

# Memory & Variables

By 윤명근 / 박수현

# 수업목표

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

# 입력(input)

- Computer 입력
  - Keyboard
    - 표준 입력 장치: standard input
  - Touch screen
  - File
  - Voice recognition
  - Fingerprint recognition
  - Iris recognition(홍채인식)
  - Facial recognition ...
- Smartphone (mobile device) 입력
  - keypad
  - Touch screen
  - File
  - NFC(Near Field Communication)
  - BLE (Bluetooth Low Energy)
  - Voice recognition
  - Facial recognition ...

# 출력(output)

- computer(PC) 출력
  - Monitor
    - 표준 출력 장치: standard output
  - Printer
  - File (pdf, jpeg, png..)
  - Voice
  - ...
- Smartphone 출력
  - Display
  - Voice
  - File
  - NFC
  - ...

# 입력과 출력

- Keyboard 입력, Monitor 출력

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

# 메모리와 변수

- “Teacher”는 어디에 저장되나? → Computer Memory

데스크탑용  
램 모듈

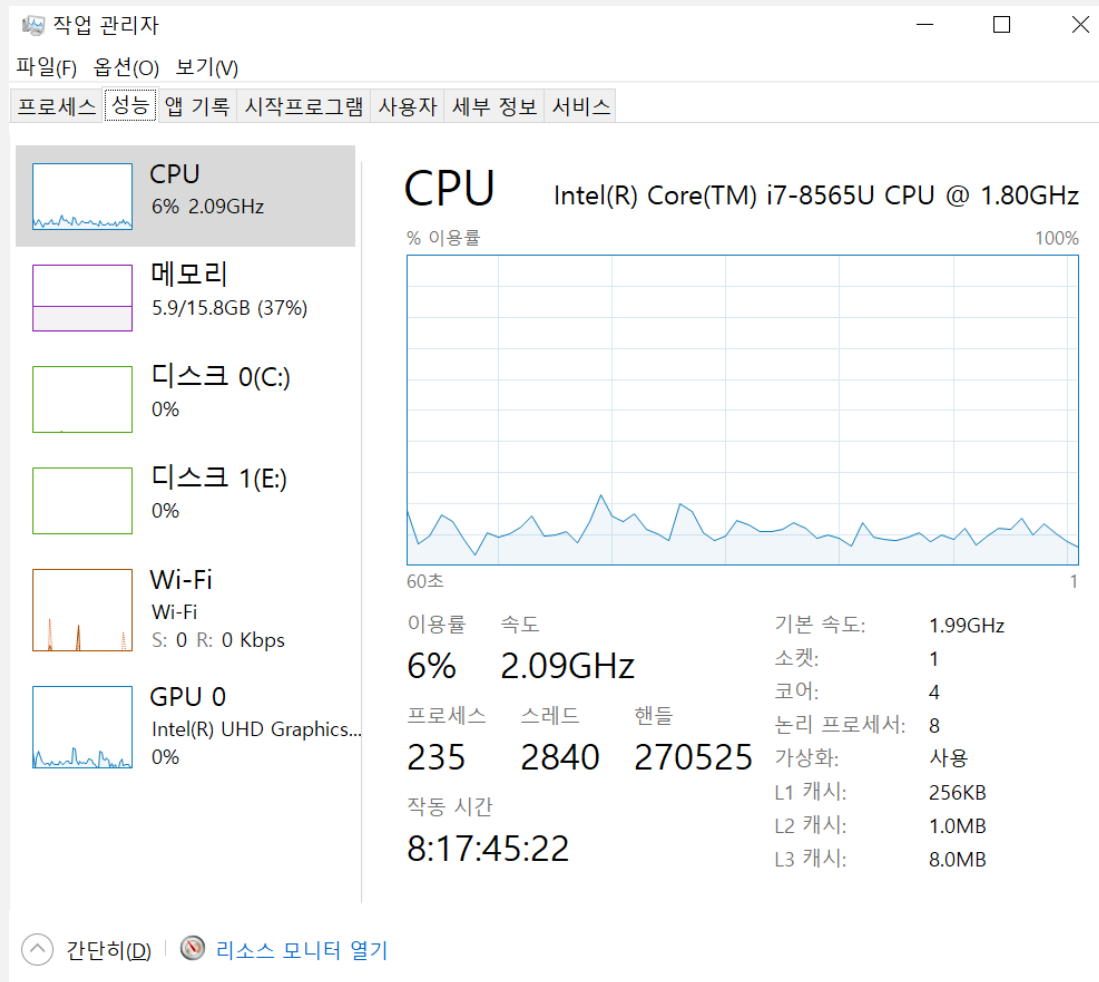


노트북용  
램 모듈



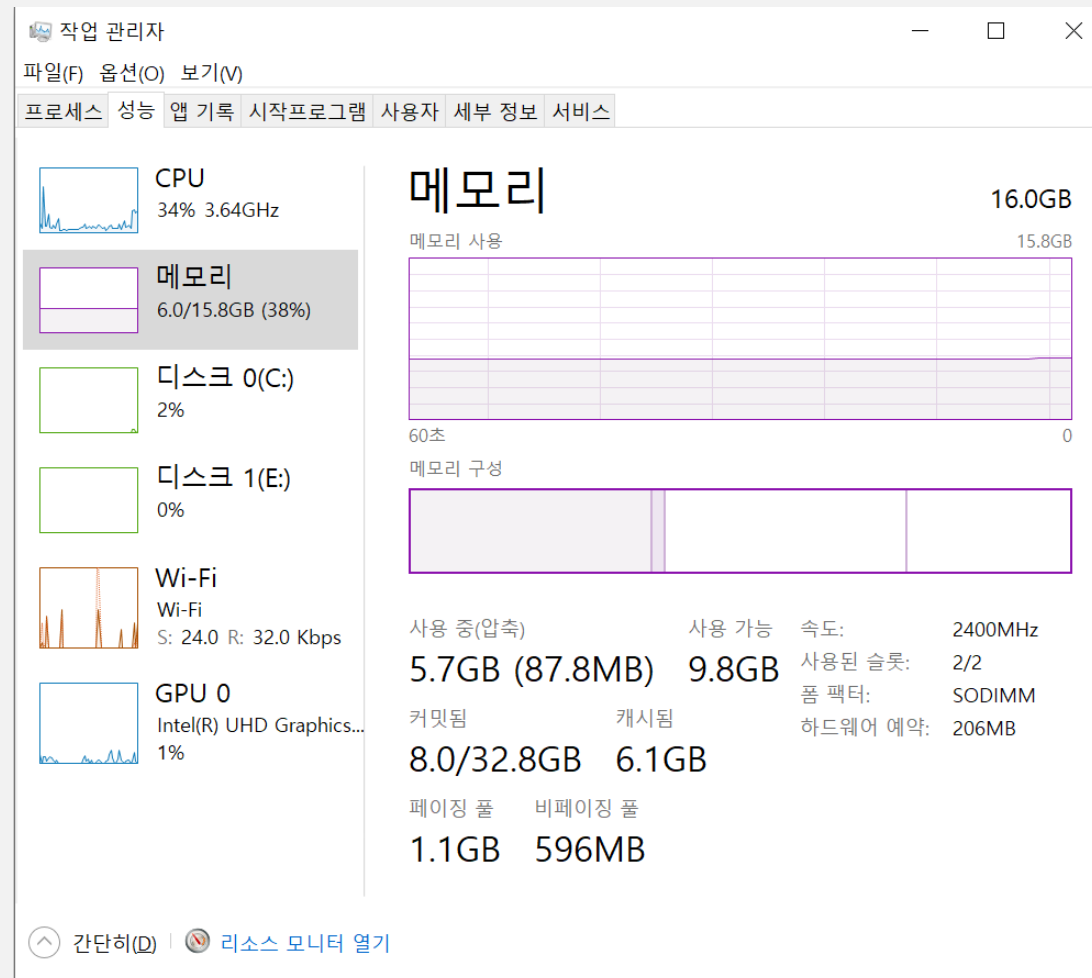
# 메모리와 변수

- Windows 'Ctrl' + 'Shift' + 'Esc'



# 메모리와 변수

- 윈도우 'Ctrl' + 'Shift' + 'Esc'

