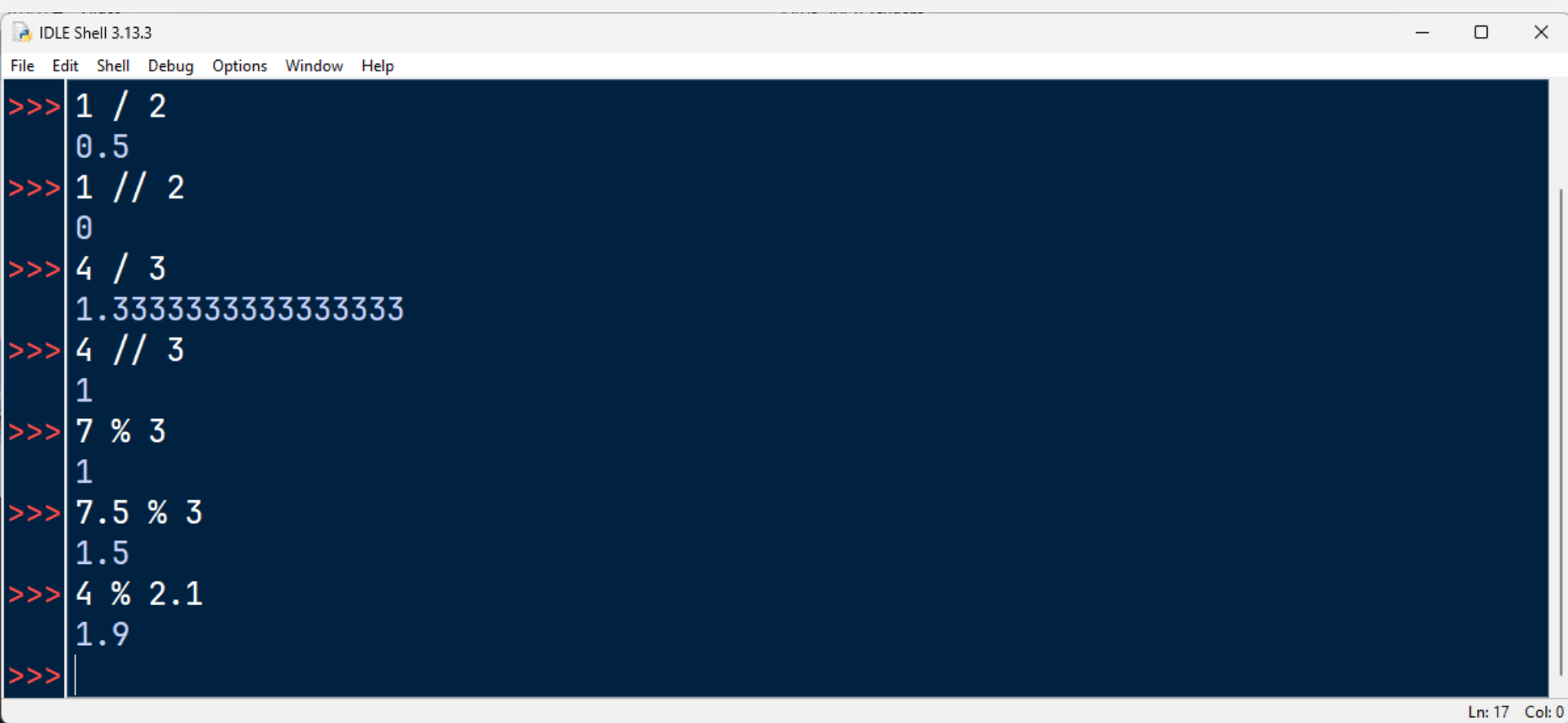
# 계산을 시켜보자.



