

Memory & Variables

국민대 소프트웨어학부

수업목표

- 입력(input)
- 출력(output)
- 메모리(memory)와 변수(variable)
- 문자열
- Turtle graphics

입력(input)

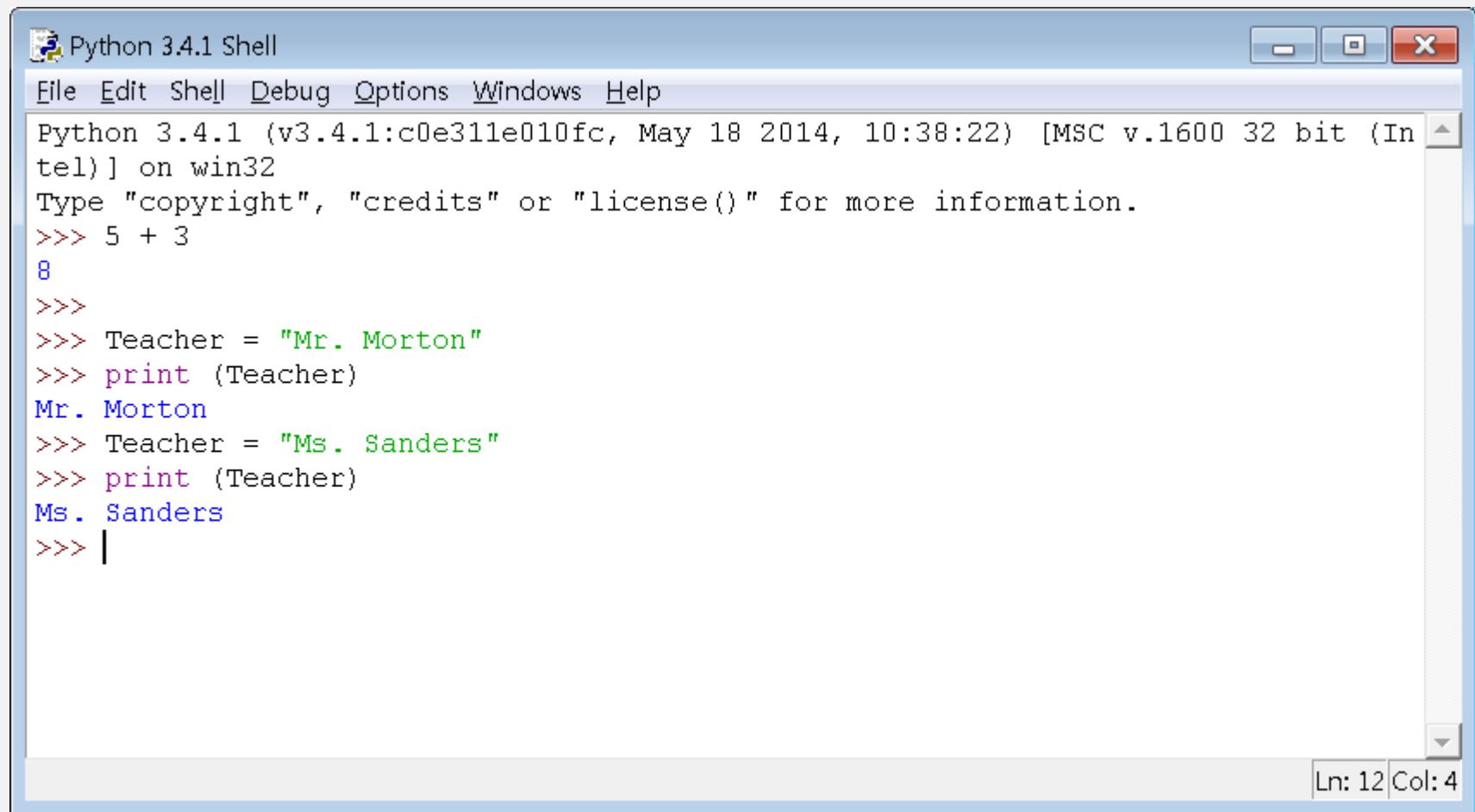
- 컴퓨터(PC) 입력
 - 키보드
 - 표준 입력 장치: standard input
 - 마우스
 - 파일
 - ...
- 스마트폰 입력
 - 키패드
 - 터치 스크린
 - 파일
 - NFC 전파
 - 음성인식
 - 얼굴인식
 - ...

출력(output)

- 컴퓨터(PC) 출력
 - 모니터
 - 표준 출력 장치: standard output
 - 프린터
 - 파일
 - ...
- 스마트폰 출력
 - 화면
 - 소리
 - 파일
 - NFC 전파
 - ...

입력과 출력

- 키보드 입력, 모니터 출력



The screenshot shows the Python 3.4.1 Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The main window displays the following Python session:

```
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (In tel)] on win32
Type "copyright", "credits" or "license()" for more information.

>>> 5 + 3
8
>>>
>>> Teacher = "Mr. Morton"
>>> print (Teacher)
Mr. Morton
>>> Teacher = "Ms. Sanders"
>>> print (Teacher)
Ms. Sanders
>>> |
```

The status bar at the bottom right indicates Ln: 12 Col: 4.

입력과 출력

- 키보드 입력, 모니터 출력

```
# print 함수 문법  
print(<출력하고 싶은 것1>, <출력하고 싶은 것2>, ...)
```

```
# 하나만 출력할 때  
print("안녕하세요")  
print("만나서 반갑습니다.")
```

```
# 여러개를 출력 할 때  
print("저는", "소프트웨어 학부", "4학년", "김영재입니다.")  
print("학번은", 20153160, "입니다.")
```

```
# 아무것도 입력 하지 않으면 개행(줄바꿈)  
print("- - -")  
print()  
print()  
print("- - -")
```

```
# end 를 사용하면 개행을 하지 않을 수 있습니다.  
print("개행 하지 않겠습니다", end=".")  
print("end=값 을 사용하면 해당 값으로 끝낼 수 있습니다.", end=">>")
```

