

Introduction to Python

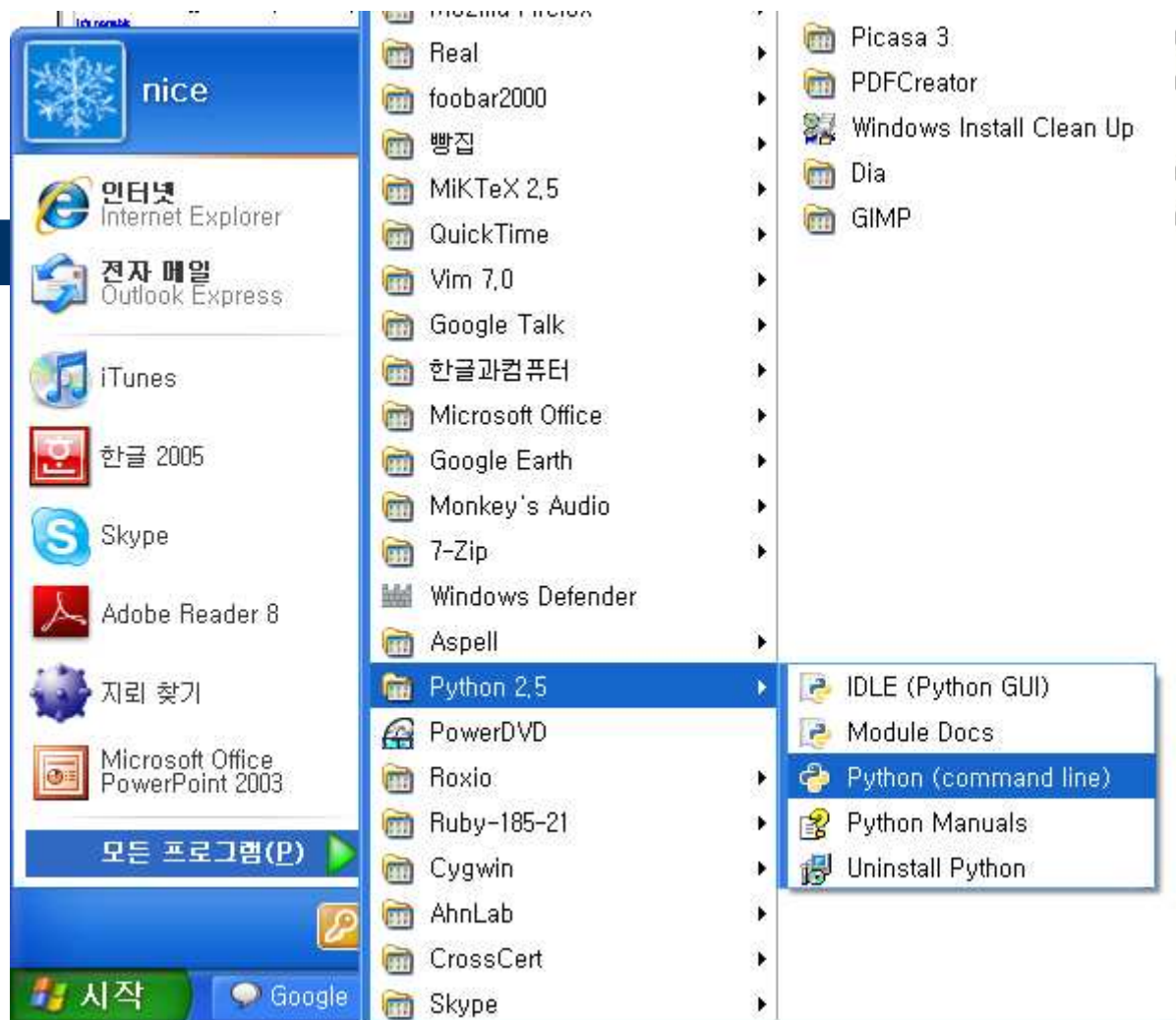


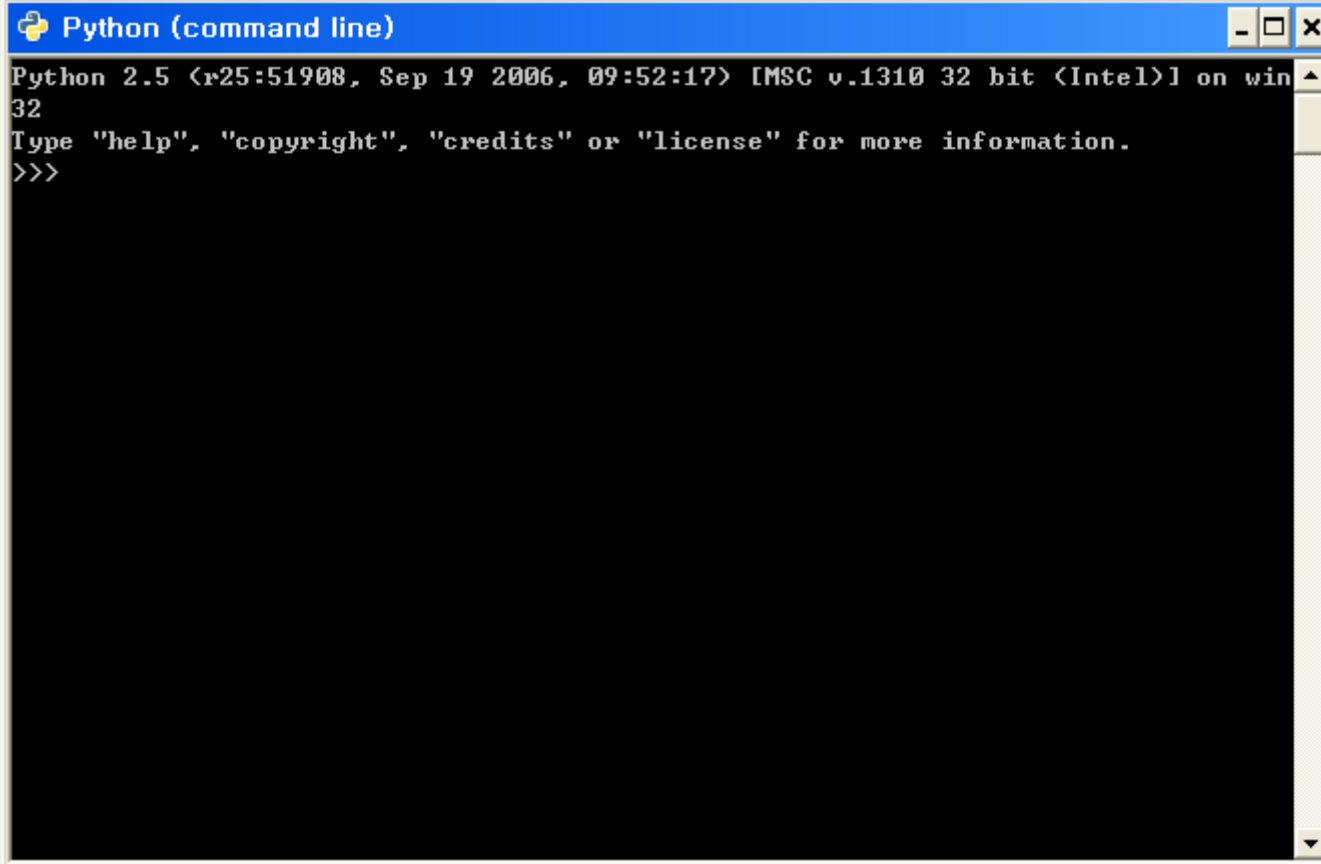
Why Python?

- 배우기 쉬운(?) OOP 언어
- Free software
- 동적 타입, 메모리 관리 자동
- 다양한 라이브러리 (NumPy, SciPy,)
- 프로그램을 빠르게 개발하는데 적합

