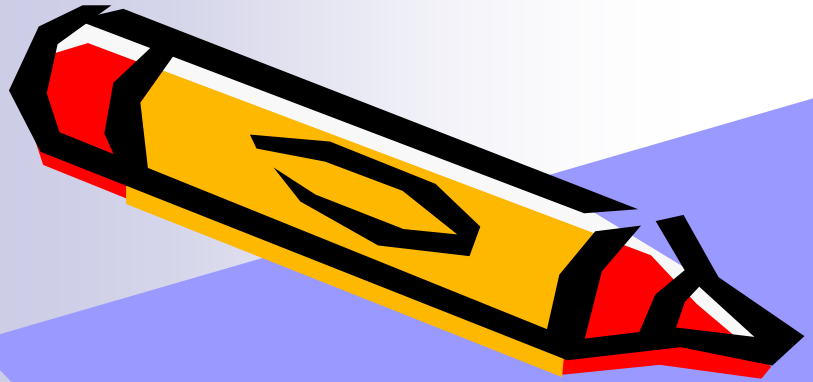
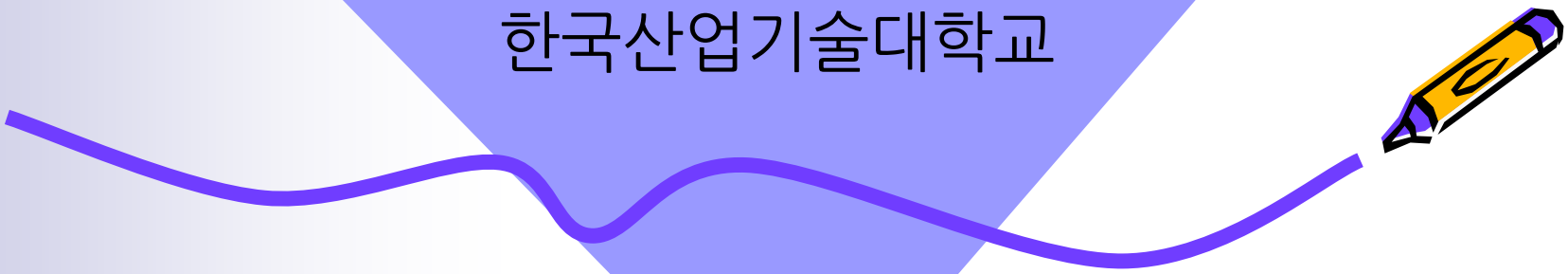


2D 게임 프로그래밍

# 제1강 2D 렌더링 기초

이대현
한국산업기술대학교



# 학습 내용

- 2D 게임의 기본 요소
- 필요 Tool 들의 설치
- 캐릭터 이미지의 렌더링과 이동

# 2D 게임?

- 게임이란?
  - “가상 월드에 존재하는 여러 객체들의 상호작용”

- 게임의 기본 구성 요소
  - 배경
  - 캐릭터, 오브젝트
  - UI - GUI, 입력(키,마우스,터치, …)
  - AI
  - 사운드

- 2D 게임?
  - 현재 진행 중인 게임 가상 월드의 내용을 화면에 2D 그림으로 보여주는 것
  - 배경,캐릭터(오브젝트)의 표현(렌더링)을 2D 이미지들의 조합으로 구성함!

# 2D 게임의 기본 요소

# 2D 게임 개발 접근법

- 플랫폼 종속적 방법
  - Direct X
  - OpenGL
  - Simple Frame Buffer

- 플랫폼 독립적 방법, Cross Platform
  - Unity3D
  - COCOS2D
  - SDL
  - 그 외의 범용 2D 렌더링 라이브러리

# SDL(Simple DirectMedia Layer)

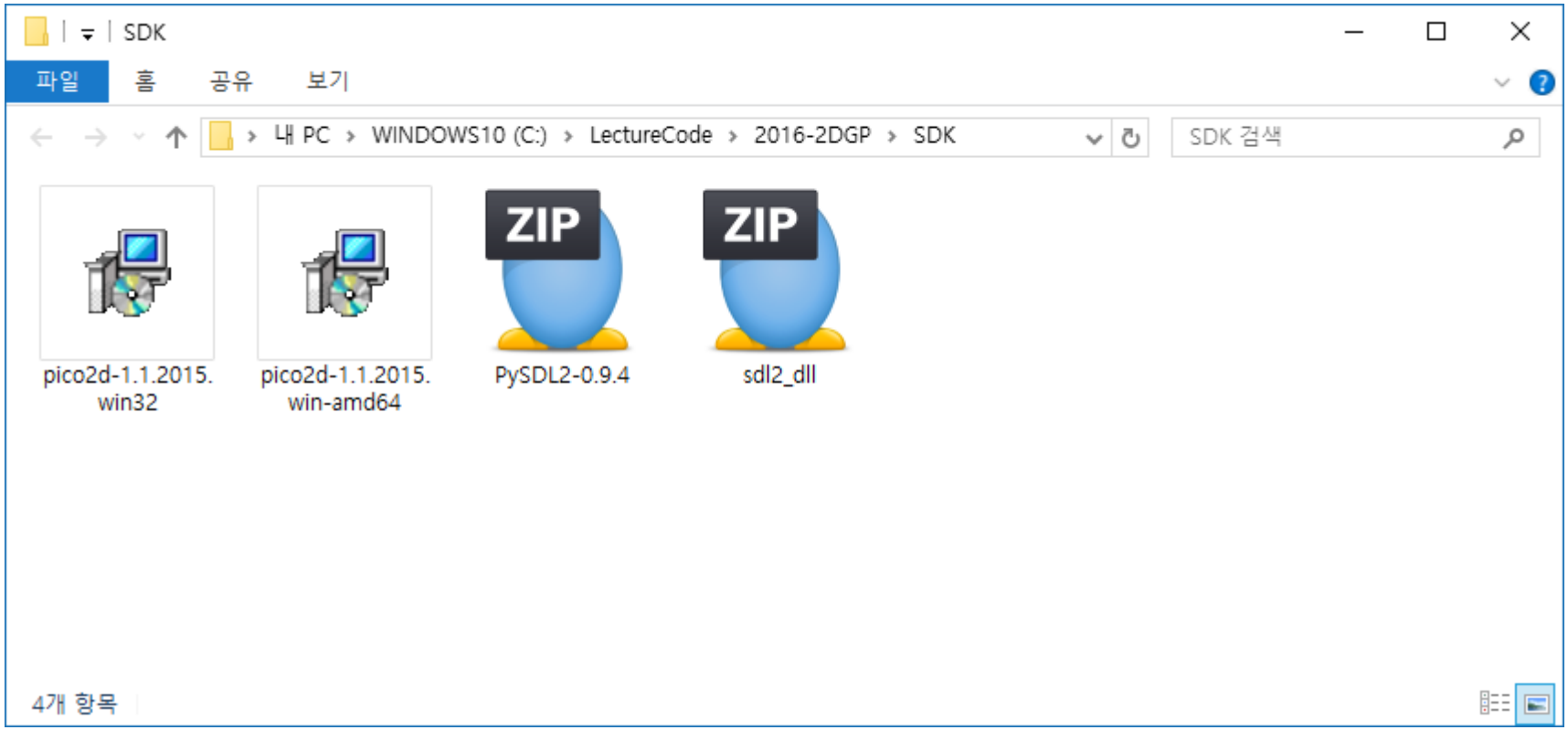
- SDL이란?
  - 크로스 플랫폼 멀티미디어 라이브러리.
  - 비디오, 오디오  및 사용자 입력을 처리하는 API로 구성.
  - 기본적으로 2D 그래픽 라이브러리. 3D는 OpenGL을 통해서 지원.