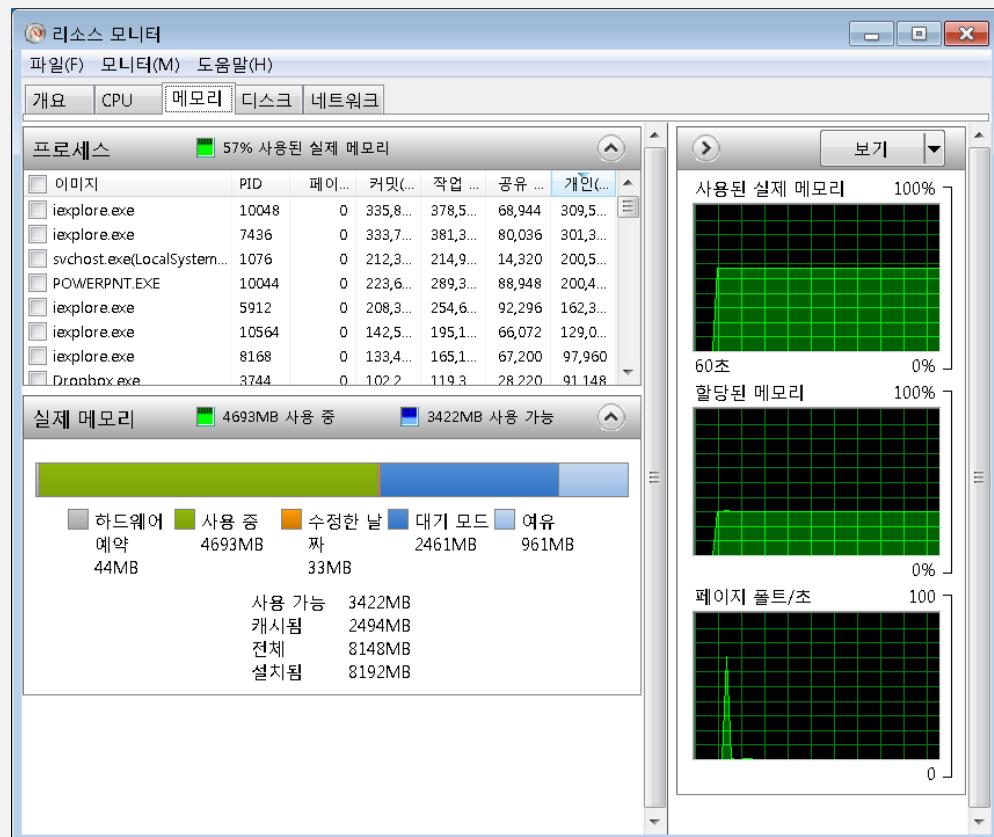
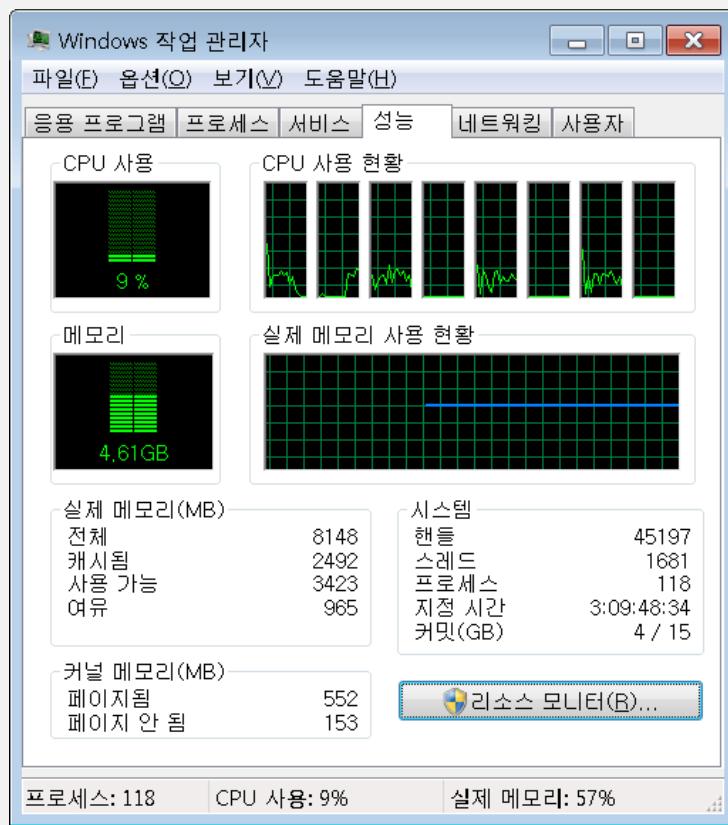
메모리와 변수

- “Teacher”는 어디에 저장되나? → 컴퓨터 메모리

The screenshot shows a web browser window displaying a product page for a Samsung DDR3 4GB RAM module. The URL in the address bar is <http://prod.danawa.com/info/?PCODE=1304396&cate1=861&cate2=874&cate3=11043&cate4=0&keyword=%B8%DE%B8%F0%B8%AE#b>. The page title is "삼성전자 DDR3 4G...". The left sidebar menu includes links for Home, Price Comparison, Product Details (selected), News/Reviews/Usage Guide, Daewoo Q&A, and Product Reviews. The main content area shows the product name "삼성전자 DDR3 4G PC3-12800 (정품)" and features a large image of the RAM module. To the right, there are several smaller images labeled "상세 이미지" (Detail Image) and "사용자 리뷰" (User Review). At the bottom, there are buttons for "상품 정보 오류 신고" (Report Product Error) and "상품 이미지 등록" (Register Product Image).

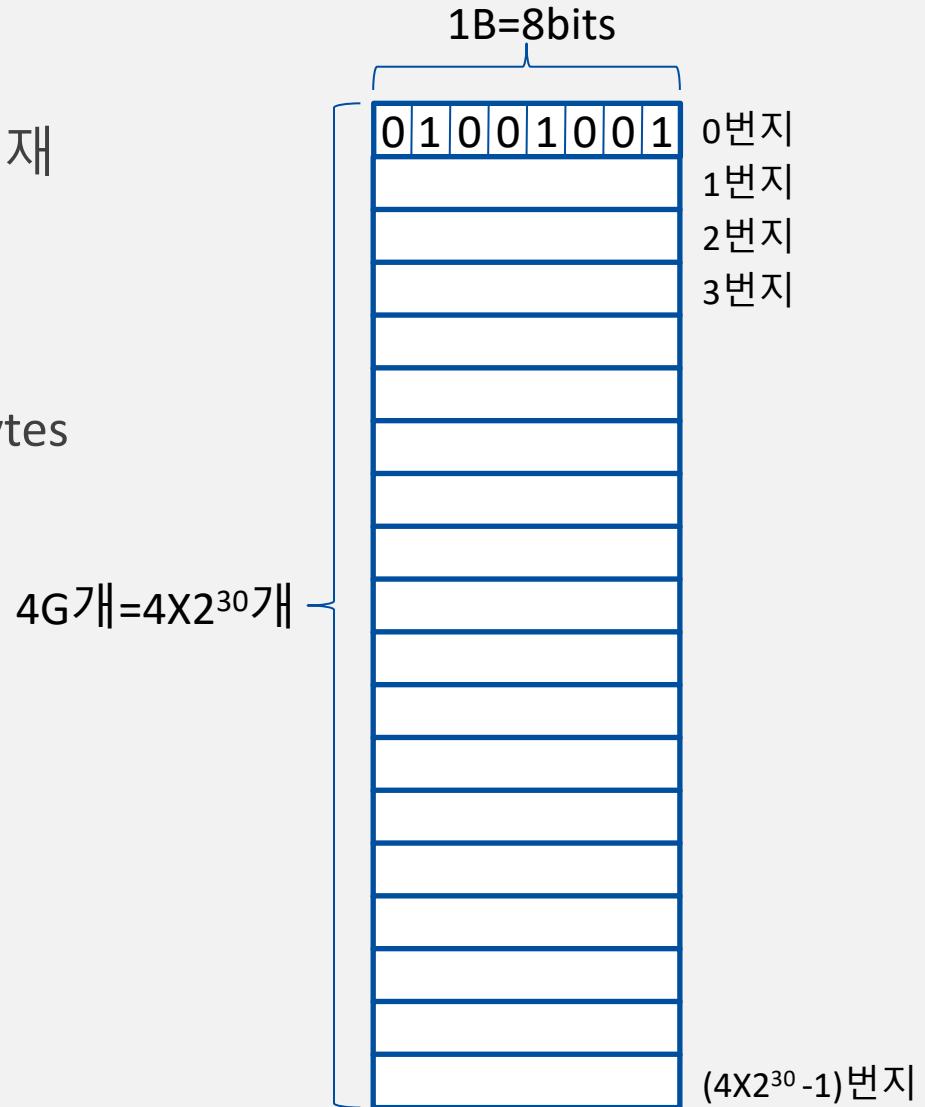
메모리와 변수

- 윈도우 ‘Ctrl’ + ‘Shift’ + ‘Esc’