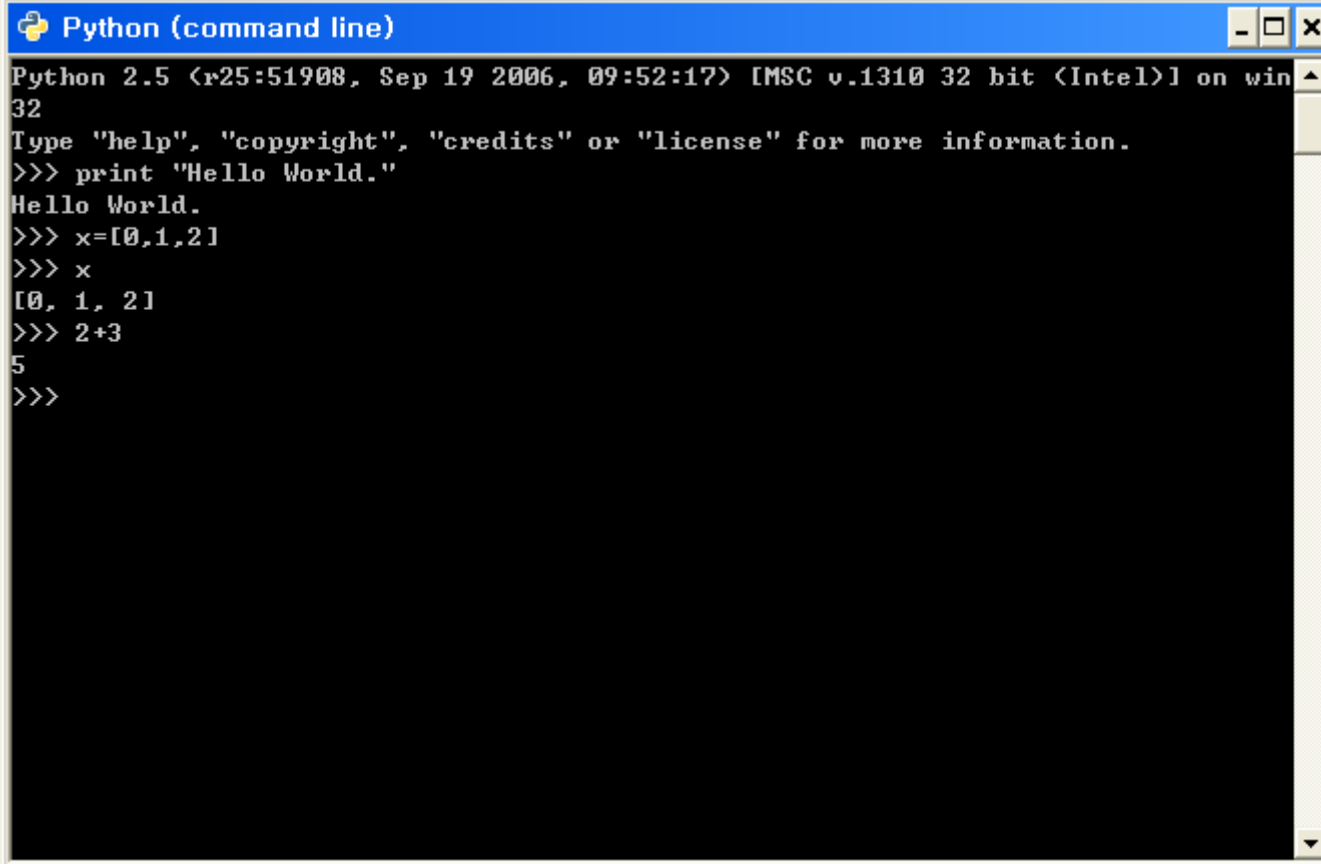
설치

- <http://www.python.org>



A screenshot of a Windows command prompt window titled "Python (command line)". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is black with white text. The text shows the Python 2.5 startup sequence: "Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310 32 bit (Intel)] on win32", followed by a prompt "Type 'help', 'copyright', 'credits' or 'license' for more information.", and finally the interactive prompt ">>>". A blue horizontal bar is positioned to the left of the window, partially overlapping the green L-shaped graphic in the background.

```
Python (command line)
Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

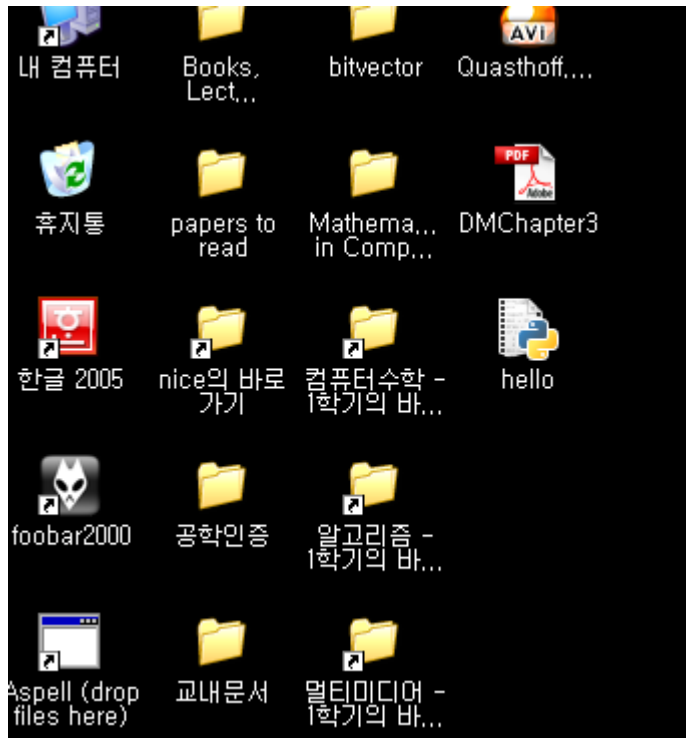


A screenshot of a Windows command prompt window titled "Python (command line)". The window has a blue title bar with standard minimize, maximize, and close buttons. The background is black with white text. The text shows the Python 2.5 version information, a prompt for help, and several interactive commands and their outputs: printing "Hello World.", creating a list [0, 1, 2], and calculating 2+3 to get 5. A blue horizontal bar is positioned to the left of the window, partially overlapping the green background.

```
Python 2.5 (r25:51908, Sep 19 2006, 09:52:17) [MSC v.1310 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Hello World."
Hello World.
>>> x=[0,1,2]
>>> x
[0, 1, 2]
>>> 2+3
5
>>>
```

Python program 실행

- 확장자가 .py로 끝나는 텍스트 파일을 만들고 더블클릭



수의 *Type*

- 수
 - 정수: 0, 1, 2, - 56, ...
>>> 7/2
3
 - 긴 정수: 정수 끝에 L 추가. 임의의 길이 가능
 - 실수: 7.0, - 3.2, ...
>>> 7.0/2.0
3.5
 - 8진수: 0177 (0으로 시작), 16진수: 0x1a (0x로 시작)
>>> 0x1a+33
59
 - 복소수(.....) 3+4j

연산자

- 사칙연산
 - $a+b$, $a-b$, $a*b$, a/b
 - 거듭제곱 $a**b$
- 비교 연산
 - $a < b$, $a > b$, $a \leq b$, $a \geq b$
 - $a == b$, $a != b$
- 비트 연산
 - Bitwise or: $a | b$, Bitwise xor: $a \wedge b$ Bitwise and: $a \& b$
 - Shift a left or right by b bits: $a \ll b$, $a \gg b$
- 형 변환
 - $\ggg 3+5.5$
8.5

문자열

- 홑따옴표 ‘’, 겹따옴표 “” 로 표현
 - Examples: 'abc', "ABC"
- 문자열 더하기, 곱하기(....)