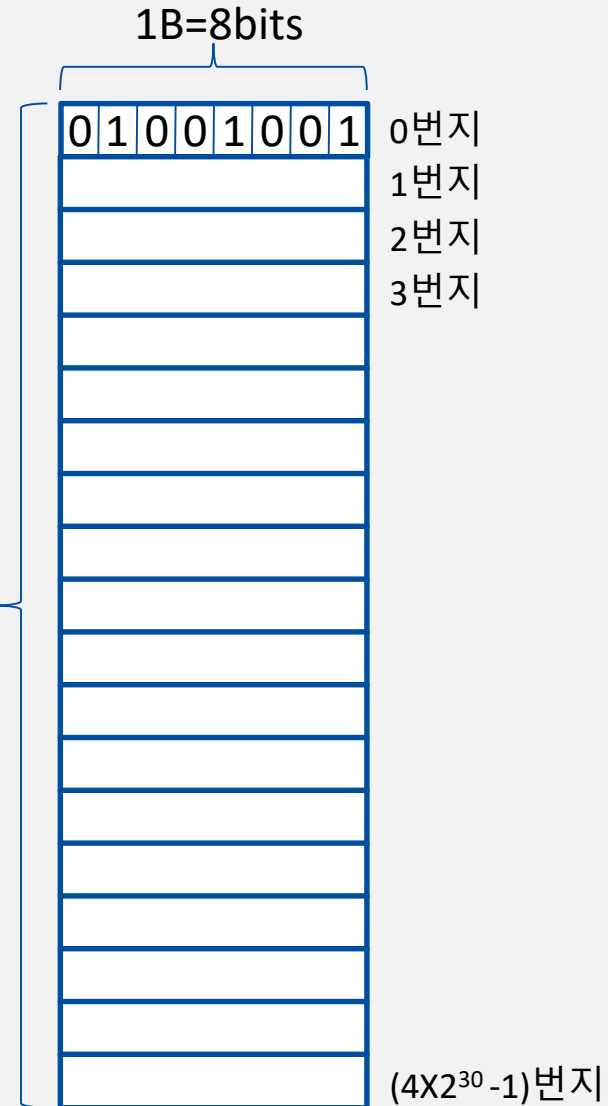




# 메모리와 변수

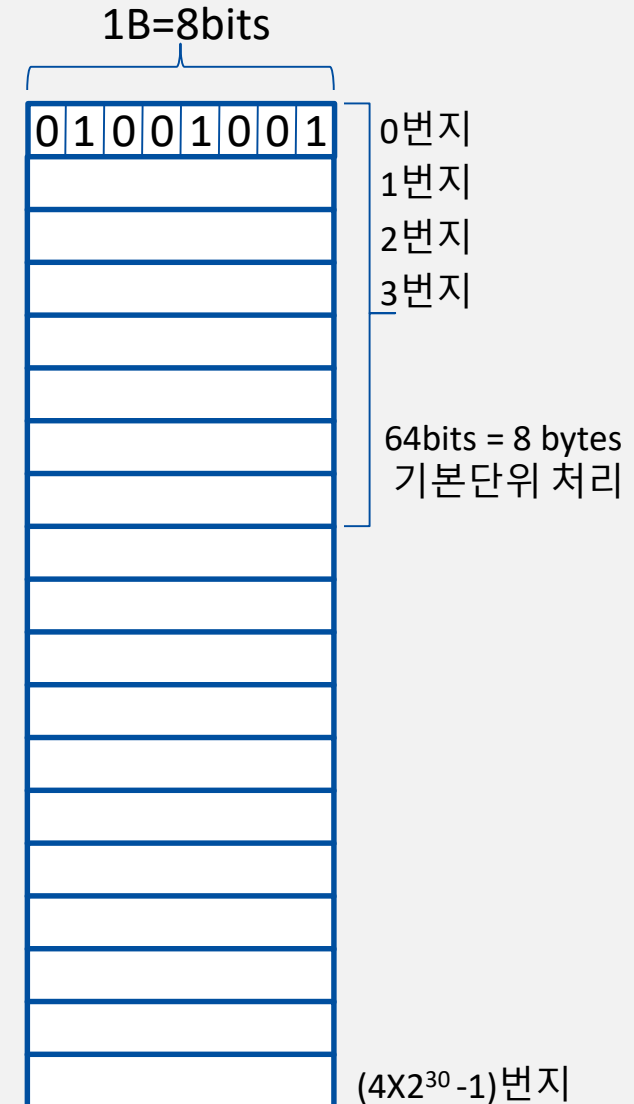
- 메모리 용량 16 GB (Gigabyte)
  - 1 Byte(=8 bits)가 16 G 개 만큼 존재
    - Computer 기본 단위는 bit
      - 0과 1 표현
    - Kilo :  $10^3 = 2^{10} = 1,024$   
Mega :  $10^6 = 2^{20} = 1,048,576$   
Giga :  $10^9 = 2^{30} = 1,073,741,824$
  - $16 \text{ GB} = 16 \times 10^9 \text{ Bytes} = 16 \times 2^{30} \text{ Bytes}$

16 G개 =  $16 \times 2^{30}$ 개



# 메모리와 변수

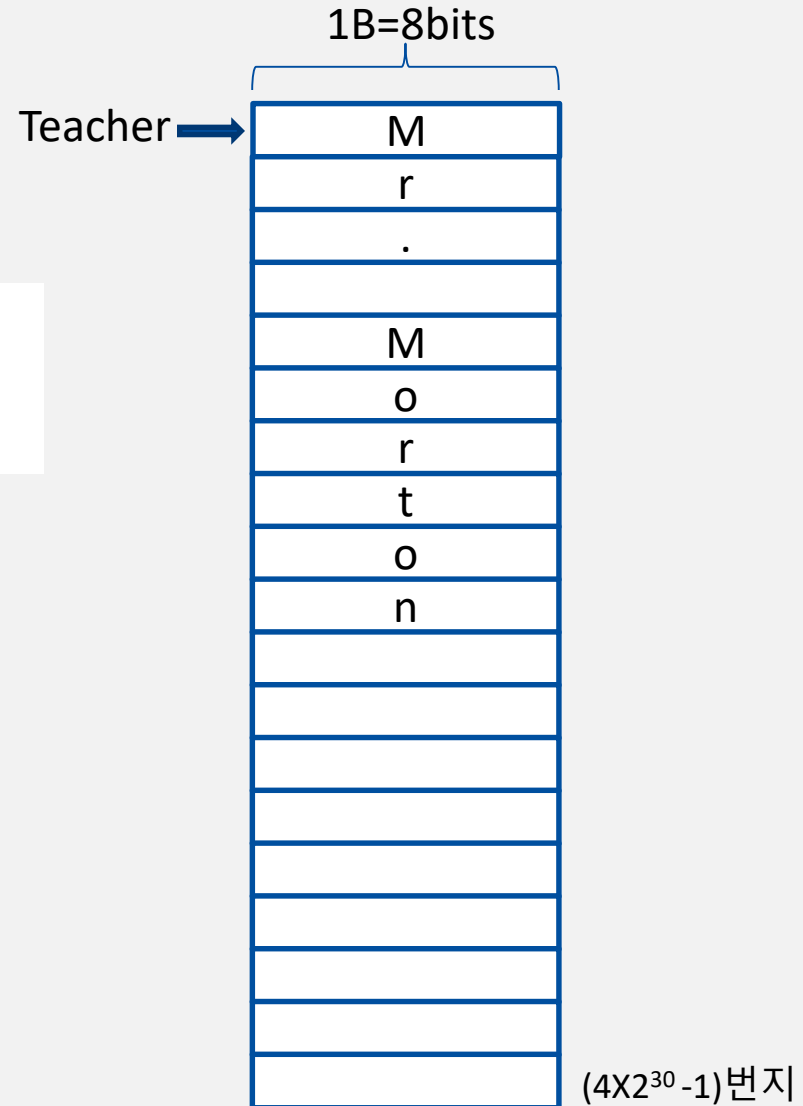
- 메모리용량 16 GB
  - Windows 운영체제 64bit vs 32bit
    - 기본 처리 단위가 8 Byte 또는 4 Byte의 차이



# 메모리와 변수

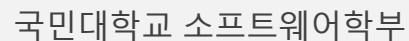
- 변수 "Teacher"

```
>>>  
>>> Teacher = "Mr . Morton"  
>>> |
```



- 변수 “Teacher”