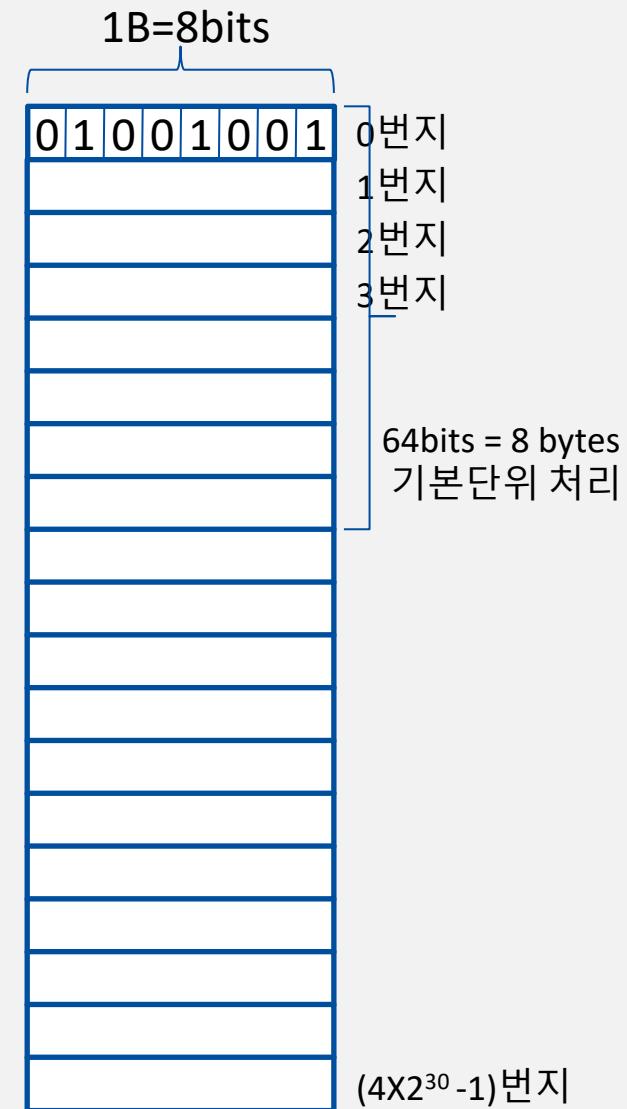
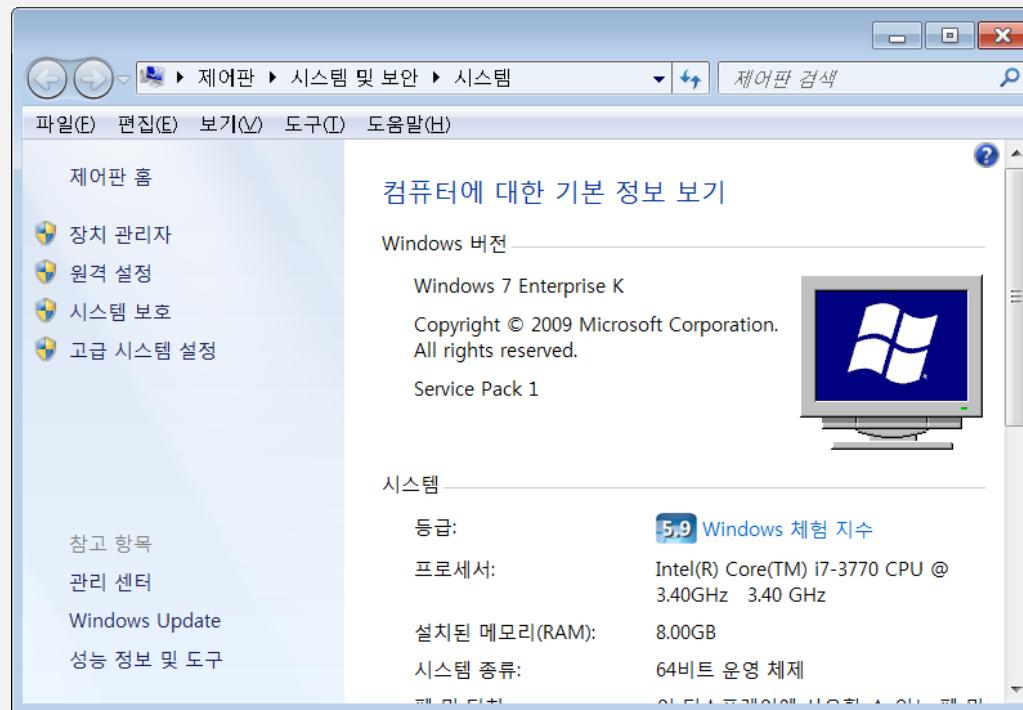
메모리와 변수

- 메모리용량 4GB
 - 1 Byte(=8 bits)가 4G개 만큼 존재
 - 컴퓨터 기본 단위는 bit
 - 0과 1 표현
 - $1,000 = 10^3 = 2^{10} = 1024$
 - $4GB = 4 \times 10^9 Bytes = 4 \times 2^{30} Bytes$



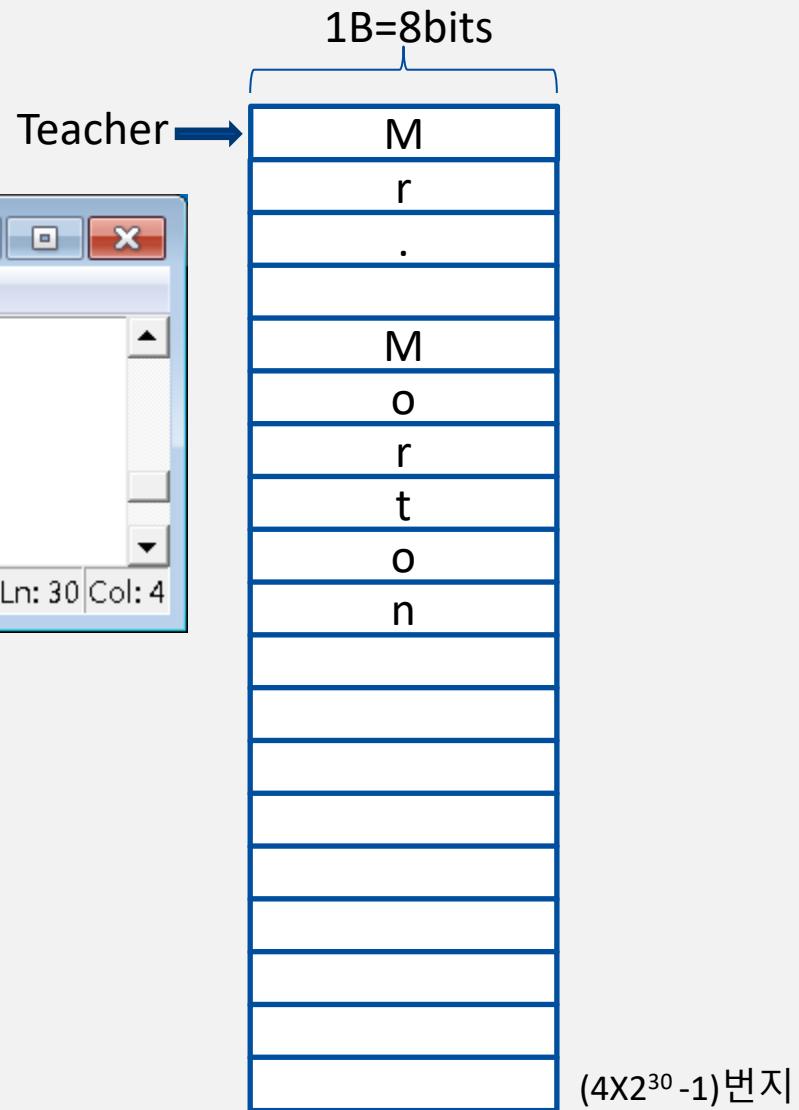
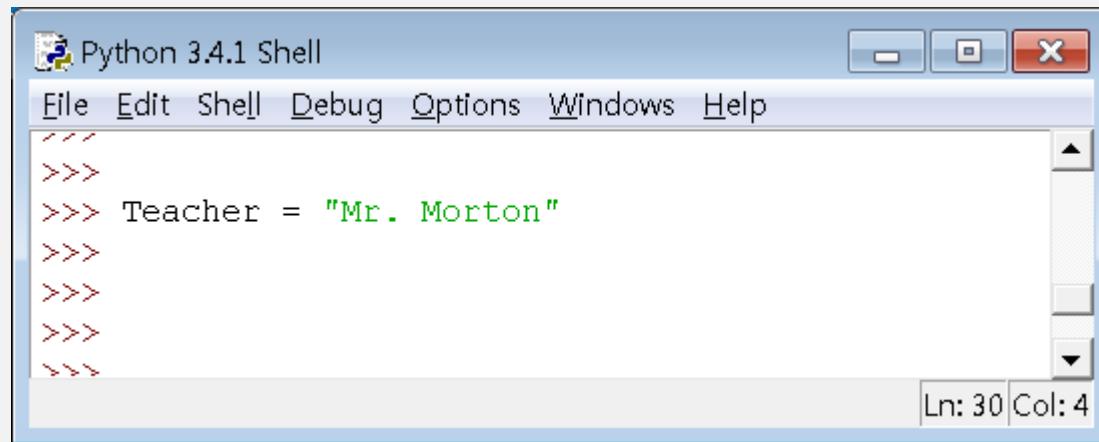
메모리와 변수