- SDL이 지원하는 플랫폼
  - PC: Windows, Linux, Mac OS
  - Phone: Android, iOS,


- 라이센싱(SDL 2.0)
  - zlib license
  - 자유롭게 상용 게임을 개발할 수 있슴.
  - SDL1.2 → GNU LGPL 라이센싱

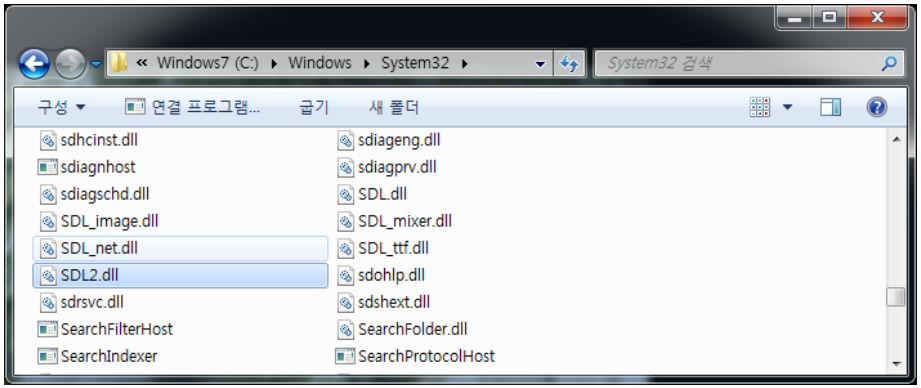
- 홈페이지
  - www.libsdl.org

# git pull



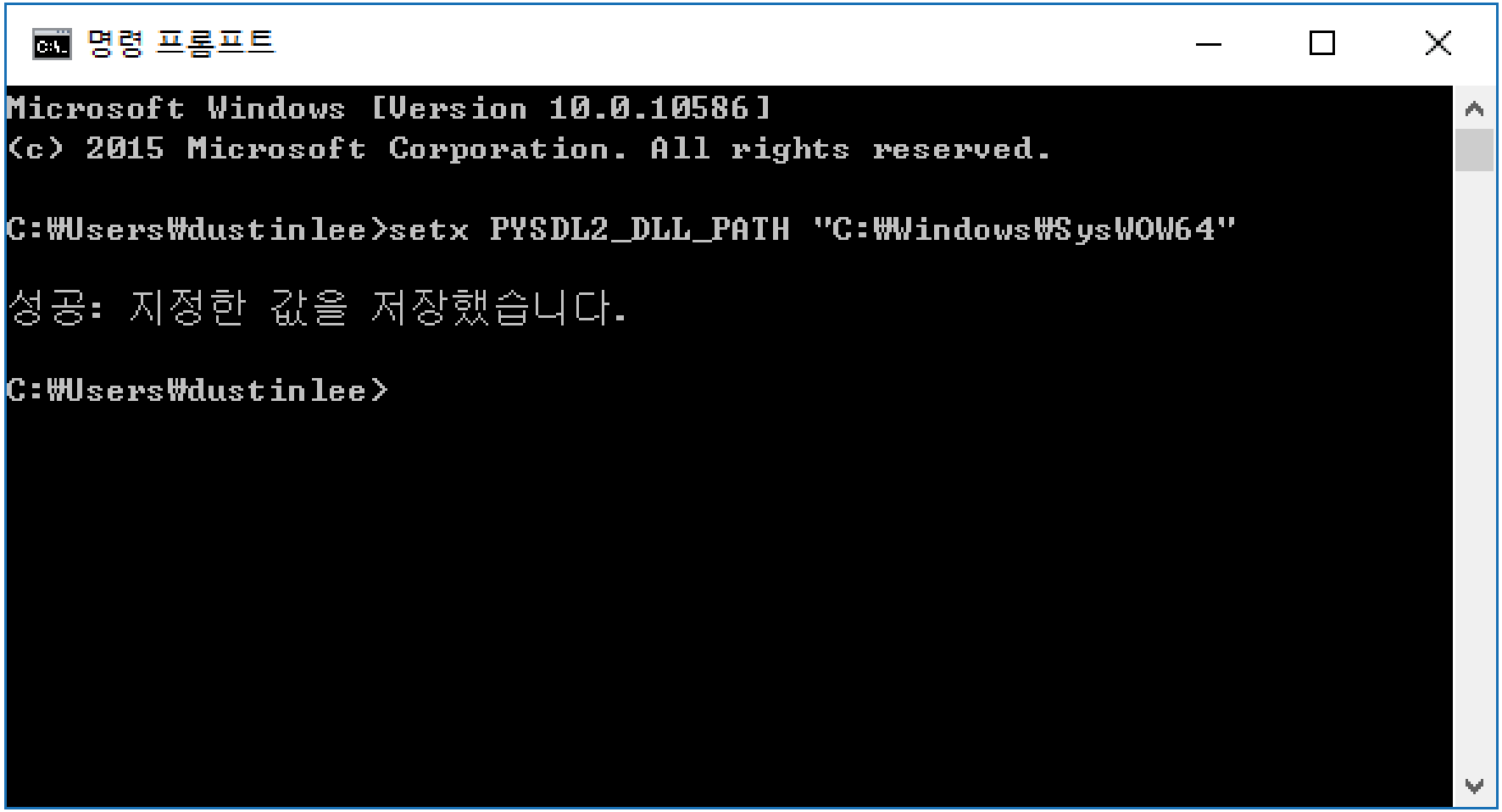
# SDL 라이브러리 DLL 등록/복사

- SDL 의 DLL 파일들을 윈도우에 등록
  - 32bit로 개발하고자 하는 경우, sdl2_dll/x86/ 폴더 안의 모든.dll 파일들을 C:₩Windows₩System32 폴더에 복사.
  - 64bit로 개발하고자 하는 경우, sdl2_dll/x64/ 폴더안의 모든.dll 파일들을 C:₩Windows₩SysWOW64 폴더에 복사
  - 64bit PC에서, 32bit python으로 32bit app을 개발하려면, SysWOW64 폴더안에, 32bit SDL DLL을 복사해야 함…