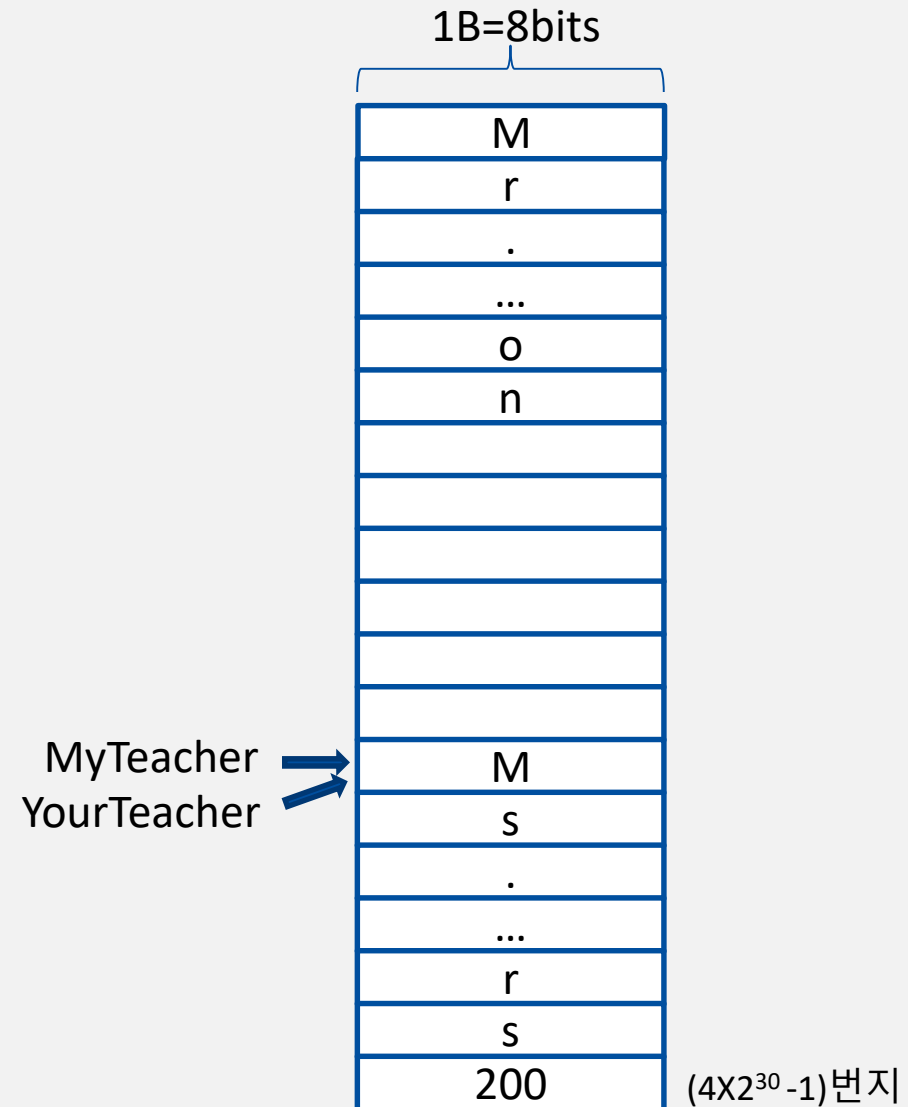
- 변수 “X”



# 메모리와 변수

- 변수 “MyTeacher”
- 변수 “YourTeacher”

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



# 메모리와 변수

- id() 함수
  - 객체의 id값 리턴

```
2.1example1.py
File Edit Format Run Options Window 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))
```

```
Teacher id= 54832248
Teacher id= 64034480
MyTeacher id= 64034480
YourTeacher id= 64034480
>>>
```

# 메모리와 변수

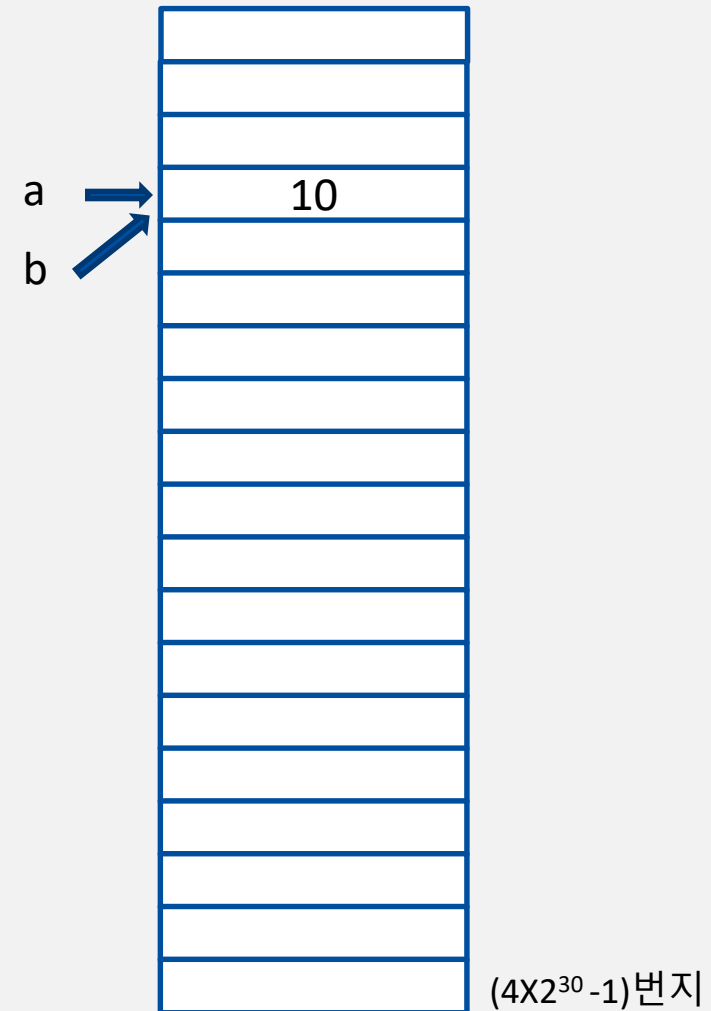
- id() 함수
  - 자주 사용하는 객체는 Python이 미리 생성해 놓고 있음
    - 예) 숫자 -5 ~ +256
  - id(p)에서 에러 발생 이유는?
  - PC마다(또는 부팅 시마다) id()값이 다른 이유는?

```
>>>
>>> id(1)
1505780912
>>> id(2)
1505780928
>>> id(100)
1505782496
>>> id(135)
1505783056
>>> id(p)
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    id(p)
NameError: name 'p' is not defined
>>> p = "abcd"
>>> id(p)
64019008
>>>
```

```
>>>
>>> id(1)
1584920496
>>> id(100)
1584922080
>>> id(135)
1584922640
>>> p = "abcd"
>>> id(p)
61914144
>>>
```