The image shows a screenshot of an IDLE Shell window titled "IDLE Shell 3.13.3". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area is a dark blue terminal with red prompt characters ">>>". It displays several arithmetic operations and their results, including addition, subtraction, multiplication, and division. The results are shown on the line immediately following the prompt. The last line shows a long multiplication result for 2\*\*1024, which is truncated on the right side of the image.

```
>>> 1 + 2
3
>>> 100 - 1
99
>>> 7 * 52
364
>>> 1024 / 8
128.0
>>> 2**3
8
>>> 2**256
115792089237316195423570985008687907853269984665640564039457584007913129639936
>>> 2**1024
17976931348623159077293051907890247336179769789423065727343008115773267580550096
31327084773224075360211201138798713933576587897688144166224928474306394741243777
67893424865485276302219601246094119453082952085005768838150682342462881473913110
540827237163350510684586298239947245938479716304835356329624224137216
>>>
```

# 나누기와 나머지 연산



A screenshot of the IDLE Shell 3.13.3 window. The window has a title bar with the text 'IDLE Shell 3.13.3' 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 is a dark blue shell with a light blue vertical line indicating the cursor position. The shell contains several lines of text, each starting with a red prompt '>>>'. The text shows the results of various division and modulo operations. The operations and their results are as follows:

```
>>> 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
>>>
```

At the bottom right of the window, the status bar shows 'Ln: 17 Col: 0'.