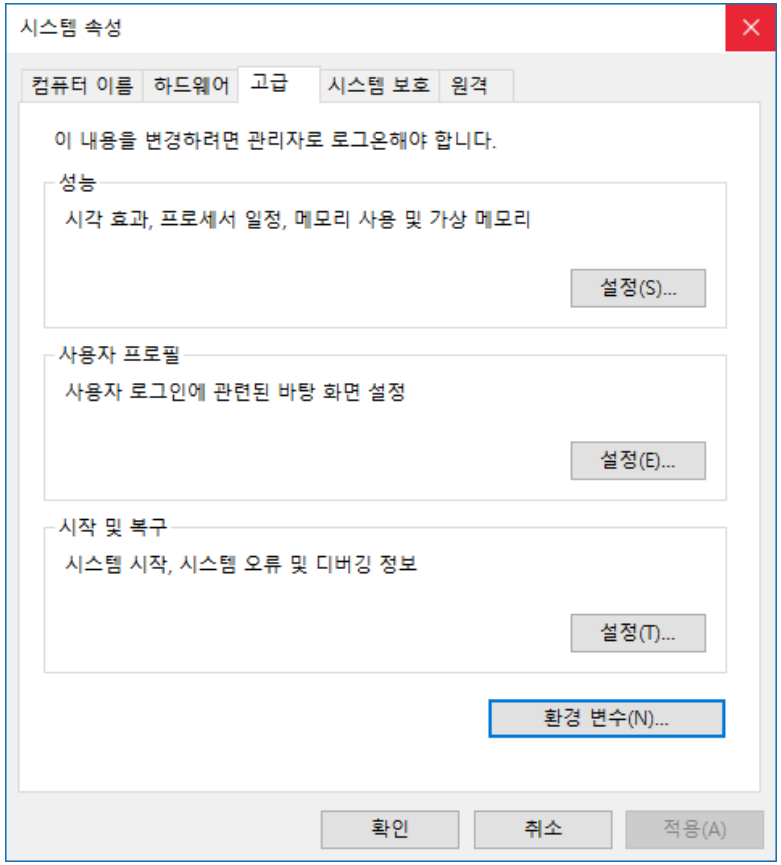


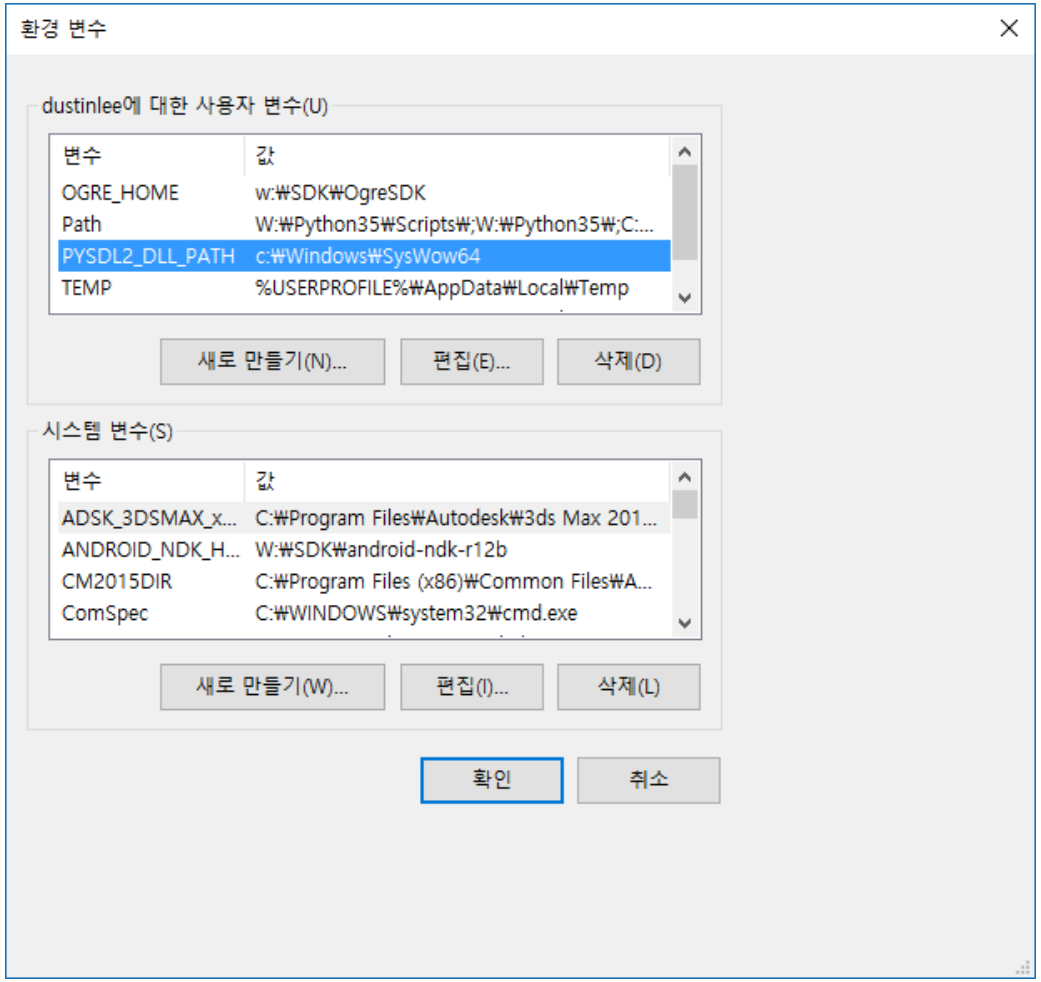
# 64비트의 경우: 환경 변수 PYSDL2_DLL_PATH 의 설정

cmd 실행