# 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





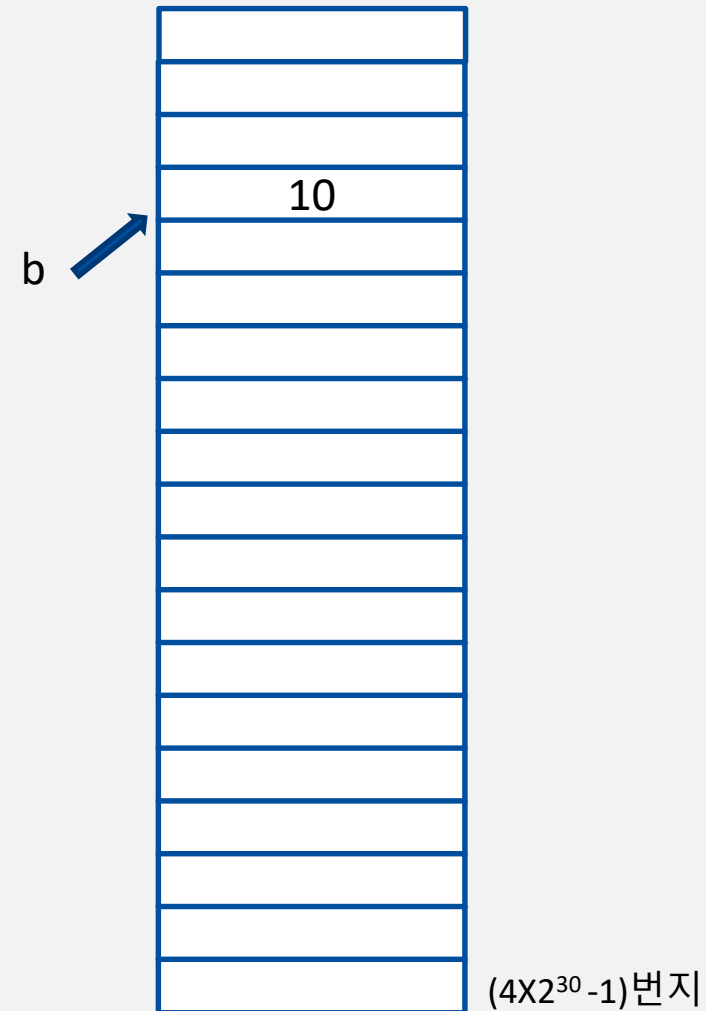
# 메모리와 변수

- 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, Python이 자동으로 수행함
  - del 명령을 사용할 일 적음
    - C/C++ 언어는 프로그래머가 일일이 신경 써야함
    - 하지 않으면 어떤 일이 발생할까?
      - » 메모리 누수 (memory leakage)

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



( $4 \times 2^{30} - 1$ )번지

# 메모리와 변수

- Identifier (식별자) : (변수(variable) 이름)
- Naming convention
  - 문자, 숫자, '\_'
  - 대문자와 소문자는 구별됨
    - teacher ≠ TEACHER
  - 시작은 문자와 '\_'만 가능함
    - 4fun (X), \_fun\_exam (o)
  - 올바른 예제
    - first\_number = 15
    - student\_name = "John"
  - 잘못된 사례
    - 23answer
    - my name
    - your-answer

# 문자열(String)

- 숫자와 문자열

- 문자열(string)은 문자(character)나 문자들(문자, 숫자, 구두점)의 나열
- 숫자 5와 문자열 "5" 차이 ← 따옴표
- 문자열 생성은 " " 와 ' ' 모두 사용 가능

```
>>>
>>> first = 5
>>> second = 3
>>> first + second
8
>>> first = '5'
>>> second = '3'
>>> first + second
'53'
>>>
```

# 문자열

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

```
>>> long_string = """Perhaps love is like a resting place,
A shelter from the storm
It exists to give you comfort
It is there to keep you warm
And in those times of trouble
When you are most alone, the memory of love will bring you home"""
>>>
>>> long_string
'Perhaps love is like a resting place.\nA shelter from the storm.\nIt exists to give\nyou comfort.\nIt is there to keep you warm.\nAnd in those times of trouble.\nWhen you\nare most alone, the memory of love will bring you home'
>>>
>>> print(long_string)
Perhaps love is like a resting place,
A shelter from the storm
It exists to give you comfort
It is there to keep you warm
And in those times of trouble
When you are most alone, the memory of love will bring you home
>>>
>>> long_string2 = """Perhaps love is like a resting place
A shelter from the storm
It exists to give you comfort
It is there to keep you warm
And in those times of trouble
When you are most alone
The memory of love will bring you home"""
>>> print(long_string2)
Perhaps love is like a resting place
A shelter from the storm
It exists to give you comfort
It is there to keep you warm
And in those times of trouble
When you are most alone
The memory of love will bring you home
```

# 문자열

- 이스케이프 코드 (escape code)

코드	설명
\n	개행 (줄바꿈)
\v	수직 탭(tab)
\t	수평 탭
\r	캐리지 리턴(carriage return)
\f	폼 피드(form feed)
\a	벨 소리(bell)
\b	백 스페이스(back space)
\000	널 문자(null character)
\\	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)`

# 실습



f\_name\_num.py

File Edit Format Run Options Window Help

```
friend_name = "김국민"  
friend_num = "20xx1234"
```

```
f_name_num = friend_name + friend_num  
print()  
print("1) 친구이름과 학번 = ", f_name_num)
```

```
f_name_num = friend_name + "\t" + friend_num  
print("2) \t 추가 후 친구이름과 학번 = ", f_name_num)  
print()
```

```
#또 다른 예제  
a = "Python "  
print(a*3)
```