```
>>> 'abc'+ 'def'
```

```
'abcdef'
```

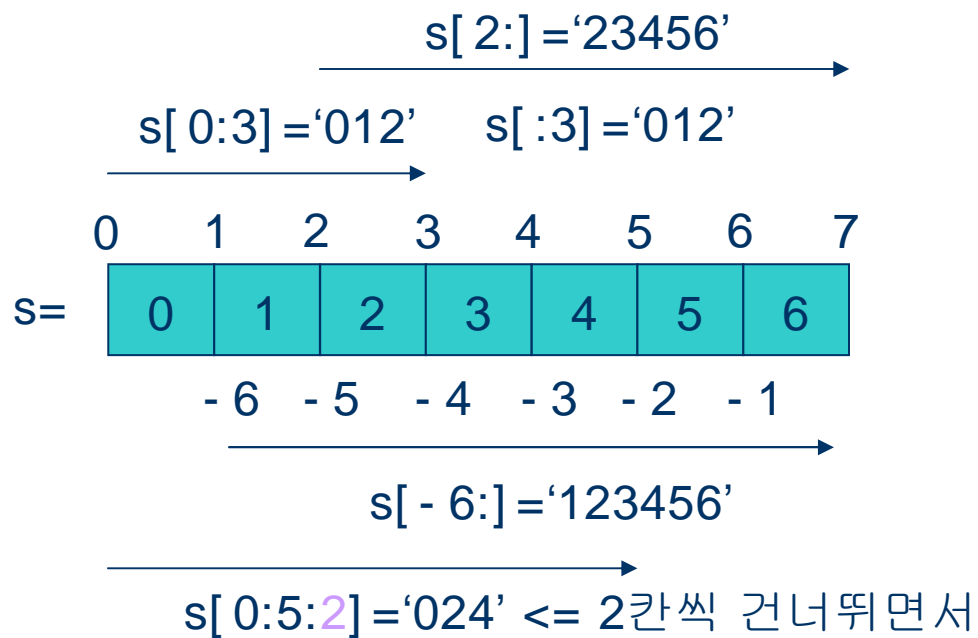
```
>>> 'abc'* 3
```

```
'abcabcabc'
```

문자열 연산

```
Python (command line)
>>> 3+5.5
8.5
>>> s="0123456"
>>> s[0]
'0'
>>> s[0,1]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: string indices must be integers
>>> s[0:1]
'0'
>>> s[1:1]
''
>>> s[0:3]
'012'
>>> s[0:]
'0123456'
>>> s[:5]
'01234'
>>> s[:-1]
'012345'
>>> s[-1:]
'6'
>>>
```

문자열 연산 (*What the...*)



```
>>>len(s)
7
>>>s[-1]
'6'
```

리스트 (*list*)

- [] 안의 내용 (C의 배열과 미묘하게 다름)
 - 리스트 안에는 수, 문자열, 다른 리스트(...) 등이 섞여서 올 수 있음
 - L1 = [0,1,2,3], L2 = ['zero', 'one'],
L3 = [0,1,[2,3], 'three', ['four,one']], L4 = []
- 리스트의 원소는 스트링과 마찬가지로 접근 가능
- 리스트의 내용은 바뀔 수 있으며, 크기도 변경 가능

```
>>> L1 = [0,1,2,3]
>>> L1[0] = 4
>>> L1[0]
4
```

리스트 연산 예

- Indexing: `L1[i]`, `L2[i][j]`
- Slicing: `L3[i:j]`
- Concatenation:

```
>>> L1 = [0,1,2]; L2 = [3,4,5]
>>> L1+L2
[0,1,2,3,4,5]
```
- Repetition:

```
>>> L1*3
[0,1,2,0,1,2,0,1,2]
```
- Appending:

```
>>> L1.append(3)
[0,1,2,3]
```
- Sorting:

```
>>> L3 = [2,1,4,3]
>>> L3.sort()
[1,2,3,4]
```

리스트 연산 예

- Reversal:

```
>>> L4 = [4,3,2,1]
>>> L4.reverse()
>>> L4
[1,2,3,4]
```
- Shrinking:

```
>>> del L4[0]
>>> L4
[2,3,4]
```
- Index and slice assignment:

```
>>> L1[1] = 1
>>> L2[1:4] = [4,5,6]
```
- Making a list of integers:

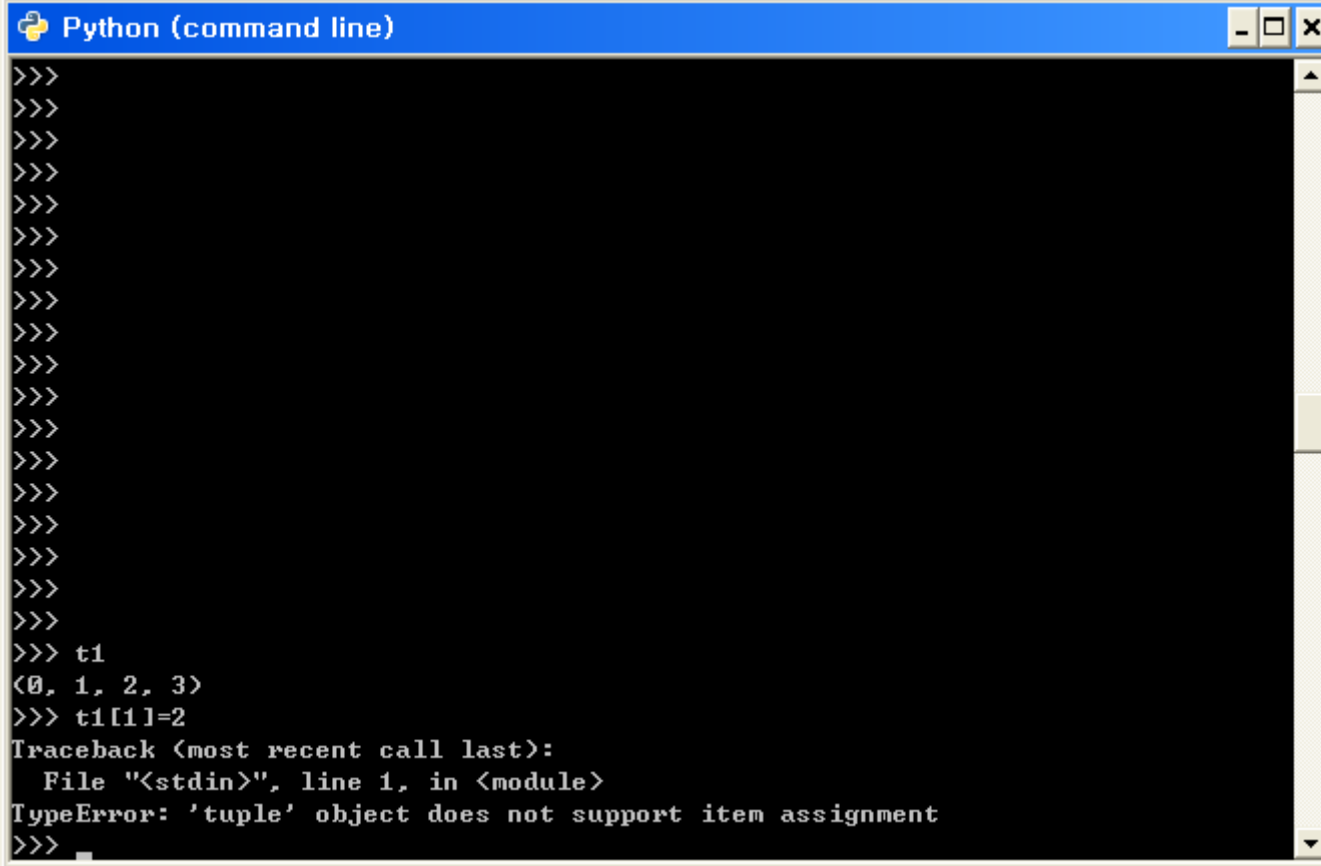
```
>>> range(4)
[0,1,2,3]
>>> range(1,5)
[1,2,3,4]
```

튜플 (*Tuple*)

- 리스트와 비슷하나, ()로 싸여있고 값을 바꿀 수 없다.
- Concatenation:

```
>>> t1 = (0,1,2,3); t2 = (4,5,6)
>>> t1+t2
(0,1,2,3,4,5,6)
```
- Repetition:

```
>>> t1*2
(0,1,2,3,0,1,2,3)
```
- 길이: `len(t1)` (리스트, 스트링에도 적용)



A screenshot of a Python command line window. The window title is "Python (command line)". The terminal shows a series of empty prompts ">>>". The user enters the following code:

```
>>> t1
<0, 1, 2, 3>
>>> t1[1]=2
```

This results in a `TypeError: 'tuple' object does not support item assignment`. The error message is displayed as follows:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>>
```

The window includes standard window controls (minimize, maximize, close) and a scrollbar on the right side.

미묘한 차이

```
>>> a = 2
>>> b = a # a and b are both numbers, and are thus immutable
>>> a = 3
>>> b
2

>>> La = [0,1,2]
>>> Lb = La # La and Lb are both lists, and are thus mutable
>>> La = [1,2,3]
>>> Lb
[1,2,3]
```

Python (command line)

```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> import copy
>>> La=[0,1,2]
>>> Lb=copy.copy(La)
>>> Lb
[0, 1, 2]
>>> La
[0, 1, 2]
>>> La[2]=3
>>> La
[0, 1, 3]
>>> Lb
[0, 1, 2]
>>>
```

if

```
# inside the interpreter
>>> if condition:
...     action
...
...
```

```
# inside a script
if condition:
    action
```

매우 중요한 차이점: Python에서는 C처럼 { }을 써서 블록을 표시하지 않고, 들여쓰기를 통해서 블록을 표시(.....)

```
if condition1:
    action1
elif condition2:
    action2
else:
    action3
```

복잡한 *if*

`and` 와 `or` 를 사용하여 조건 결합

```
if condition1 and condition2:
```

```
    action1
```

```
if condition1 or condition2:
```

```
    action2
```

Conditions may be expressed using the following operations:

`<`, `<=`, `>`, `>=`, `==`, `!=`, `in`

별로 안그럴싸한 *if* 예제

```
>>> x = 2; y = 3; L = [0,1,2]
>>> if (1 < x <= 3 and 4 > y >= 2) or (1 == 1 or 0 !=
    1) or 1 in L:
...     print 'Hello world'
...
Hello world
>>>
```

while

while condition:

action

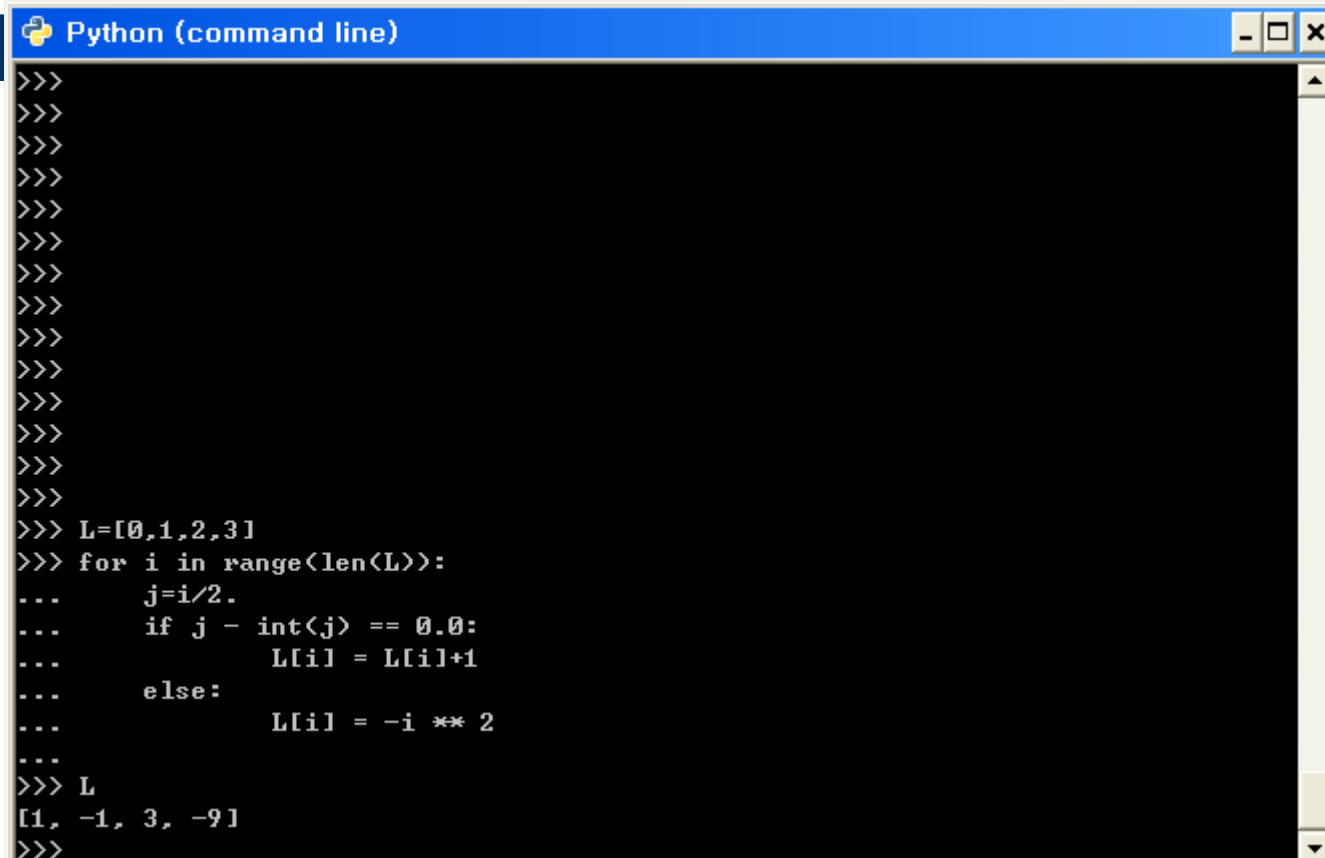
```
>>> x = 1
>>> while x < 4: # as long as x < 4...
...   print x**2 # print the square of x
...   x = x+1    # increment x by +1
...
1
4           9
>>>
```


for

for item i in set s:
action on item i

```
>>> for i in range(1,7):  
...   print i, i**2, i**3, i**4  
...  
1 1 1 1  
2 4 8 16  
3 9 27 81  
4 16 64 256  
5 25 125 625  
6 36 216 1296  
>>>
```

for 예제



```
Python (command line)
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> L=[0,1,2,3]