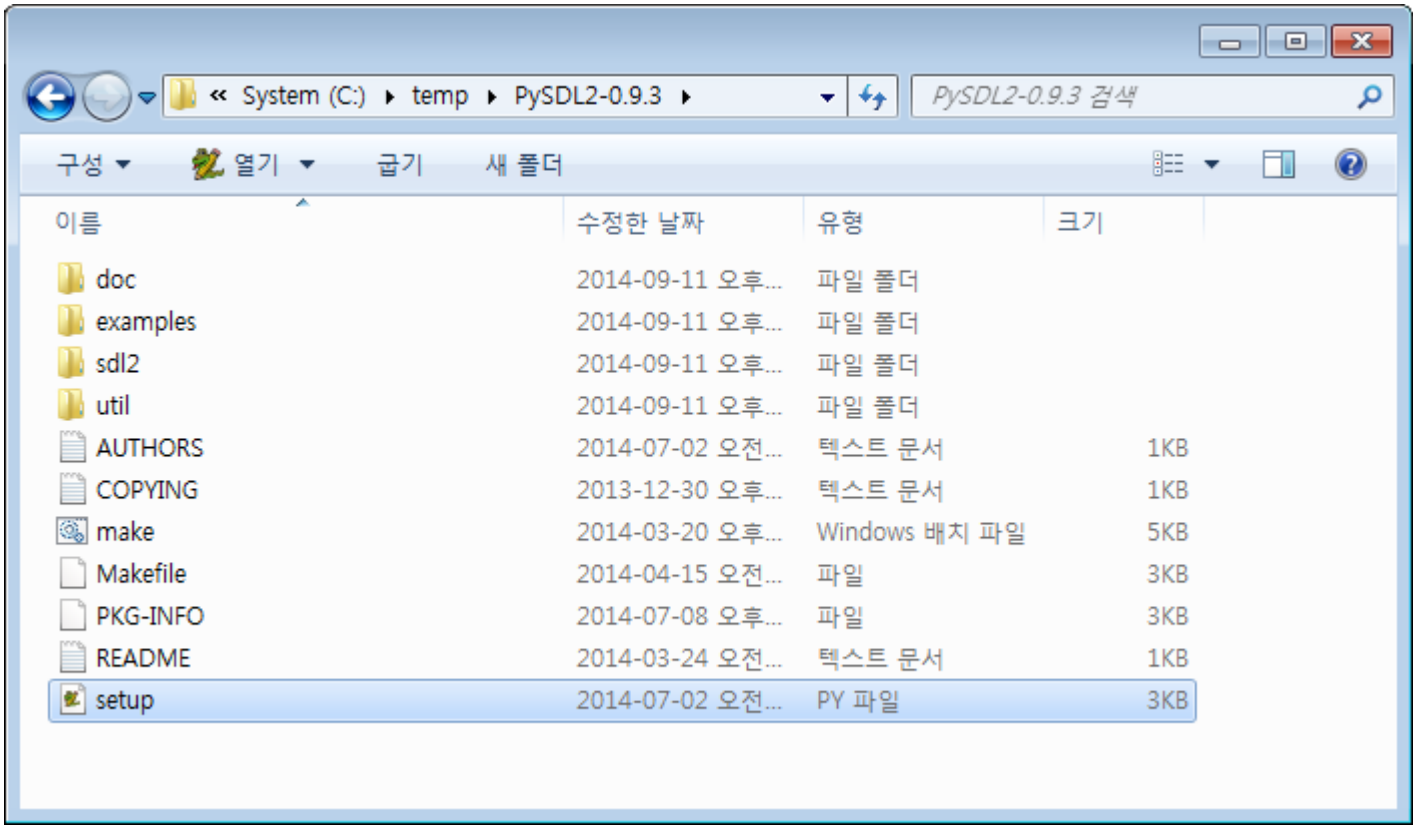
# 시스템 환경 변수 확인



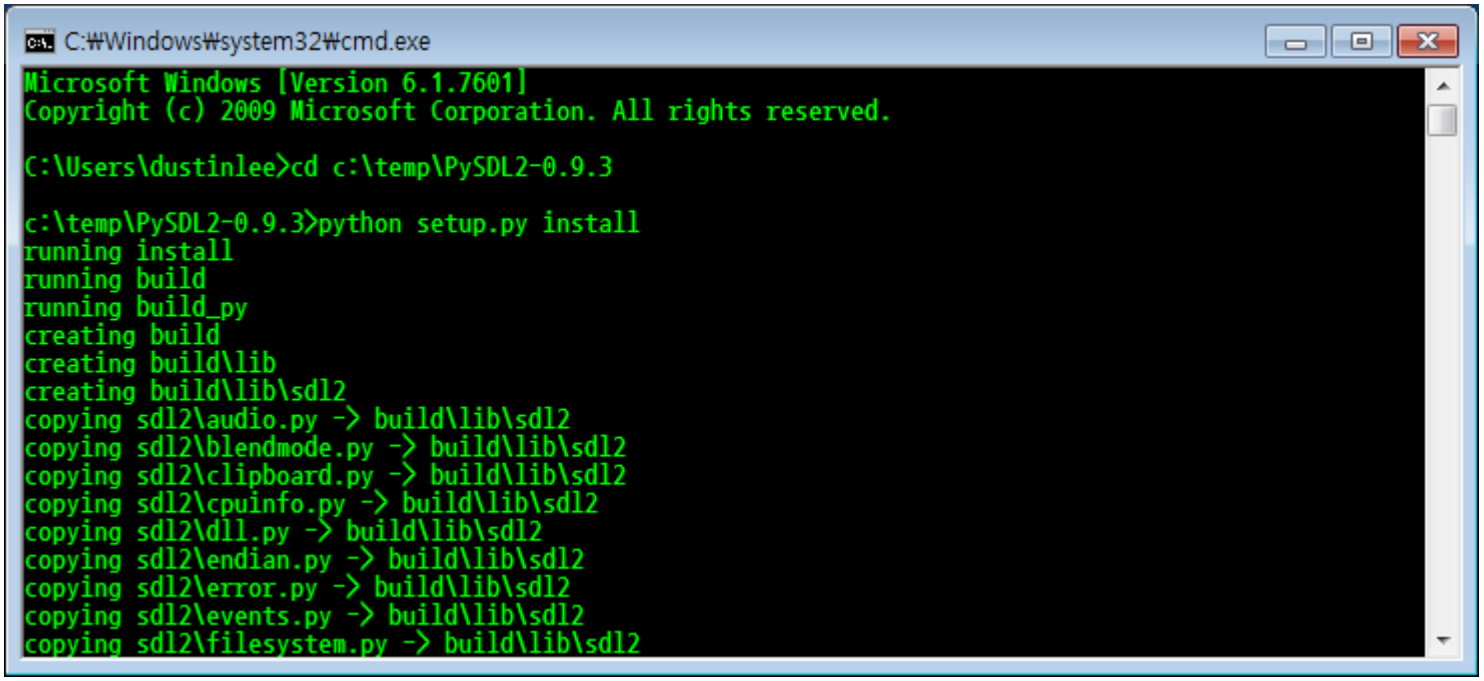



# PySDL 설치

- 적당한 곳에 압축 해제
  - C:₩temp