# 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.

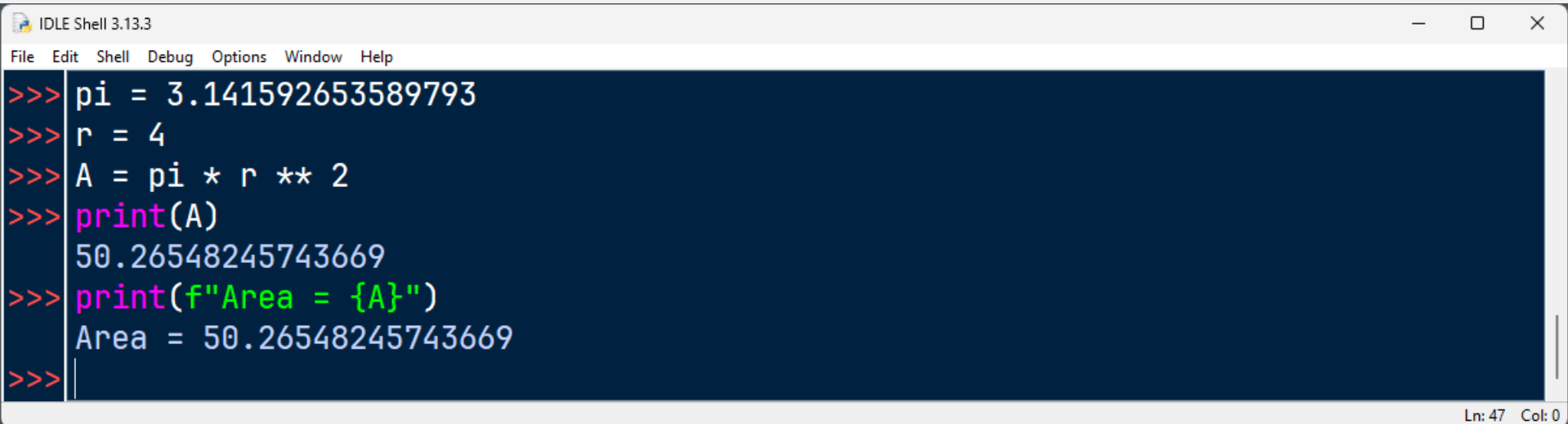
# 연산 기호

---

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

# 변수(variable)

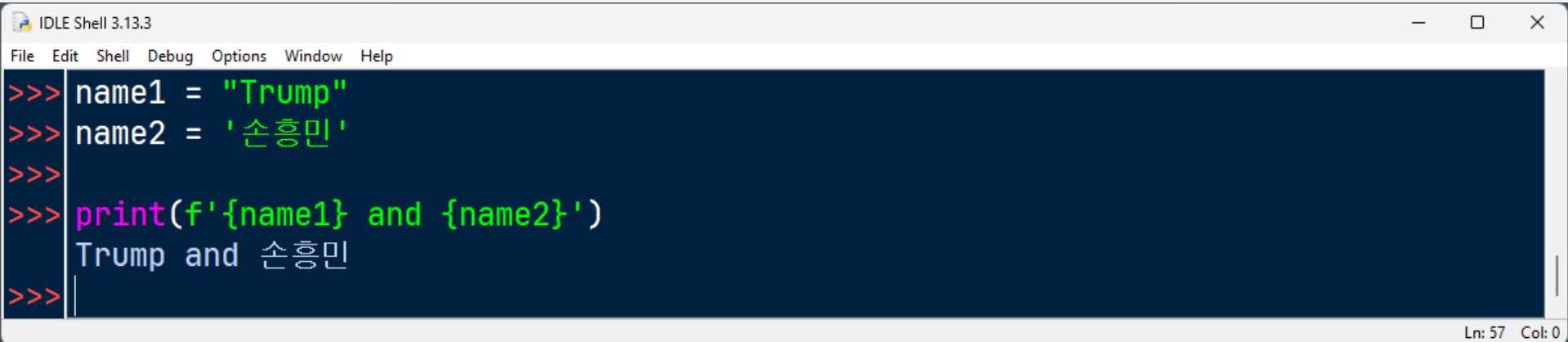
- 변수: 값을 저장해놓는 컴퓨터 메모리 안의 공간
- 변수는 해당되는 이름이 있다. 프로그래머가 이름을 지어야 함.
- 이름은 영문자와 숫자를 조합해서 씀. 단, 파이썬의 기본 단어는 쓰면 안됨.



```
IDLE Shell 3.13.3
File Edit Shell Debug Options Window Help
>>> pi = 3.141592653589793
>>> r = 4
>>> A = pi * r ** 2
>>> print(A)
50.26548245743669
>>> print(f"Area = {A}")
Area = 50.26548245743669
>>>
```

Ln: 47 Col: 0

# 문자열(String)



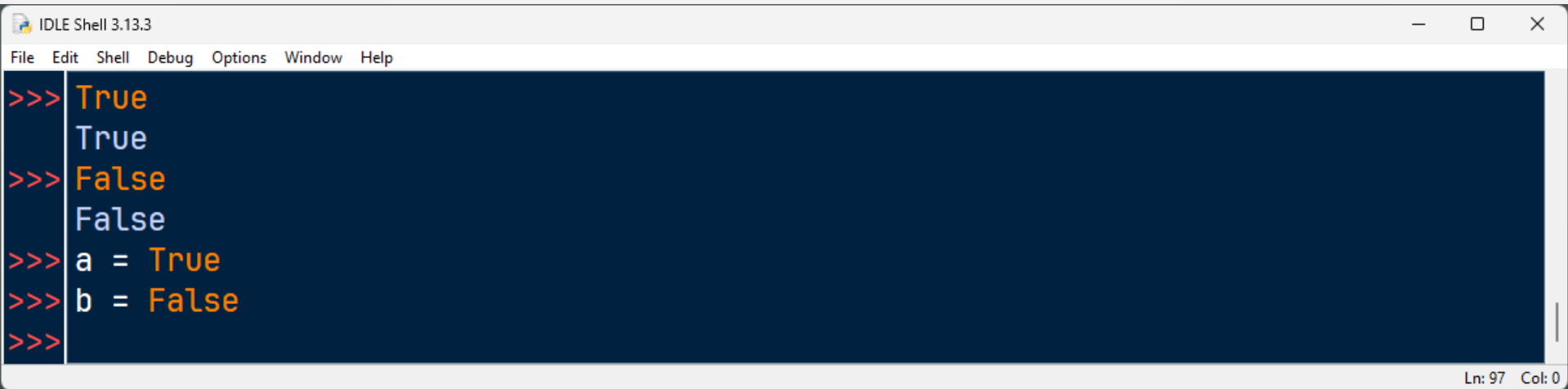
```
IDLE Shell 3.13.3
File Edit Shell Debug Options Window Help
>>> name1 = "Trump"
>>> name2 = '손흥민'
>>>
>>> print(f'{name1} and {name2}')
Trump and 손흥민
>>>
```