>>> for i in range(len(L)):
...     j=i/2.
...     if j - int(j) == 0.0:
...         L[i] = L[i]+1
...     else:
...         L[i] = -i ** 2
...
>>> L
[1, -1, 3, -9]
>>>
```

함수

def func(args):
 return values

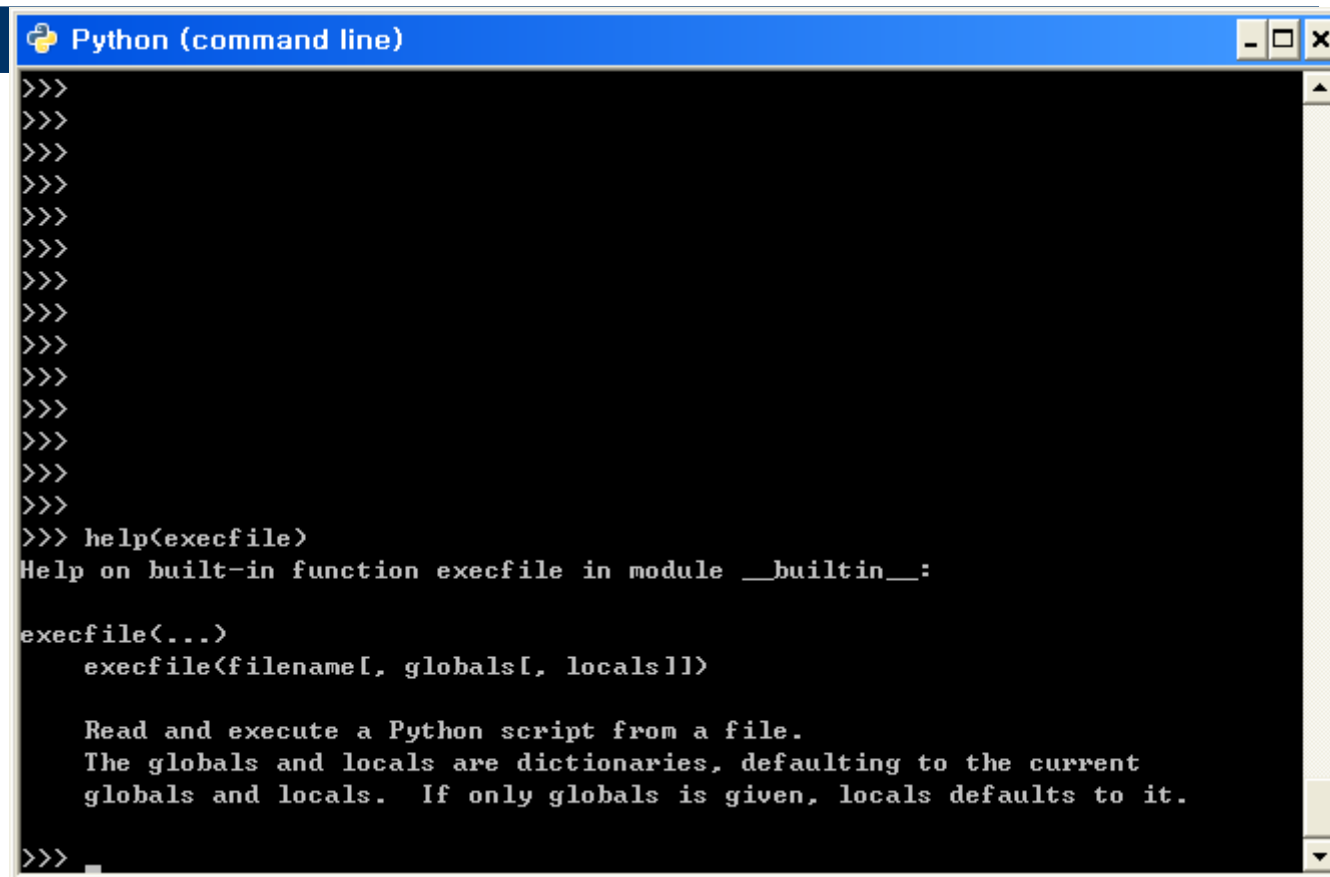
```
Python (command line)
>>> def fib(n):
...     if n == 1:
...         return 1
...     elif n == 2:
...         return 2
...     else:
...         return fib(n-1) + fib(n-2)
...
>>> fib(3)
3
>>> fib(1)
1
>>> fib(2)
2
>>> fib(5)
8
>>> fib(4)
5
>>>
>>>
>>>
>>>
>>>
>>>
```