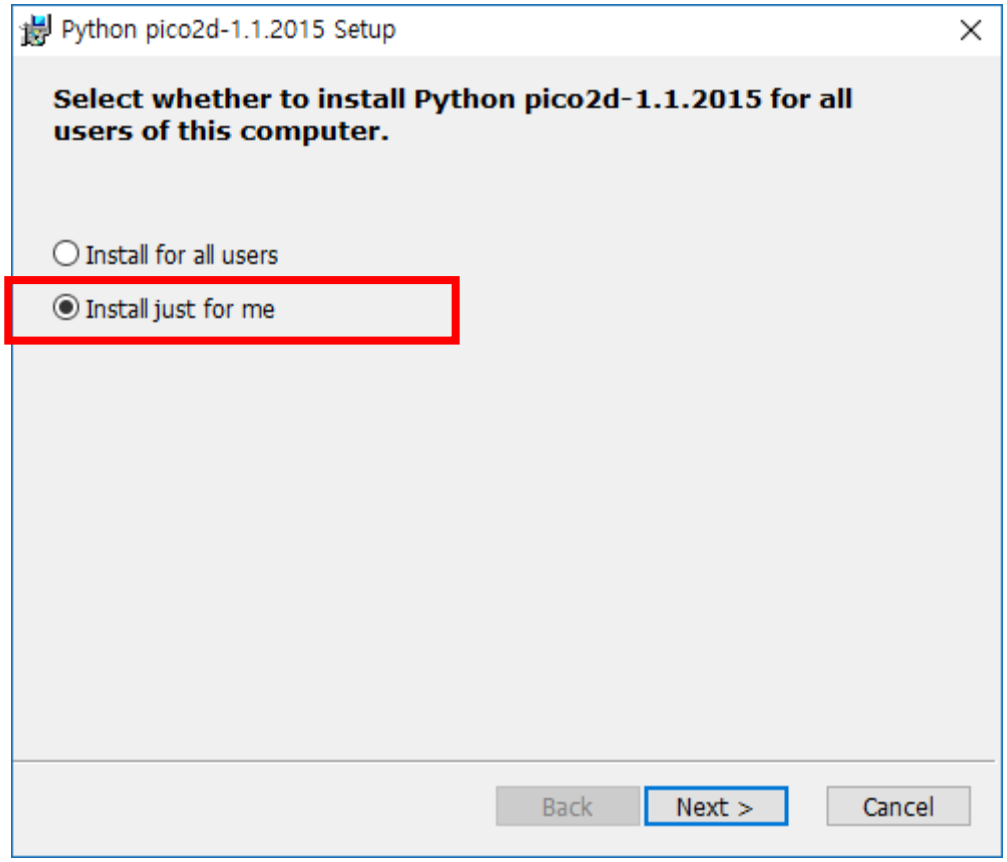


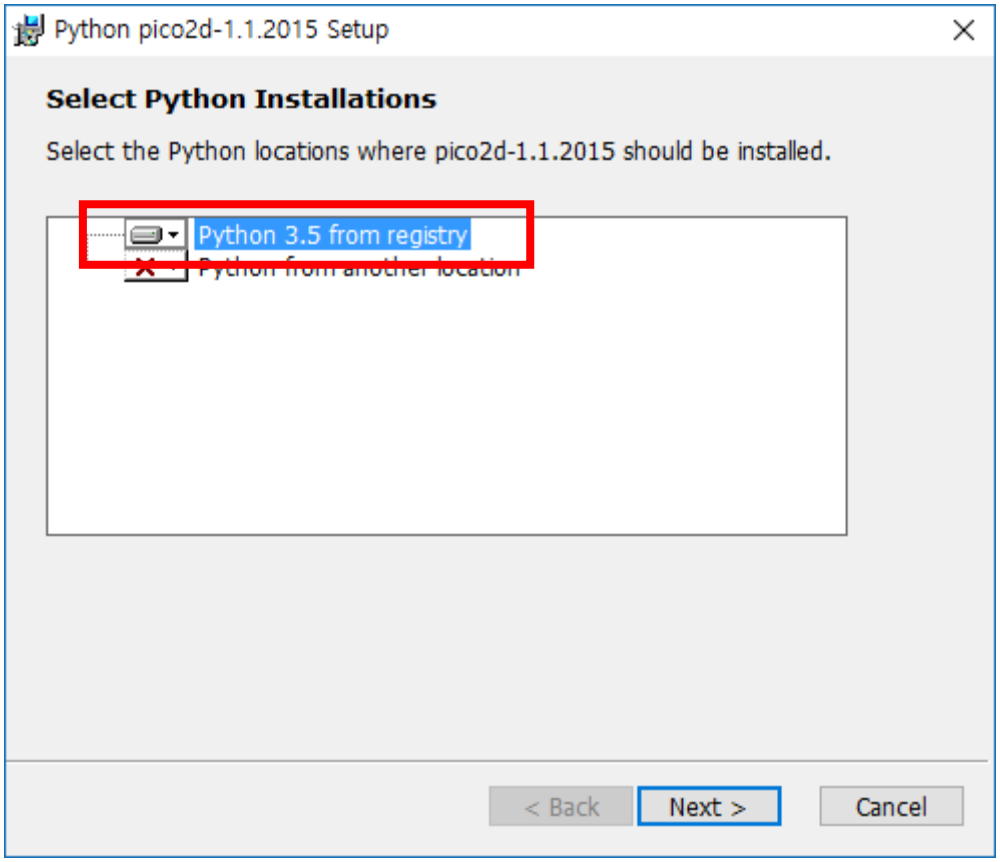
- cmd (코맨드) 창을 열고
- 폴더를 c:₩temp₩PySDL2-0.9.3 으로 변경한 후,
- python setup.py install