Ln: 57 Col: 0

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

# bool

- 참(True), 또는 거짓(False)을 나타내는데 사용되는 자료형

A screenshot of the IDLE Shell 3.13.3 window. The window has a title bar with the text 'IDLE Shell 3.13.3' 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 is a dark blue shell with a light blue border on the left. It contains the following text:

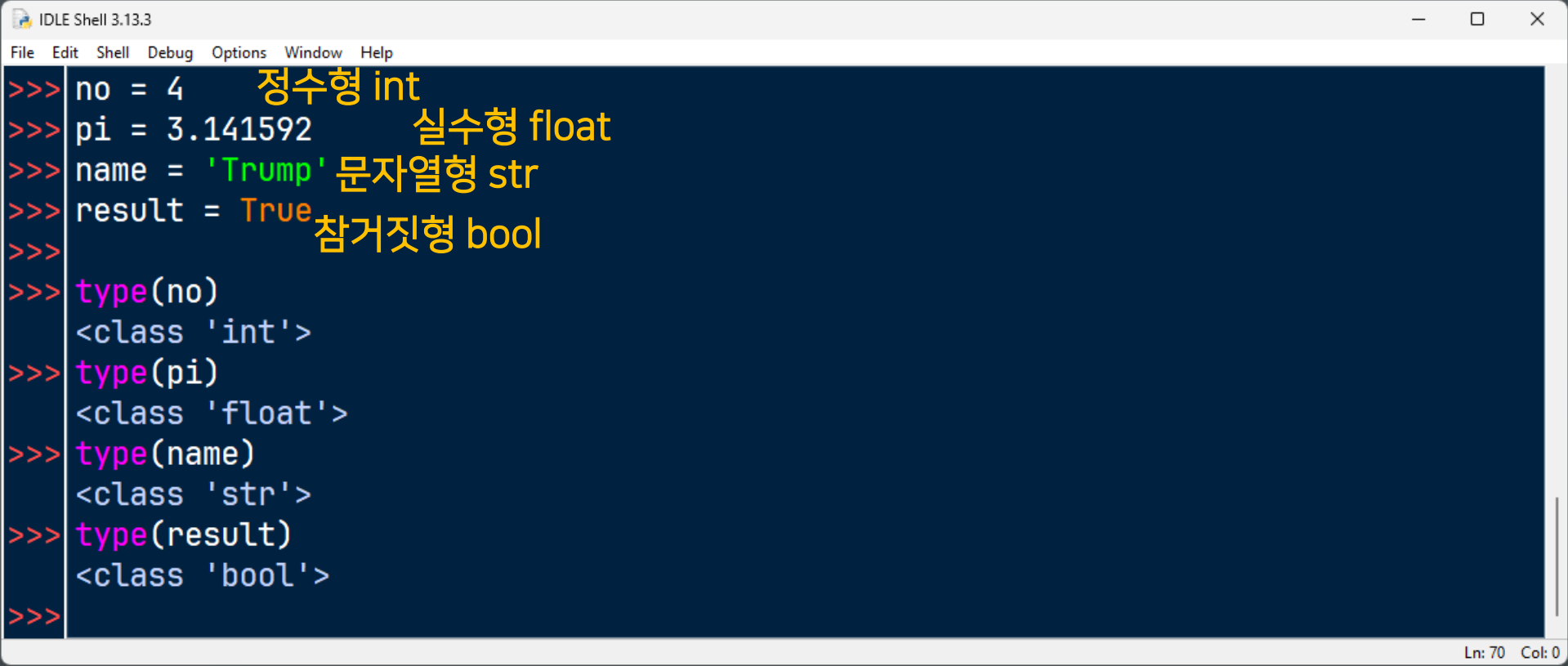
```
>>> True
True
>>> False
False
>>> a = True
>>> b = False
>>>
```

At the bottom right of the shell, the text 'Ln: 97 Col: 0' is displayed.