!!

Python (command line)

```
>>> def hello():
...     print("Hello World")
...
>>> def hi():
...     print("Hi World")
...
>>> A
[1, 2, 3]
>>> B
[3, 4, 5]
>>> def f5(func, list):
...     print list
...     func()
...
>>> f5(hello, A)
[1, 2, 3]
Hello World
>>> f5(hi, A)
[1, 2, 3]
Hi World
>>> f5(hello, B)
[3, 4, 5]
Hello World
>>> f5(hi, A)
[1, 2, 3]
```

부명



```
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>> help(execfile)  
Help on built-in function execfile in module __builtin__:  
  
execfile(...)  
    execfile(filename[, globals[, locals]])  
  
    Read and execute a Python script from a file.  
    The globals and locals are dictionaries, defaulting to the current  
    globals and locals.  If only globals is given, locals defaults to it.  
  
>>>
```

주요 라이브러리들

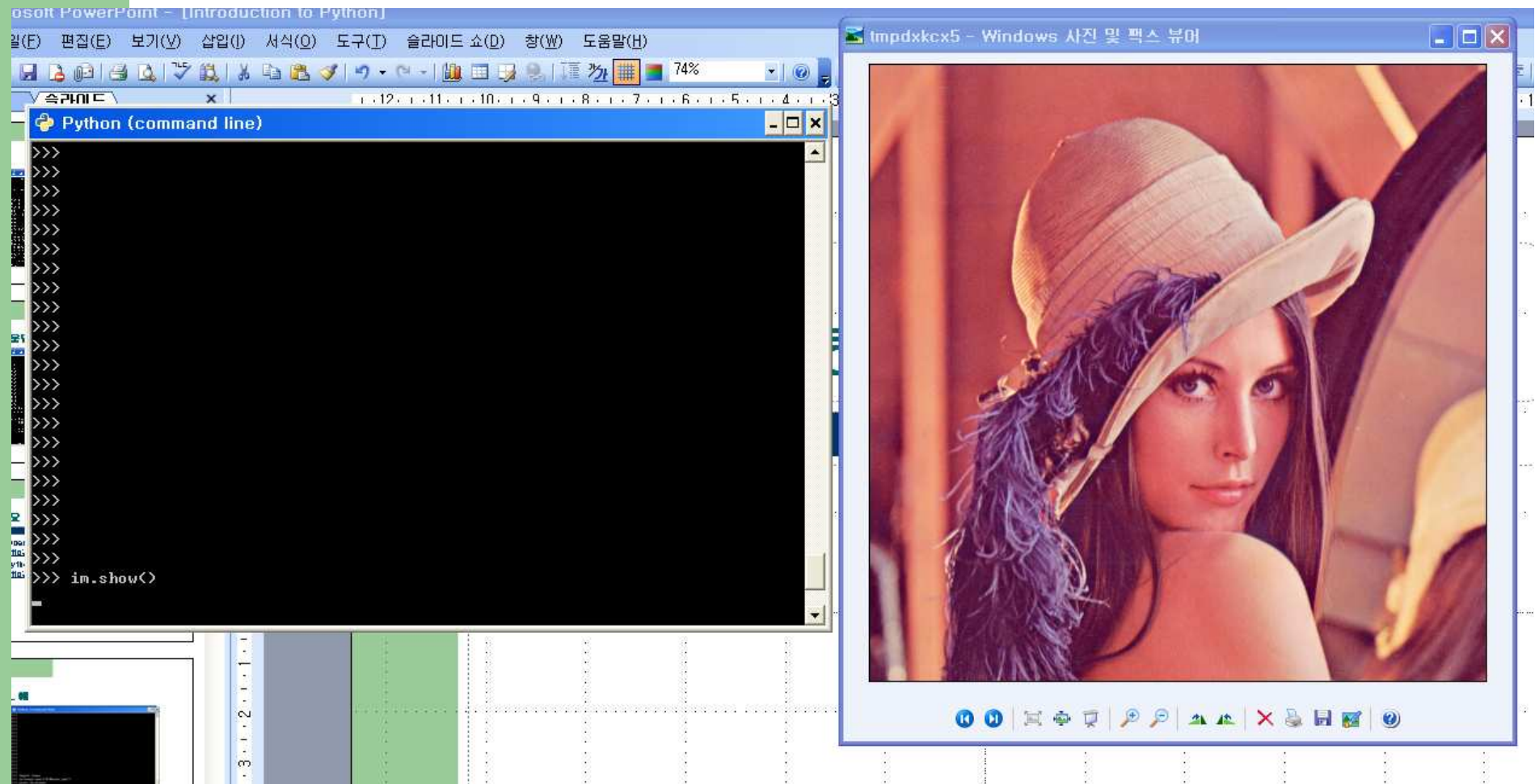
- OpenGL 프로그래밍: PyOpenGL
<http://pyopengl.sourceforge.net/>
- Python Image Library (PIL)
<http://www.pythonware.com/products/pil/>