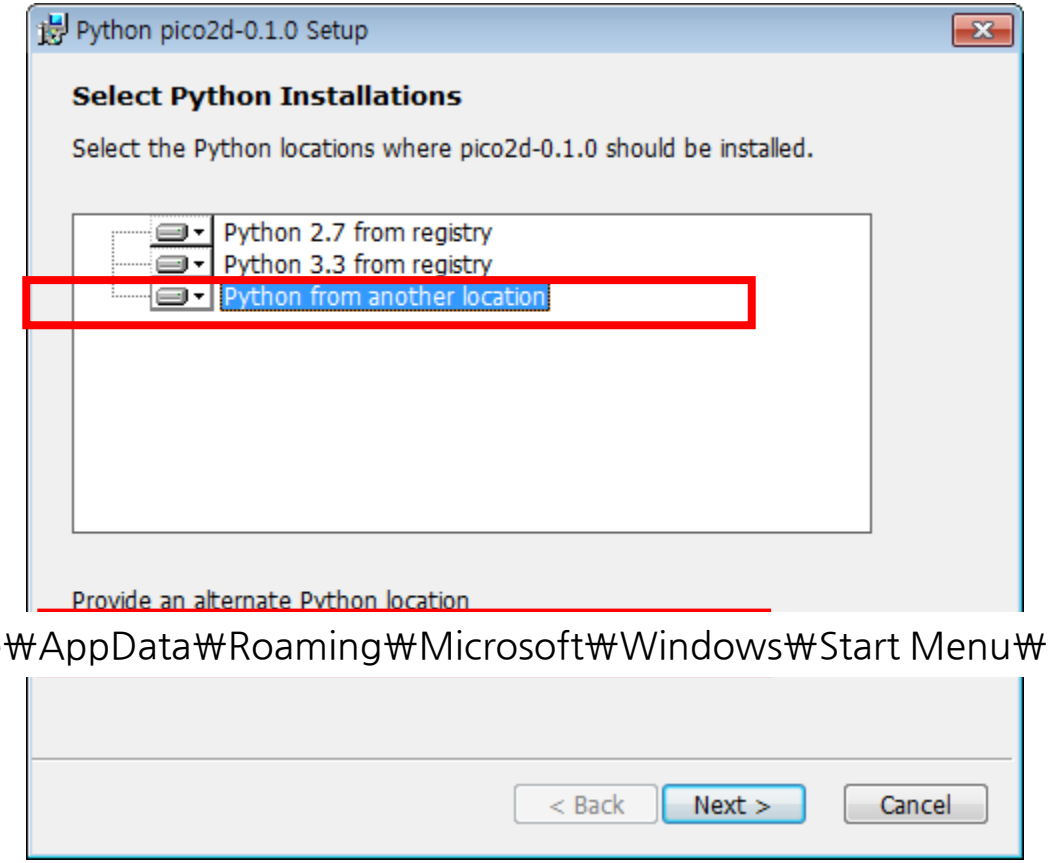
# Pico2d 설치

- 32 bit
  - pico2d 32bit 설치
- 64 bit
  - pico2d 64bit 설치



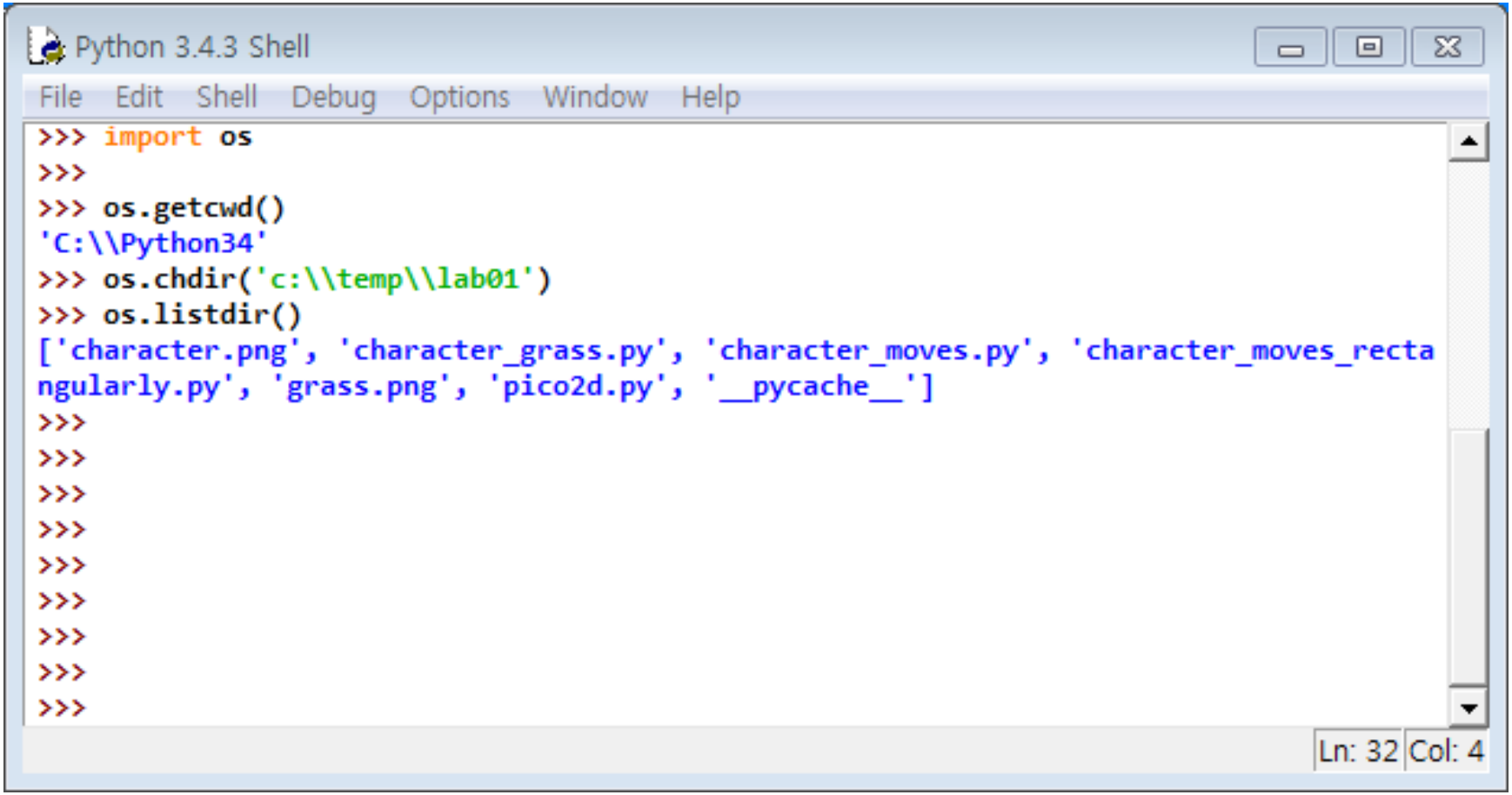