# 기본 자료형(Type)

- 변수에는 다양한 종류의 정보를 담을 수 있음.



The screenshot shows the IDLE Shell 3.13.3 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area has a dark blue background with yellow and green text. It shows the following code and its output:

```
>>> no = 4      정수형 int
>>> pi = 3.141592 실수형 float
>>> name = 'Trump' 문자열형 str
>>> result = True 참거짓형 bool
>>> type(no)
<class 'int'>
>>> type(pi)
<class 'float'>
>>> type(name)
<class 'str'>
>>> type(result)
<class 'bool'>
>>>
```

At the bottom right of the window, it says "Ln: 70 Col: 0".

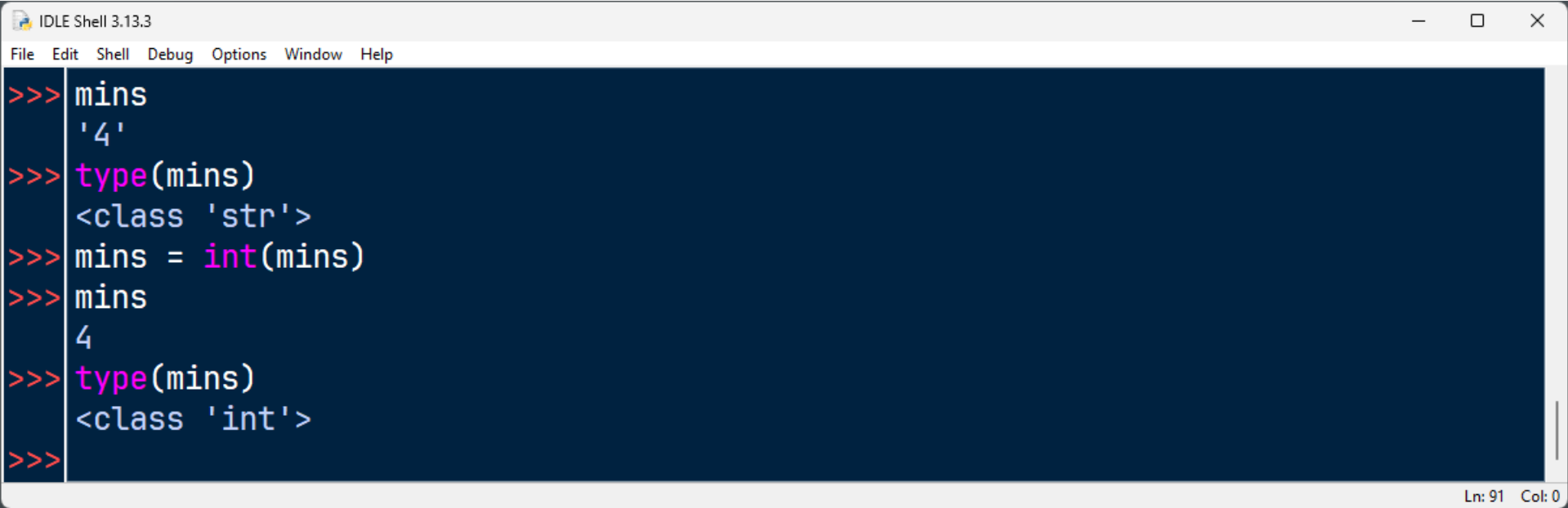
## 사용자로부터 입력 받기

- input 함수를 이용함.
- 사용자가 입력한 정보가 "문자열"로 되어 넘어옴.