- 메모리용량 4GB
 - 윈도우 운영체제 64bit vs 32bit
 - 기본 처리 단위가 8Byte 또는 4Byte의 차이



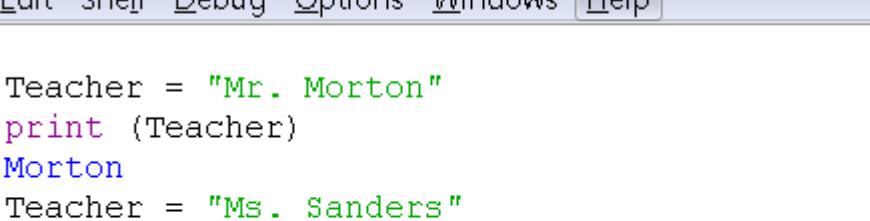
메모리와 변수

- 변수 “Teacher”



메모리와 변수

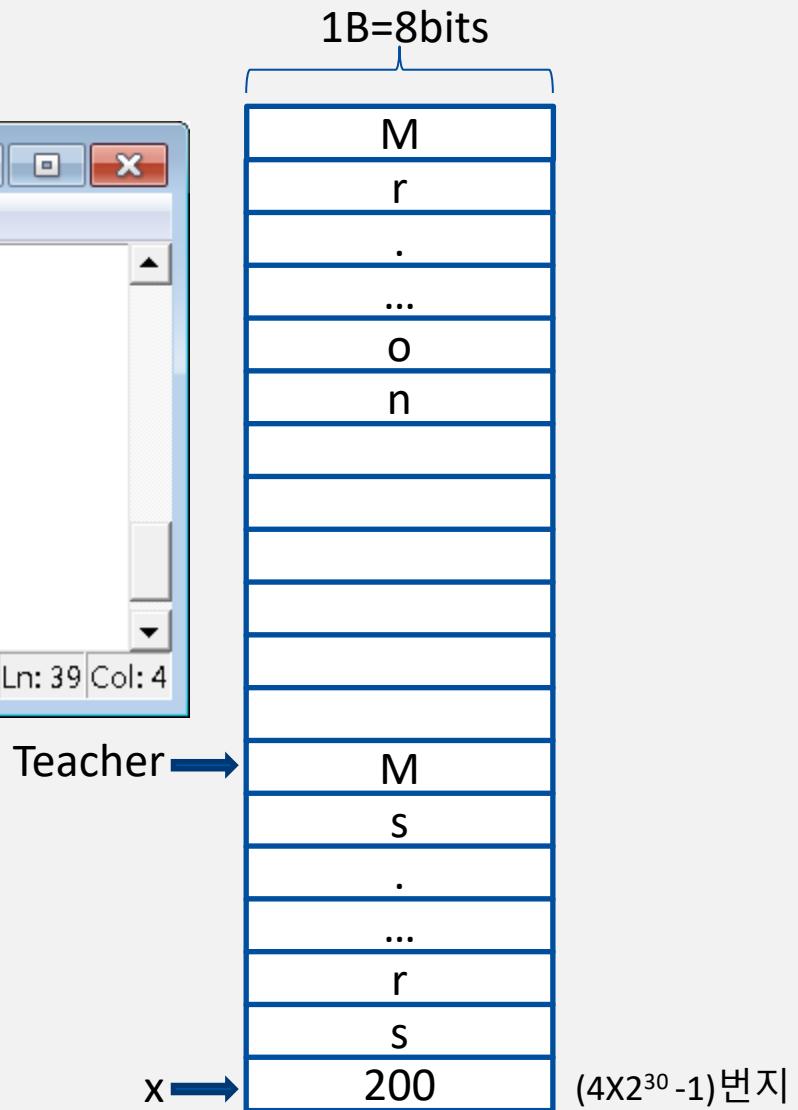
- 변수 “Teacher”



The image shows a screenshot of the Python 3.4.1 Shell window. The title bar reads "Python 3.4.1 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main area displays the following Python code and its output:

```
>>>
>>> Teacher = "Mr. Morton"
>>> print (Teacher)
Mr. Morton
>>> Teacher = "Ms. Sanders"
>>> print (Teacher)
Ms. Sanders
>>> x = 200
>>> print (x)
200
```

In the bottom right corner, there is a status bar with "Ln: 39 Col: 4".



- 변수 “x”