1) 친구이름과 학번 = 김국민20xx1234

2) \t 추가 후 친구이름과 학번 = 김국민

20xx1234

Python Python Python

>>> |



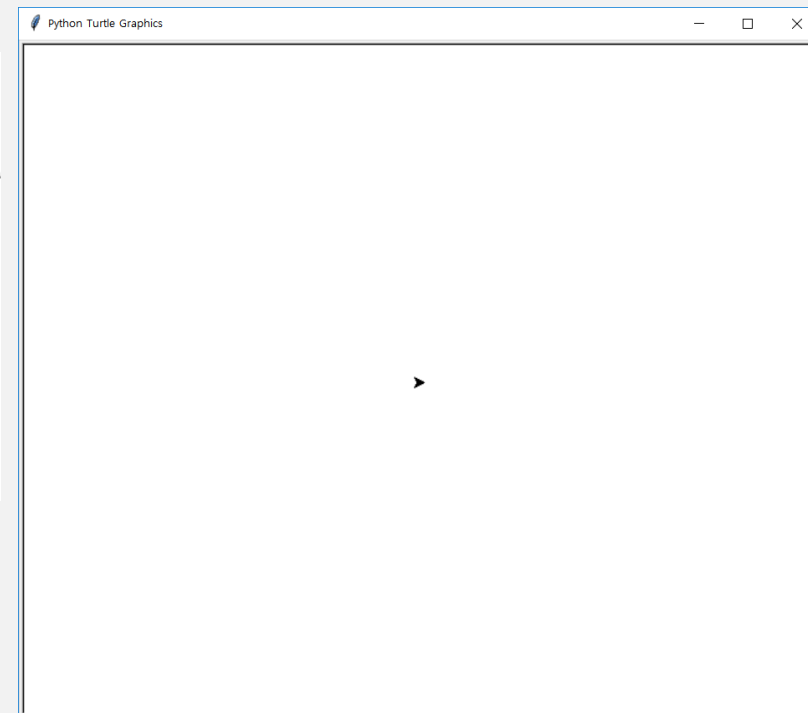
# Turtle graphics

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

```
2.1example2.py
File Edit Format Run Options Window Help

import turtle

s = turtle.Screen()
t = turtle.Turtle() #화면중앙(0,0)위치에 거북이 생성
                    #오른쪽을 보고 있음.
```



# Turtle graphics

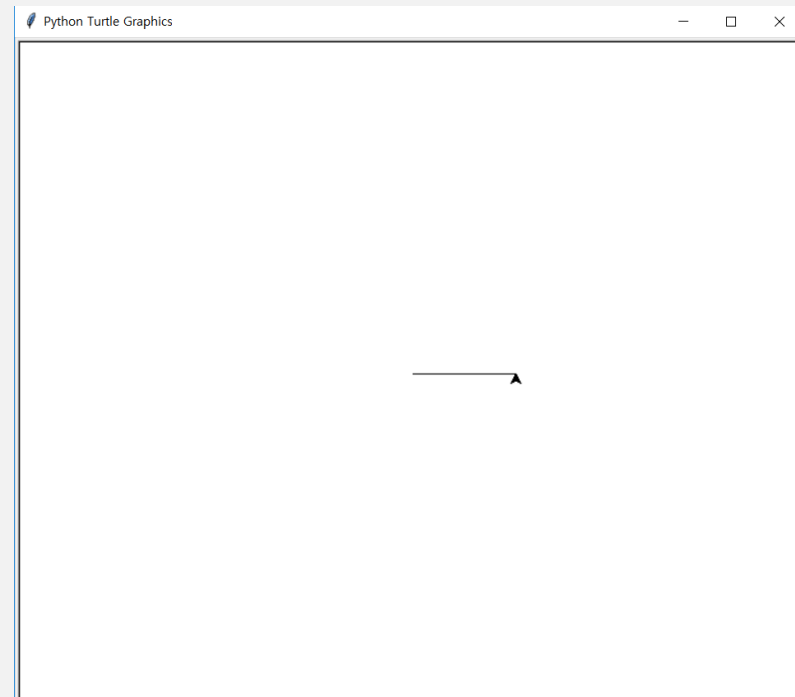
- 그래픽 예제
  - turtle 모듈
  - 거북이(들)이 펜 역할을 하면서 그래픽 생성
  - 방향전환, 전후좌우 직진, 원 그리기 등 가능

```
*2.1example3.py
File Edit Format Run Options Window Help

import turtle

s = turtle.Screen()
t = turtle.Turtle() #화면중앙(0,0)위치에 거북이 생성
                    #오른쪽을 향함.

t.forward(100)      #100 픽셀 전진
t.left(90)           #왼쪽으로 90도 회전. 위쪽 향함.
```