[illegible]

# 자료형 변환

- mins의 값은 4가 아니고, '4'임. 즉, 정수가 아니고 문자열임.
- 이것을 정수로 바꾸기 위해서는 int() 라는 함수를 사용함.



```
IDLE Shell 3.13.3
File Edit Shell Debug Options Window Help

>>> mins
'4'
>>> type(mins)
<class 'str'>
>>> mins = int(mins)
>>> mins
4
>>> type(mins)
<class 'int'>
>>>
```

Ln: 91 Col: 0

# 자료형 변환

---

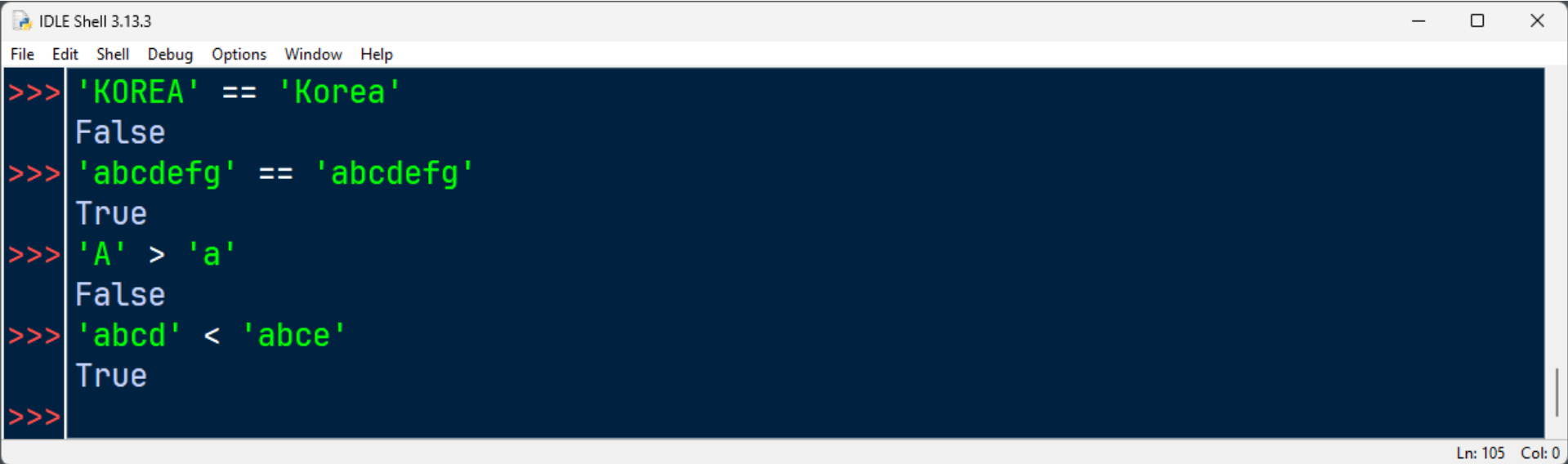
```
>>> str(0)
'0'
>>> str(-3.14)
'-3.14'
>>> int('42')
42
>>> int(1.25)
1
>>> float('3.14')
3.14
>>> float(10)
10.0
```