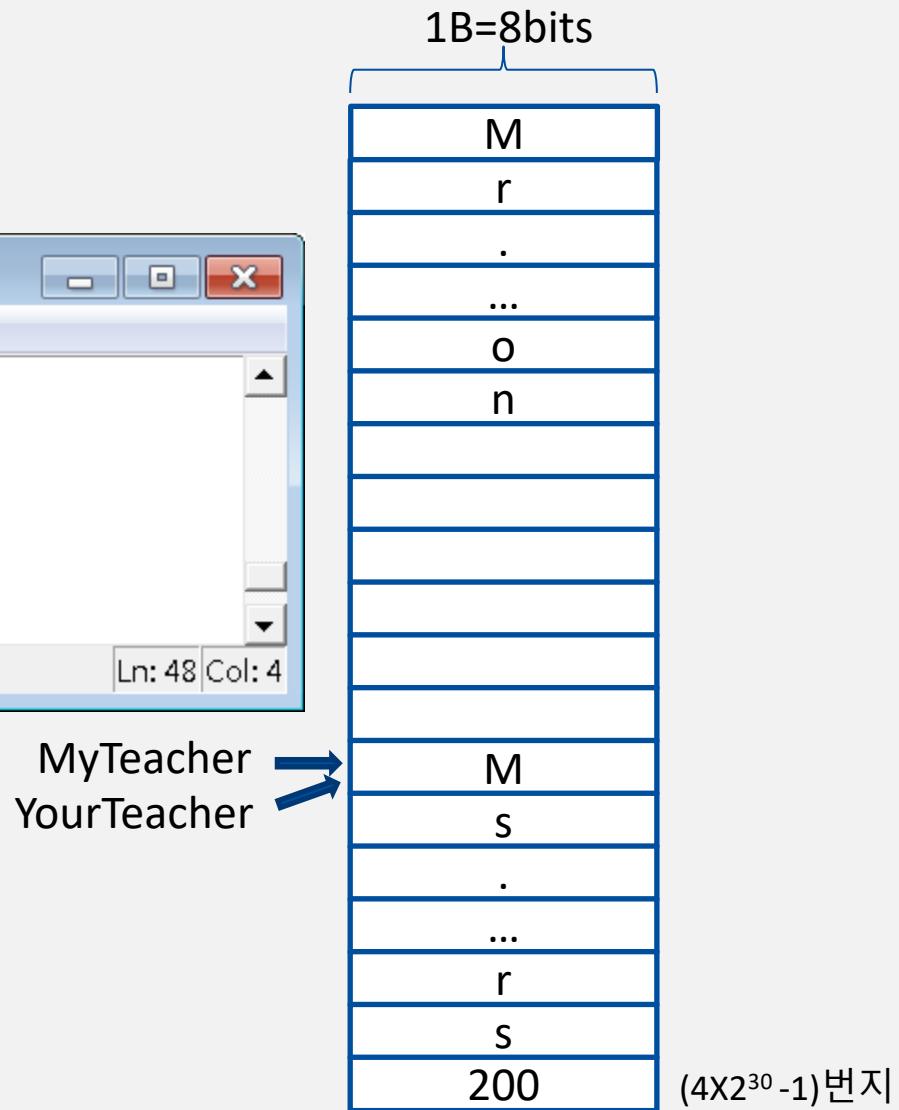
메모리와 변수

- 변수 “MyTeacher”
- 변수 “YourTeacher”

The screenshot shows the Python 3.4.1 Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The command history pane displays the following code:

```
>>> MyTeacher = "Ms. Sanders"
>>> YourTeacher = MyTeacher
>>> MyTeacher
'Ms. Sanders'
>>> YourTeacher
'Ms. Sanders'
>>>
```

The status bar at the bottom right indicates Ln: 48 Col: 4.



메모리와 변수

- `id()` 함수
 - 객체의 id값 리턴

```
Python 3.4.1: 2.1example1.py - C:/Users/mkyoon/Desktop/2.1ex...
File Edit Format Run Options Windows Help
Teacher = "Mr. Morton"
print("Teacher id=", id(Teacher))
Teacher = "Ms. Sanders"
print("Teacher id=", id(Teacher))

MyTeacher = "Ms. Sanders"
YourTeacher = MyTeacher
print("MyTeacher id=", id(MyTeacher))
print("YourTeacher id=", id(YourTeacher))
```

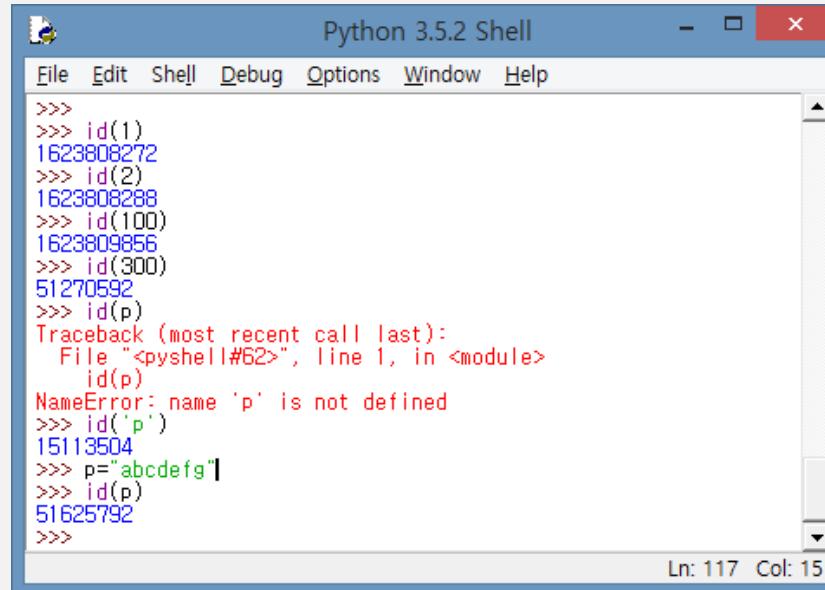
Ln: 3 Col: 23

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> ===== RESTART =====
=====
>>>
Teacher id= 49972832
Teacher id= 49972952
MyTeacher id= 49972952
YourTeacher id= 49972952
>>> |
```

Ln: 38 Col: 4

메모리와 변수

- `id()` 함수
 - 자주 사용하는 객체는 파이썬이 미리 생성해 놓고 있음
 - 예) 숫자 -5~+256
 - `id(p)`에서 에러 발생 이유는?
 - PC마다(또는 부팅 시마다) `id()`값이 다른 이유는?



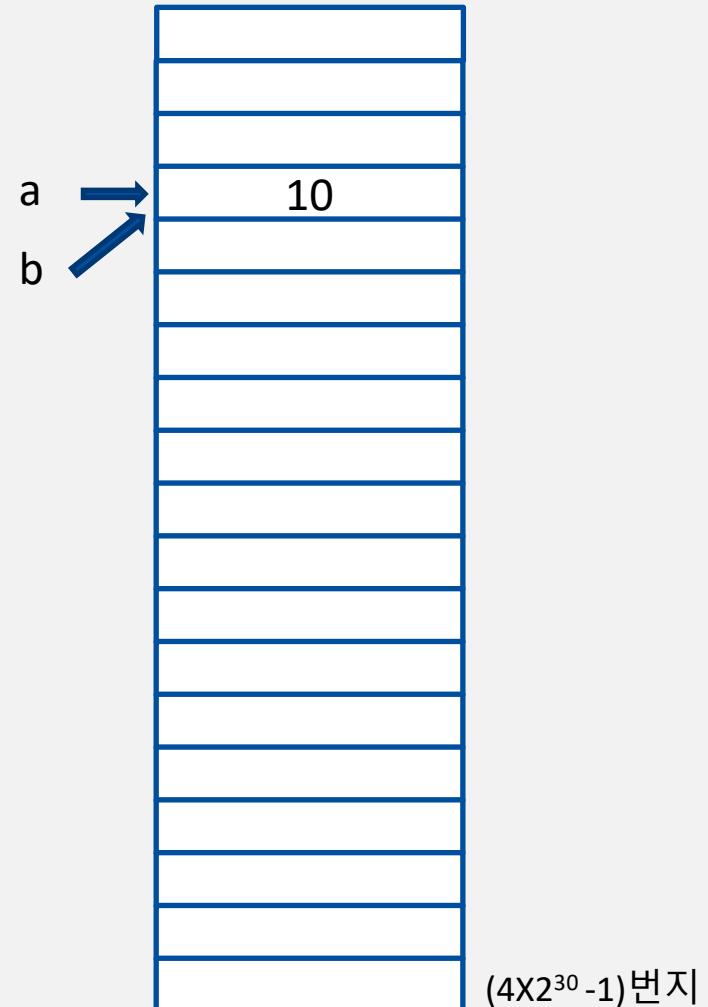
The screenshot shows a Windows-style window titled "Python 3.5.2 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays a Python session:

```
>>>
>>> id(1)
1623808272
>>> id(2)
1623808288
>>> id(100)
1623809856
>>> id(300)
51270592
>>> id(p)
Traceback (most recent call last):
  File "<pyshell#62>", line 1, in <module>
    id(p)
NameError: name 'p' is not defined
>>> id('p')
15113504
>>> p="abcdefg"
>>> id(p)
51625792
>>>
```

The status bar at the bottom right indicates "Ln: 117 Col: 15".

메모리와 변수

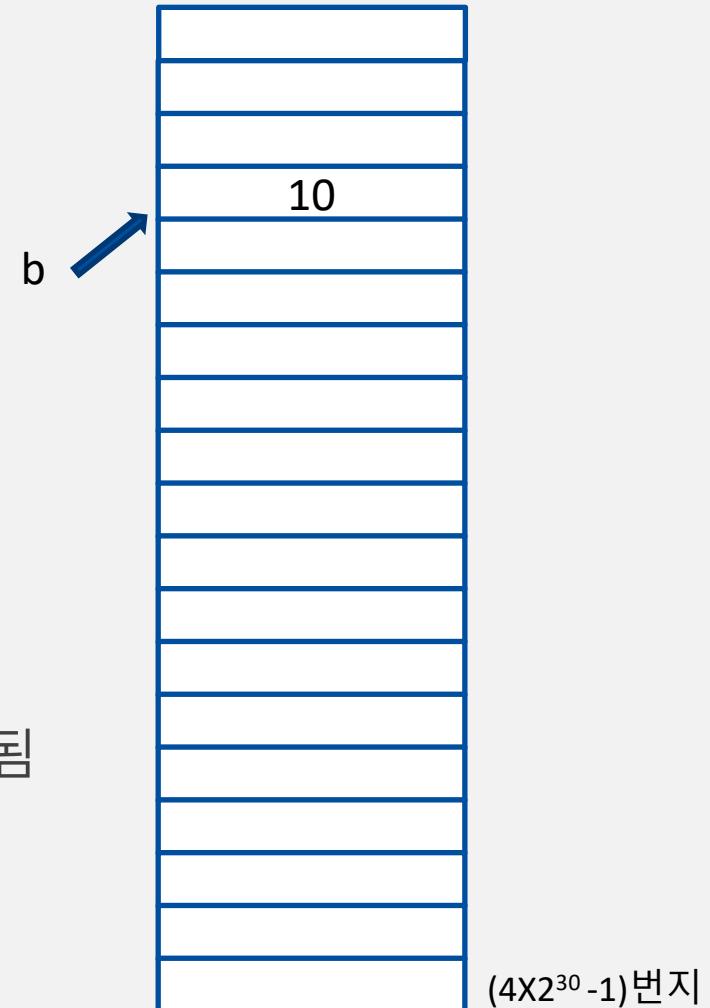
- 변수 이름 vs 값 객체
 - $a=10$
 - a: 변수 이름
 - 10: 값 객체
 - $b=a$
 - 이름만 하나 더 생성
- 참조 횟수 (reference counter)
 - 모든 객체는 참조 횟수를 가짐
 - 객체 10의 참조 횟수는 2가 됨



메모리와 변수

- 변수 이름 vs 값 객체

- $a=10$
 - a: 변수 이름
 - 10: 값 객체
 - $b=a$
 - 이름만 하나 더 생성



- 참조 횟수 (reference counter)

- 모든 객체는 참조 횟수를 가짐
 - 객체 10의 참조 횟수는 2가 됨
 - “del a”를 실행시키면 참조 횟수는 1이 됨

메모리와 변수

- 참조 횟수 (reference counter)
 - “del b”를 실행시키면 참조 횟수는 0이 됨
 - 참조 횟수가 0인 객체의 의미?
 - 없음 → 메모리에서 제거됨
 - 이 작업을 쓰레기 수집 (garbage collection)이라 하며, 파이썬이 자동으로 수행함
 - del 명령을 사용할 일 적음
 - C/C++ 언어는 프로그래머가 일일이 신경써야 함
 - 하지 않으면 어떤 일이 발생할까?
 - » 메모리 누수 (memory leakage)

```
int* set = new int[100];  
  
delete [] set;
```

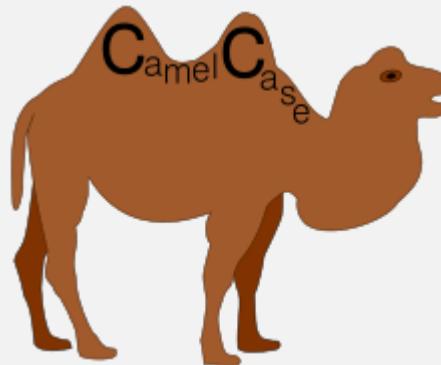


메모리와 변수

- 식별자 (identifier)
 - 파이썬에서 이름을 불일 때 사용하는 단어
 - 변수, 함수, 모듈, 클래스, 객체를 식별하는데 사용되는 이름
 - 문자, 숫자, '_'
 - 대문자와 소문자는 구별됨
 - teacher ≠ TEACHER
 - 시작은 문자와 '_'만 가능함
 - 4fun (X)
 - 올바른 예제
 - first_number = 15
 - student_name="John"
 - 잘못된 사례
 - 23answer
 - my name
 - your-answer

메모리와 변수

- 식별자 (identifier)
 - 스네이크 케이스(snake_case)
 - 단어 사이에 '_' 기호를 붙여서 만든 식별자
 - 주로 변수, 함수 이름으로 사용
 - 캐멀 케이스(CamelCase)
 - 단어들의 첫 글자를 대문자로 만든 식별자
 - 주로 클래스 이름으로 사용



https://en.wikipedia.org/wiki/Camel_case

메모리와 변수

- 식별자 (identifier)

사용 가능한 식별자	사용 불가능한 식별자
alpha alpha10 _alpha AlpHa ALPHA	break 273alpha has space special-character

스네이크 케이스	캐멀 케이스
item_list login_status character_hp rotate_angle	ItemList LoginStatus CharacterHp RotateAngle

키워드, 예약어

- Keyword, reserved word
 - 파이썬이 만들어질 때 정해 짐

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

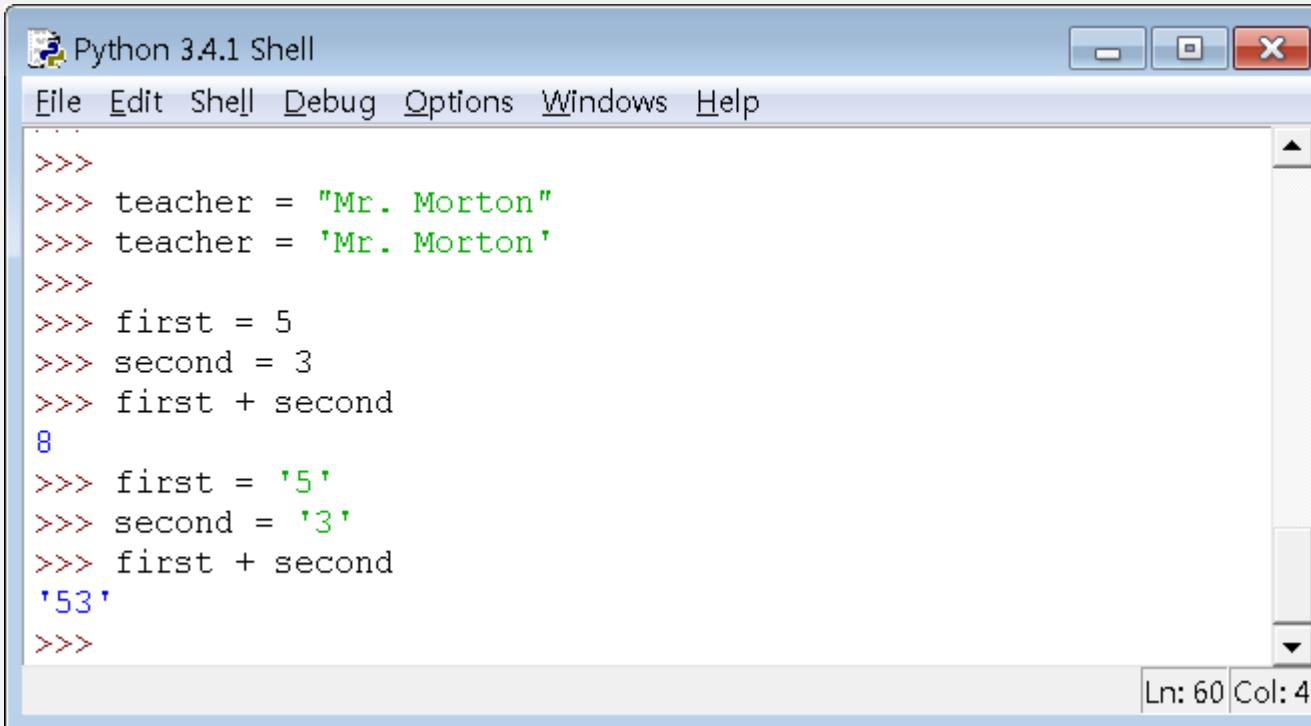
```
# 파이썬 키워드 확인하기
```