# 만약 python 3.5 from registry 가 나오지 않으면….

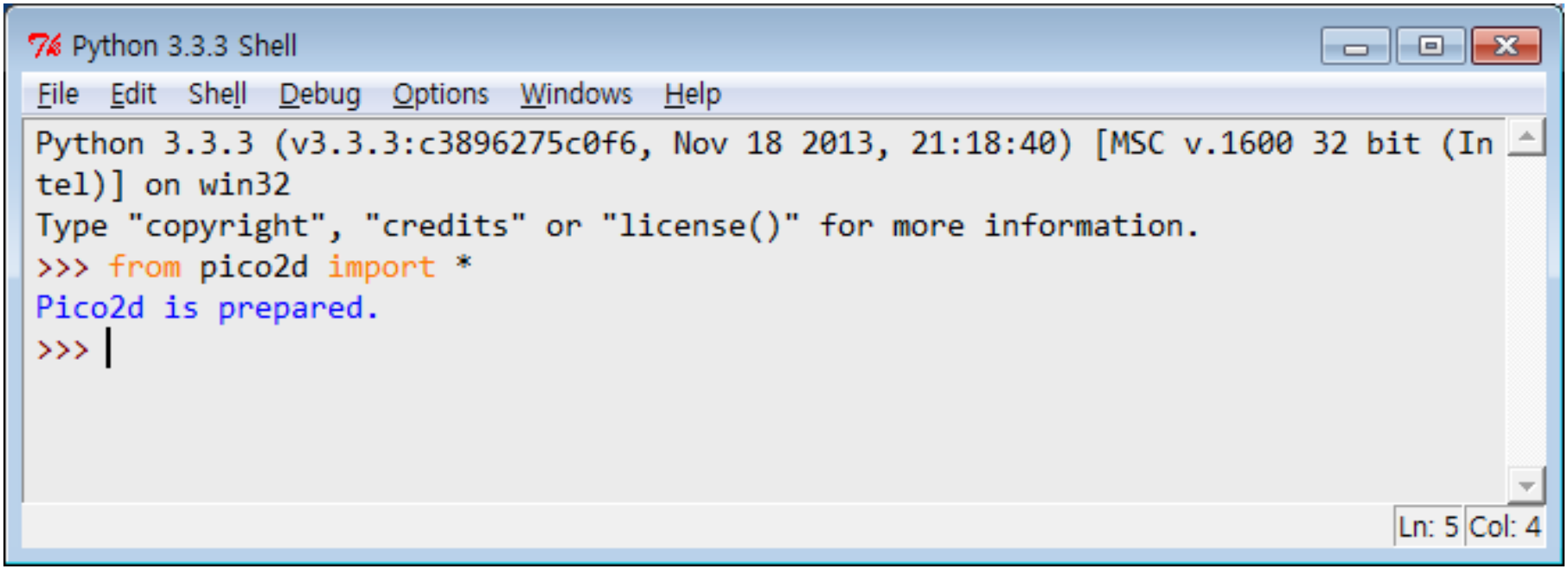



C:₩Users₩dustinlee₩AppData₩Roaming₩Microsoft₩Windows₩Start Menu₩Programs₩Python 3.5