# 비교 연산 기호

---

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

# 문자열의 비교

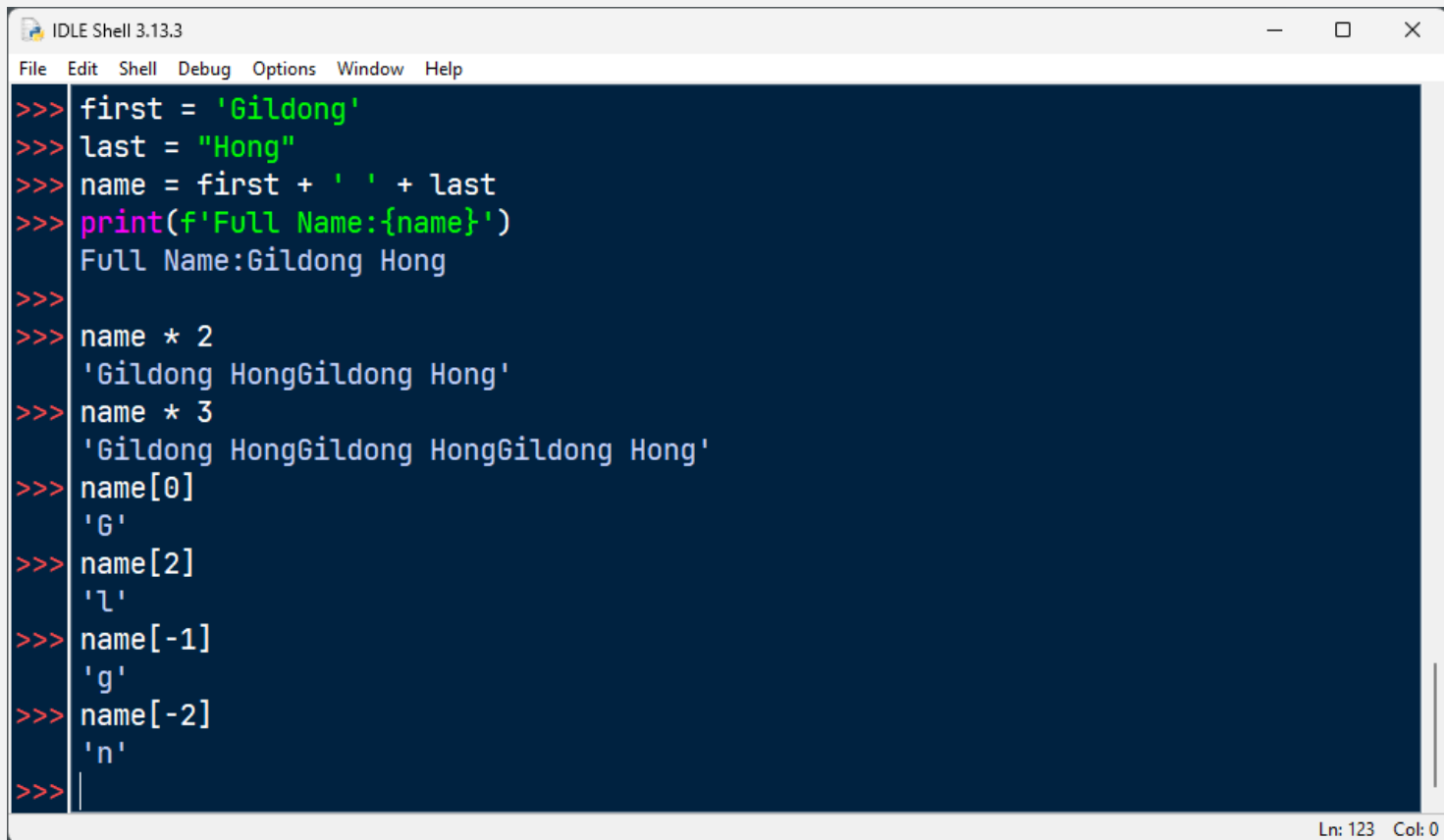


The screenshot shows the IDLE Shell 3.13.3 window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a dark blue command prompt area. The following commands and their outputs are shown:

```
>>> 'KOREA' == 'Korea'
False
>>> 'abcdefg' == 'abcdefg'
True
>>> 'A' > 'a'
False
>>> 'abcd' < 'abce'
True
>>>
```

At the bottom right of the window, the status bar indicates "Ln: 105 Col: 0".

# 문자열 연산



```
IDLE Shell 3.13.3
File Edit Shell Debug Options Window Help
>>> first = 'Gildong'
>>> last = "Hong"
>>> name = first + ' ' + last
>>> print(f'Full Name:{name}')
Full Name:Gildong Hong
>>>
>>> name * 2
'Gildong HongGildong Hong'
>>> name * 3
'Gildong HongGildong HongGildong Hong'
>>> name[0]
'G'
>>> name[2]
'\ '
>>> name[-1]
'g'
>>> name[-2]
'n'
>>> 
```

Ln: 123 Col: 0

# 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', 'jeni', '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', 'jeni', 'rose', 'risa']
>>> unite.remove('momo')
>>> unite
['dahyun', 'jihyo', 'nayun', 'sana', 'zwi', 'jisu', 'jeni', '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

 Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

```
>>> 'zwi' in score
```

```
True
```

```
>>> 'momo' in score
```

```
False
```

```
>>>
```

```
>>> score.clear()
```

```
>>> score
```

```
{}
```

# 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, ... }