# Turtle graphics

```
2.1example4.py
File Edit Format Run Options Window Help

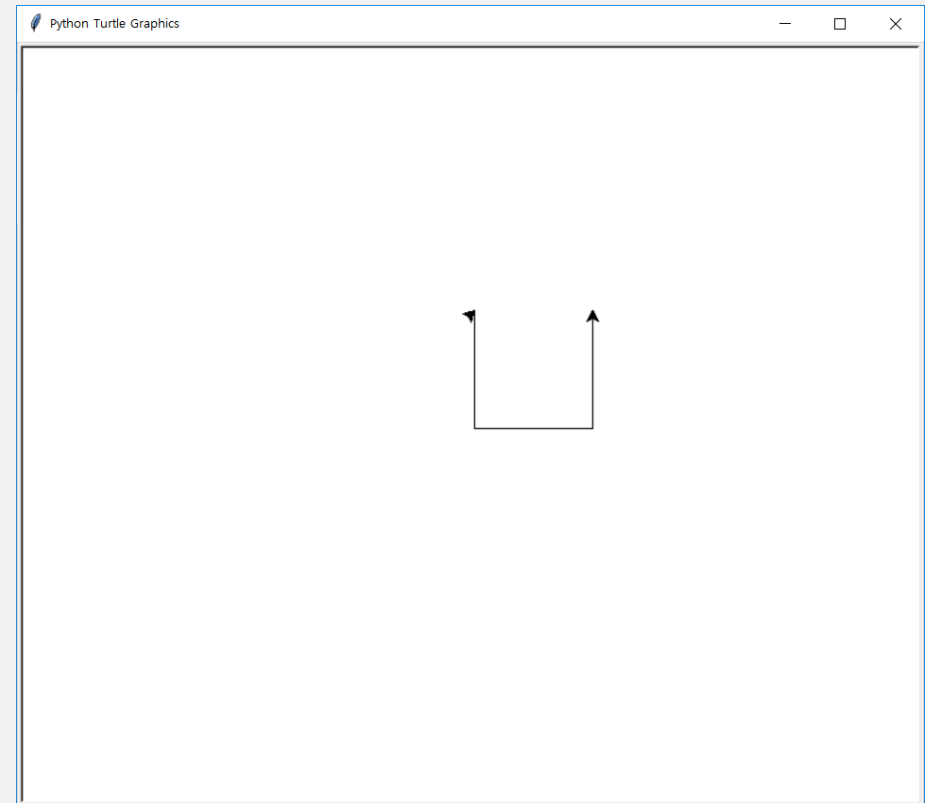
import turtle

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.
```



# Turtle graphics

- 명령어 모음

Usage	Explanation
<code>t.forward(distance)</code>	Move turtle in the direction the turtle is headed by distance pixels
<code>t.left(angle)</code>	Rotate turtle counterclockwise by angle degrees
<code>t.right(angle)</code>	Rotate turtle clockwise by angle degrees
<code>t.undo()</code>	Undo the previous move
<code>t.goto(x, y)</code>	Move turtle to coordinates defined by x and y; if pen is down, draw line
<code>t.setx(x)</code>	Set the turtle's first coordinate to x
<code>t.sety(y)</code>	Set the turtle's second coordinate to y
<code>t.setheading(angle)</code>	Set orientation of turtle to angle, given in degrees; Angle 0 means east, 90 is north, and so on
<code>t.circle(radius)</code>	Draw a circle with given radius; the center of the circle is radius pixels to the left of the turtle