```
import keyword
```

```
print(keyword.kwlist)
```

문자열

- 숫자와 문자열
 - 문자열(string)은 문자나 문자들(문자, 숫자, 구두점)의 나열
 - 숫자 5와 문자열 “5” 차이 ← 따옴표
 - 문자열 생성은 ””와 ” 모두 사용 가능



The screenshot shows the Python 3.4.1 Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The code input area contains the following Python session:

```
>>>
>>> teacher = "Mr. Morton"
>>> teacher = 'Mr. Morton'
>>>
>>> first = 5
>>> second = 3
>>> first + second
8
>>> first = '5'
>>> second = '3'
>>> first + second
'53'
>>>
```

The status bar at the bottom right indicates Ln: 60 Col: 4.

문자열

- 긴 문자열
 - 한 줄이 넘는 문자열을 사용하려면 삼중 따옴표 문자열(triple quoted string)이라는 특수한 문자열을 사용함

The screenshot shows the Python 3.4.1 Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The code area displays the following examples:

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> first + second
'53'
>>> long_string = """Sing a song of sixpence, a pocket full of rye,
Four and twenty black birds baked in a pie."""
>>> long_string
'Sing a song of sixpence, a pocket full of rye,\nFour and twenty black birds baked in a pie.'
>>> long_string = '''Sing a song of sixpence, a pocket full of rye,
Four and twenty black birds baked in a pie.'''
>>> long_string
'Sing a song of sixpence, a pocket full of rye,\nFour and twenty black birds baked in a pie.'
>>>
```

The status bar at the bottom right indicates Ln: 73 Col: 4.

문자열

- 이스케이프 코드

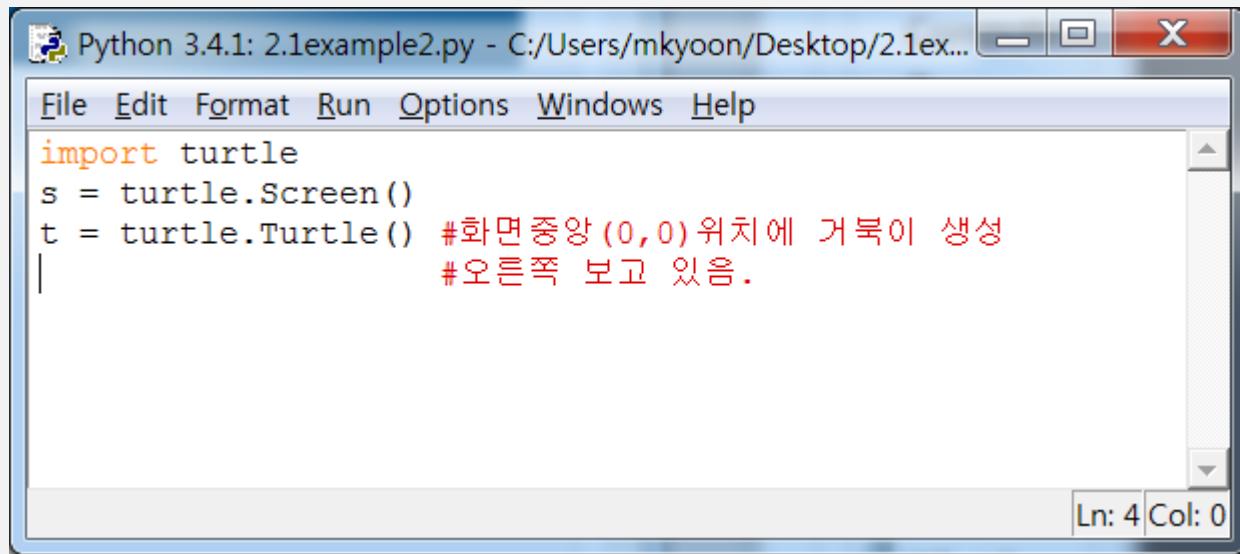
코드	설명
\n	개행 (줄바꿈)
\v	수직 탭
\t	수평 탭
\r	캐리지 리턴
\f	폼 피드
\a	벨 소리
\b	백 스페이스
\000	널문자
\\	문자 "\\"
'	단일 인용부호(')
"	이중 인용부호(")

실습

- 주변 친구들의 이름과 학번을 변수로 선언한다.
- 이름과 학번을 string concatenation시켜서 출력하시오. 단, 중간에 tab 문자를 집어넣는다.
- 다음 코드를 확인하시오.
 - >>> a = “python”
 - >>> print (a*3)

Turtle graphics

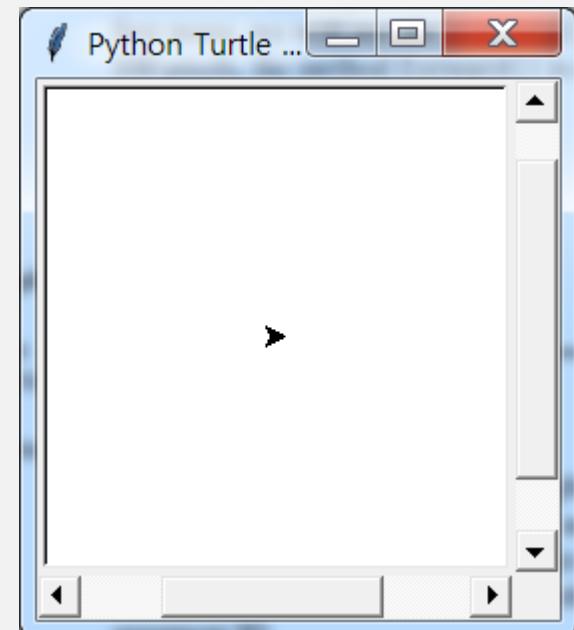
- 그래픽 예제
 - turtle 모듈
 - 거북이(들)이 펜 역할을 하면서 그림 그리기 가능
 - 반향전환, 전후좌우 직진, 원 그리기 등 가능



Python 3.4.1: 2.1example2.py - C:/Users/mkyoon/Desktop/2.1ex...

```
File Edit Format Run Options Windows Help
import turtle
s = turtle.Screen()
t = turtle.Turtle() #화면중앙(0,0) 위치에 거북이 생성
#오른쪽 보고 있음.
```

Ln: 4 Col: 0



Turtle graphics

- 그래픽 예제
 - turtle 모듈
 - 거북이(들)이 펜 역할을 하면서 그래픽 생성
 - 반향전환, 전후좌우 직진, 원 그리기 등 가능

The screenshot shows two windows side-by-side. The left window is a code editor titled "Python 3.4.1: 2.1example3.py - C:/Users/mkyoon/Desktop/2.1ex...". It contains the following Python code:

```
import turtle
s = turtle.Screen()
t = turtle.Turtle() #화면중앙 (0, 0) 위치에 거북이 생성
#오른쪽 향함.
t.forward(100) #100 픽셀 전진
t.left(90) #왼쪽으로 90도 회전. 위쪽 향함.
```

The right window is titled "Python Turtle Graphics" and displays a blank white canvas with a small black arrow cursor at the bottom center.