PIL 예

Python (command line)

```
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>> import Image  
>>> im=Image.open('D:\#Lenna.png')  
>>> print im.format  
PNG  
>>> print im.size  
(512, 512)  
>>> print im.mode  
RGB
```

im.show()



Python (command line)

```
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>> in.resize((256,256)).show()
```

32

```
>>> r,g,b=in.split()  
>>> Image.merge("RGB", (b,g,r)).s
```

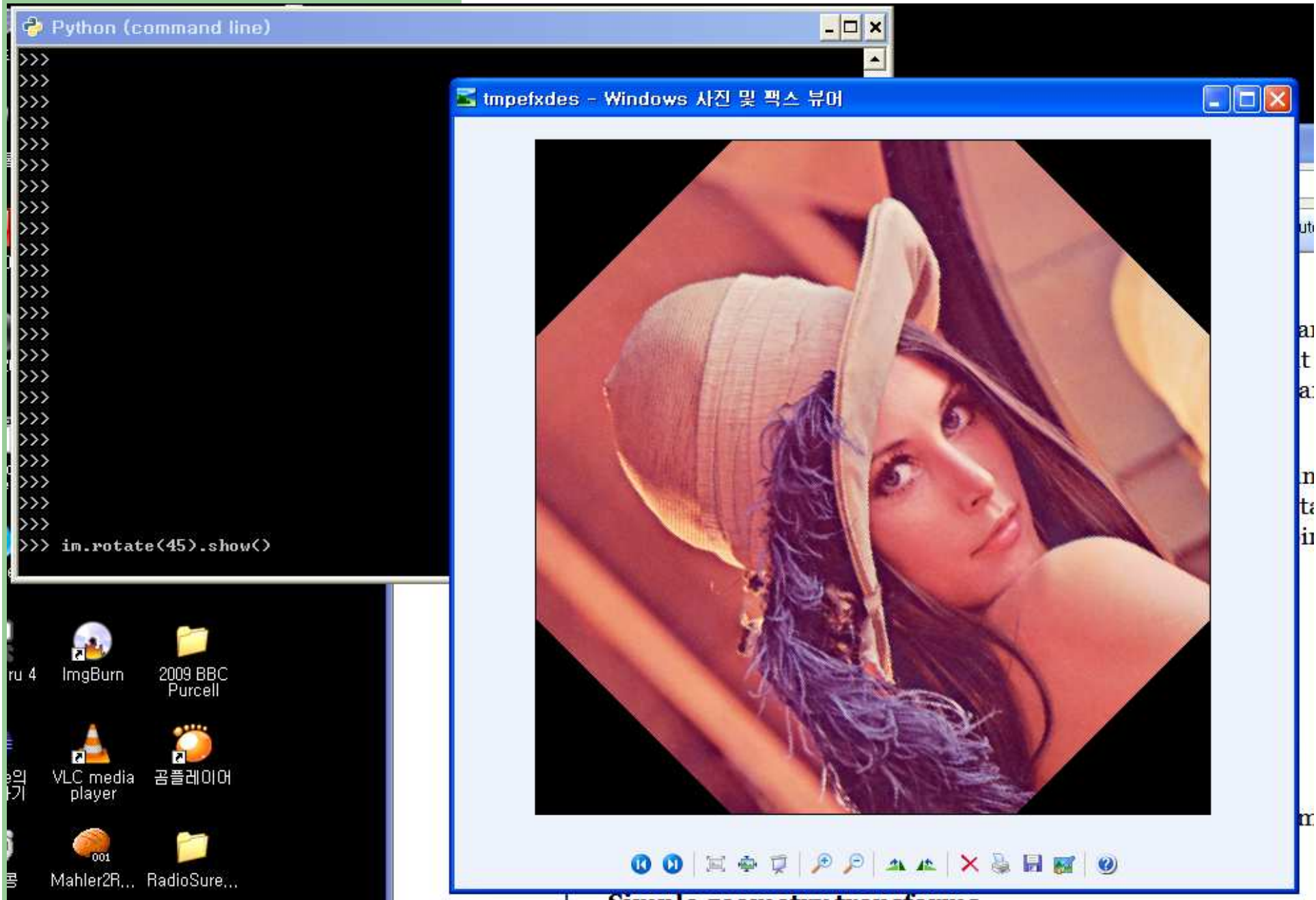
- OpenGL 학습코딩 : PyOpenGL
<http://www.opengl.org/resources/>
- Python Image Library (PIL)
<http://www.pythonware.com/products/pil/>

33



tmps8c9wl - Windows 사진 및 팩스 뷰어





Simple geometry transforms