<code>t.circle(radius, angle)</code>	Draw only the part the circle (see above) corresponding to angle
<code>t.dot(diameter, color)</code>	Draw a dot with given diameter and color
<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 width
<code>t.pencolor(color)</code>	Set the pen color to color described by string color

# Turtle

2.1example5.py

File Edit Format Run Options Window Help

```
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)
```

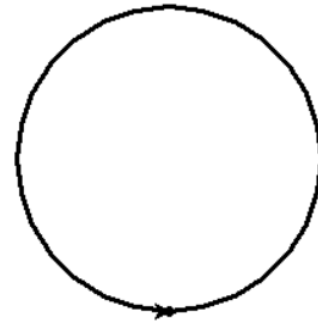
```
t.pendown() #그릴 준비
```

```
t.dot()
```

```
t.circle(100) #반지름 100짜리 원그리기.
```

```
#거북이가 바라보는 방향의 왼쪽이 원중심.
```

Python Turtle Graphics



- 원그리기

# Turtle

2.1example5.py

File Edit Format Run Options Window Help

```
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
```

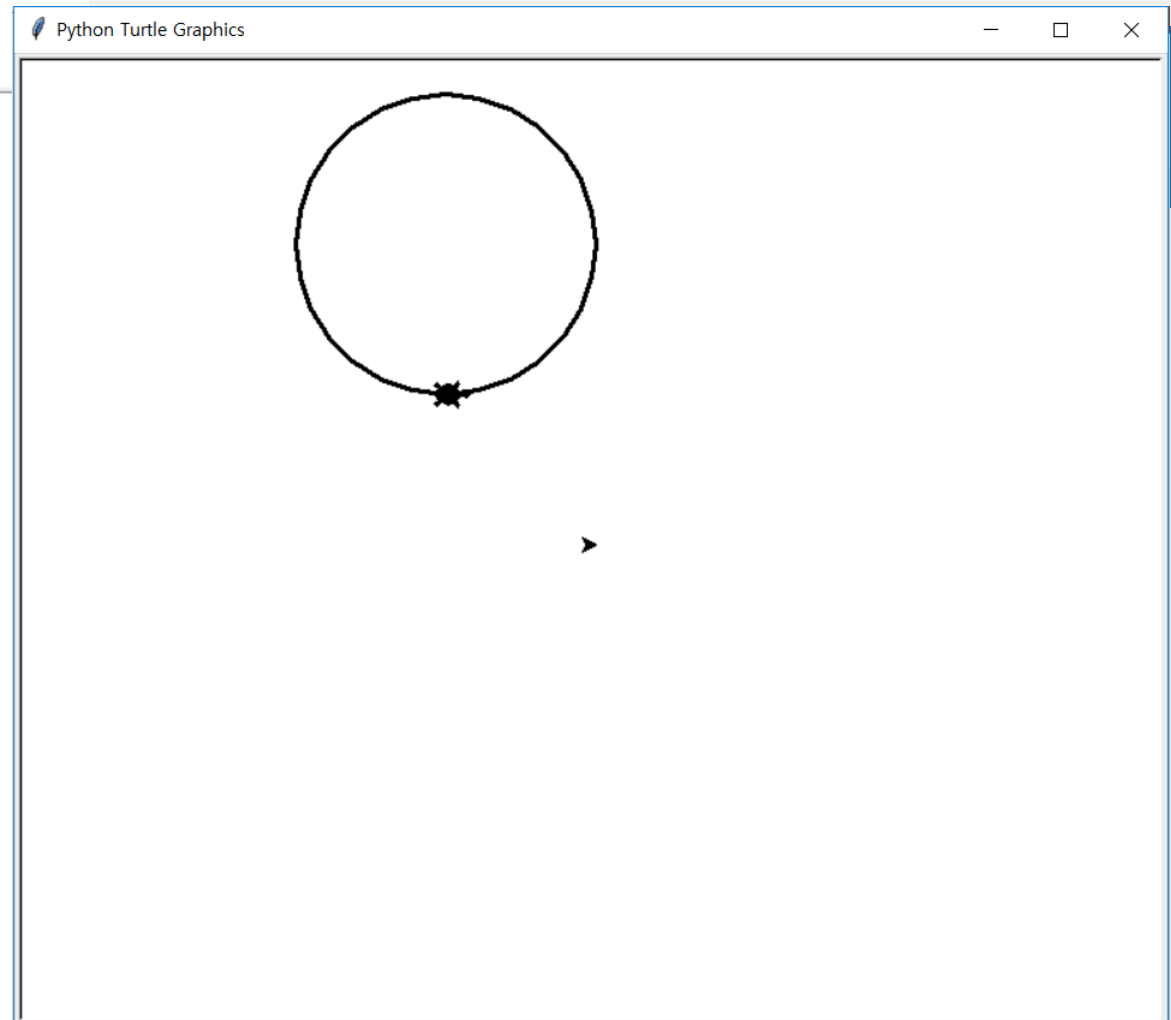
```
t.goto(x, y)
```

```
t.pendown() #그릴 준비
```

```
t.dot()
```

```
t.circle(100) #반지름 100짜리 원그리기.
```

```
#거북이가 바라보는 방향의 왼쪽이 원중심.
```