Turtle graphics

- 그래픽 예제
 - turtle 모듈
 - 거북이(들)이 펜 역할을 하면서 그래픽 생성
 - 반향전환, 전후좌우 직진, 원 그리기 등 가능

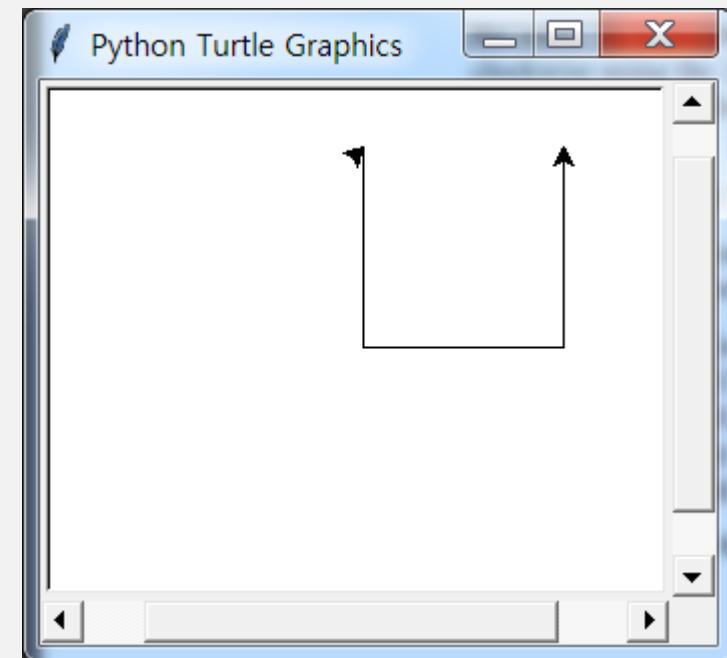
Python 3.4.1: 2.1example4.py - C:/Users/mkyoon/D...

```
File Edit Format Run Options Windows Help

import turtle
s = turtle.Screen()
t = turtle.Turtle() #화면중앙(0,0)위치에 거북이 생성
#오른쪽 향함.
t.forward(100) #100 픽셀 전진
t.left(90) #왼쪽으로 90도 회전. 위쪽 향함.

u = turtle.Turtle() # 두번째 거북이 생성
u.left(90) # 두번째 거북이 위쪽 향함.
u.forward(100) # 두번째 거북이 위로 100 픽셀 전진.
t.forward(100) # 첫번째 거북이 위로 100 픽셀 전진.
u.right(45) # 두번째 거북이 오른쪽으로 45도 회전.

Ln: 11 Col: 30
```



Turtle graphics

- 명령어 모음

Usage	Explanation
<code>t.forward(distance)</code>	Move turtle in the direction the turtle is headed by <code>distance</code> pixels
<code>t.left(angle)</code>	Rotate turtle counterclockwise by <code>angle</code> degrees
<code>t.right(angle)</code>	Rotate turtle clockwise by <code>angle</code> degrees
<code>t.undo()</code>	Undo the previous move
<code>t.goto(x, y)</code>	Move turtle to coordinates defined by <code>x</code> and <code>y</code> ; if pen is down, draw line
<code>t.setx(x)</code>	Set the turtle's first coordinate to <code>x</code>
<code>t.sety(y)</code>	Set the turtle's second coordinate to <code>y</code>
<code>t.setheading(angle)</code>	Set orientation of turtle to <code>angle</code> , given in degrees; Angle 0 means east, 90 is north, and so on
<code>t.circle(radius)</code>	Draw a circle with given <code>radius</code> ; the center of the circle is <code>radius</code> pixels to the left of the turtle
<code>t.circle(radius, angle)</code>	Draw only the part the circle (see above) corresponding to <code>angle</code>
<code>t.dot(diameter, color)</code>	Draw a dot with given <code>diameter</code> and <code>color</code>
<code>t.penup()</code>	Pull pen up; no drawing when moving
<code>t.pendown()</code>	Put pen down; drawing when moving
<code>t.pensize(width)</code>	Set the pen line thickness to <code>width</code>
<code>t.pencolor(color)</code>	Set the pen color to <code>color</code> described by string <code>color</code>

Turtle graphics

- 원그리기

The image shows a Windows desktop environment. On the left is a code editor window titled "Python 3.4.1: 2.1example5.py - C:/Users/mkyoon/D...". The code is written in Python using the turtle module to draw a circle:

```
import turtle
s = turtle.Screen()
t = turtle.Turtle()
#t = turtle.Turtle(shape='turtle')
t.pensize(3) #펜 진하게

x = -100
y = 100
t.goto(x,y) # 좌표로 선그리면서 이동
t.undo() #취소

t.penup() #그리지 않고 좌표 이동
t.goto(x, y)
t.pendown() #그릴 준비

t.dot()
t.circle(100) #반지름 100짜리 원그리기.
#거북이가 바라보는 방향의 왼쪽이 원중심.
```

The status bar at the bottom of the code editor shows "Ln: 17 Col: 0". To the right of the code editor is a separate window titled "Python Turtle Graphics" which displays a single black circle centered on the screen.

Turtle graphics

```
Python 3.4.1: 2.1example6.py - C:/Users/mkyoon/De...
File Edit Format Run Options Windows Help
import turtle
s = turtle.Screen()
t = turtle.Turtle()
t.pensize(3) #펜 진하게

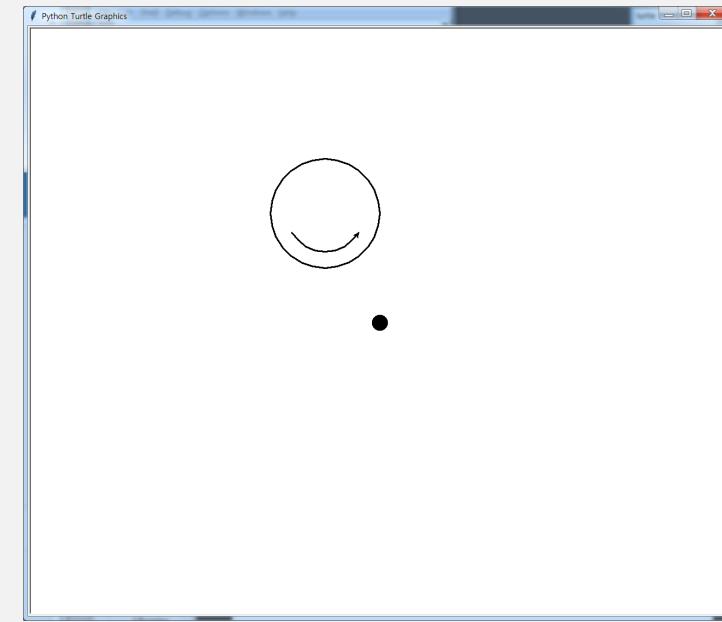
t.dot(30) #점 그리기

x = -100
y = 100
t.penup()
t.goto(x,y)
t.pendown()

t.circle(100) #반지름 100짜리 원그리기.
#거북이가 바라보는 방향의 왼쪽이 원중심.

t.penup()
t.goto(x-60.62, y+65)
t.pendown()
t.setheading(-60)
t.circle(70, 120)

Ln: 22 Col: 0
```



숙제

- 스마일 그림 그리기