- 원그리기

# Turtle

```
2.1example6.py
File Edit Format Run Options Window Help

import turtle

s = turtle.Screen()
t = turtle.Turtle()

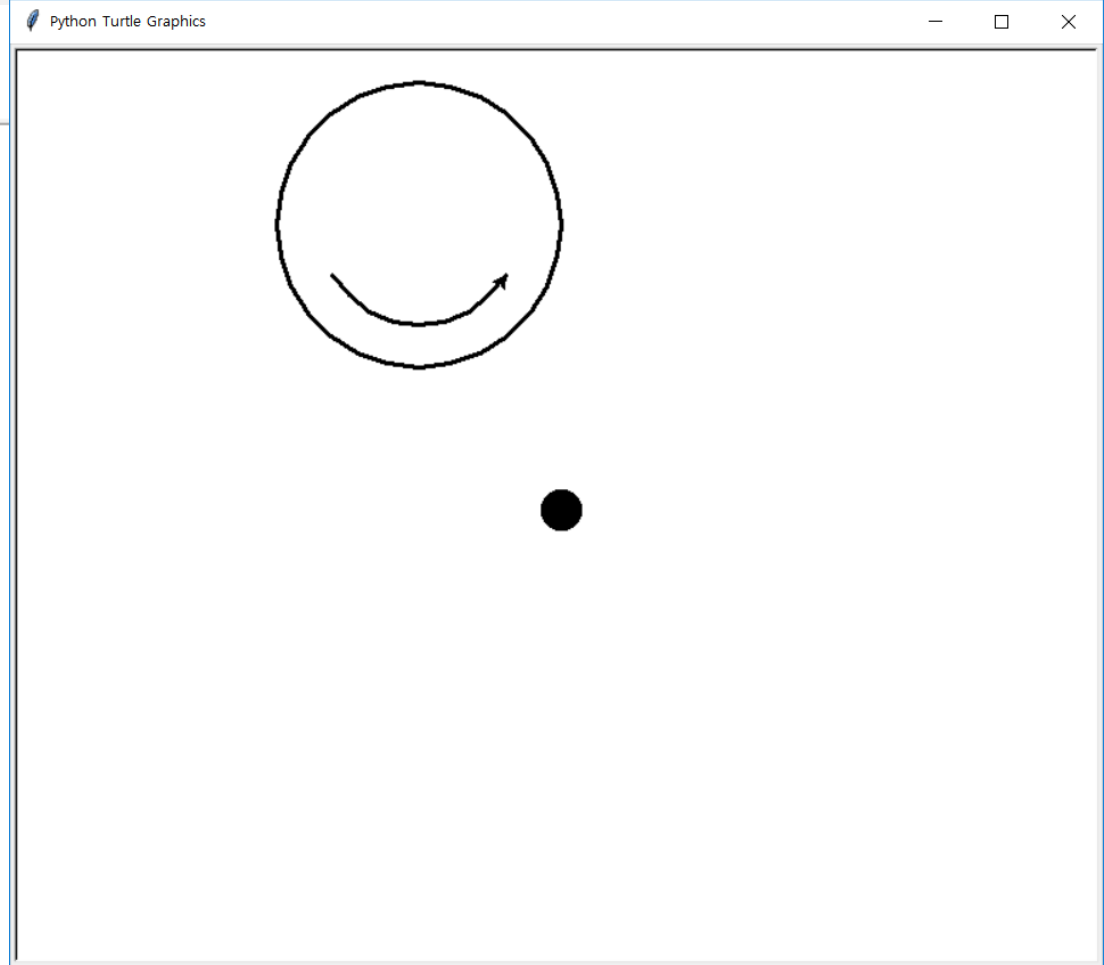
t.pensize(3) #펜 진하게

t.dot(30) #점그리기

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)
```



# 숙제

- 스마일 그림 그리기





# Homework

- 다음 수업시작 전까지 ecampus로 upload

- 제출방법

파일명 : 과소사-학번-이름-chapter번호

예) 과소사-20201234-김국민-2.1장

- 파일이 여러 개일 경우 zip으로 묶어서 제출