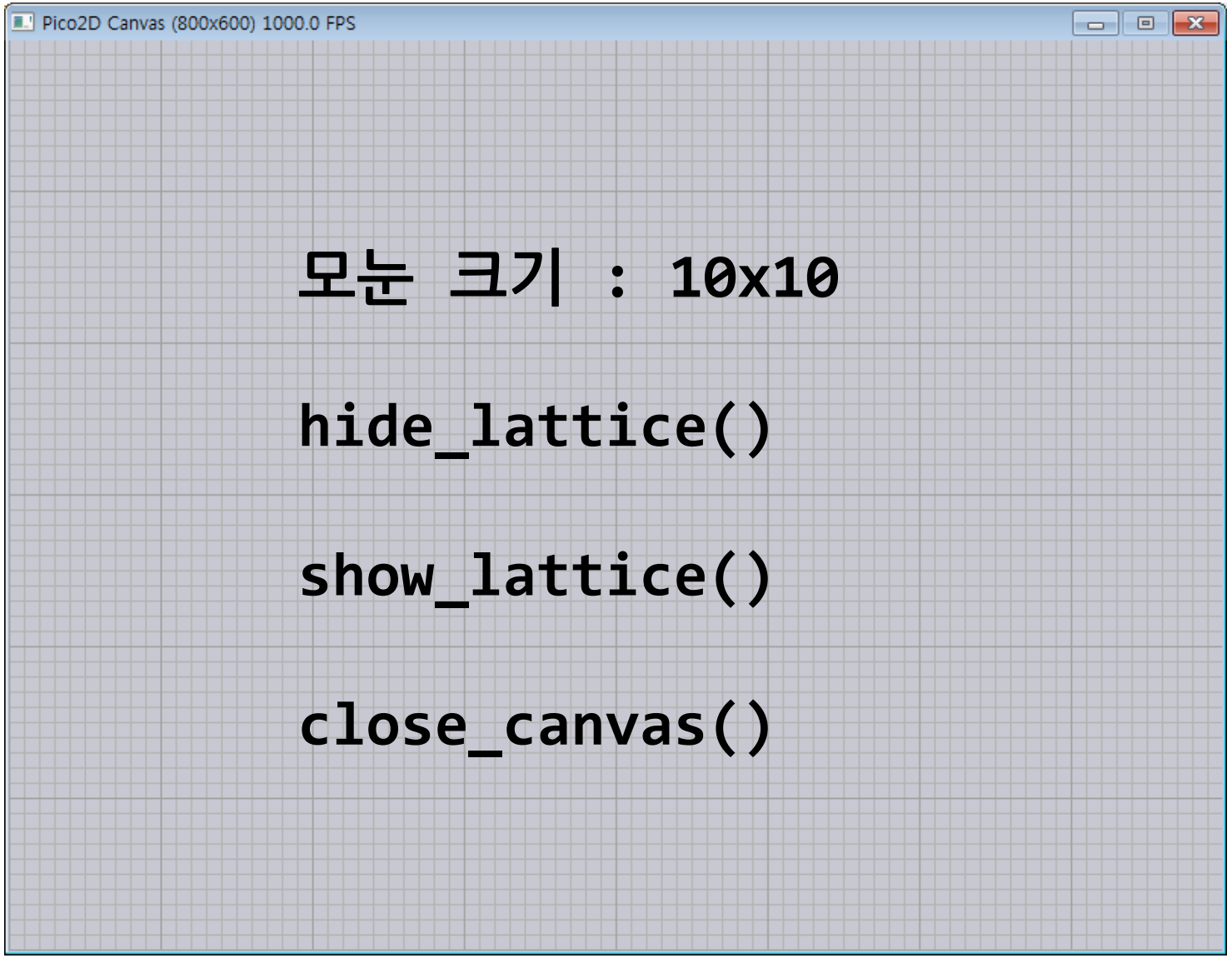
# OS 모듈을 이용한 Working Directory 설정



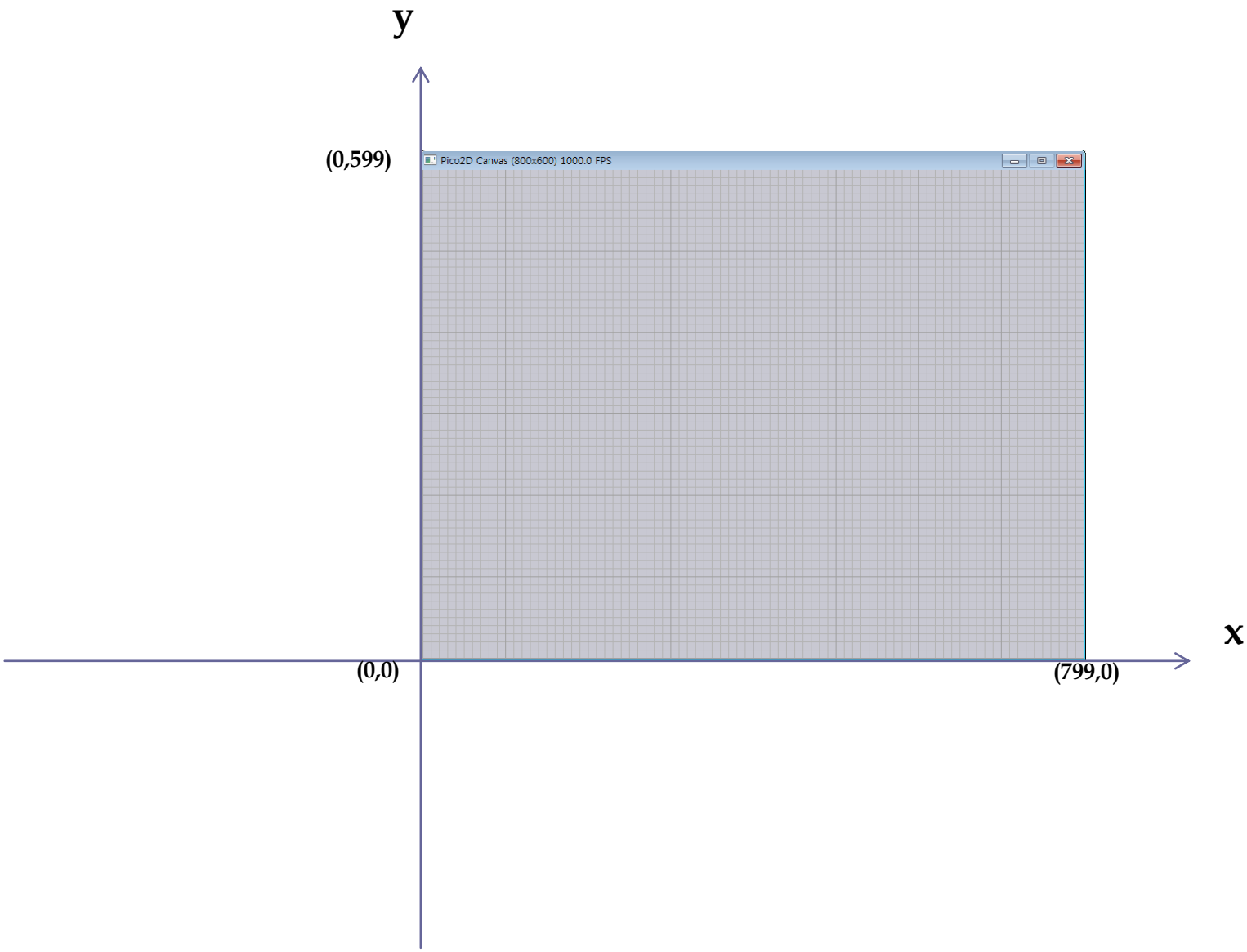
# pico2d 라이브러리 준비



# open_canvas(800, 600)



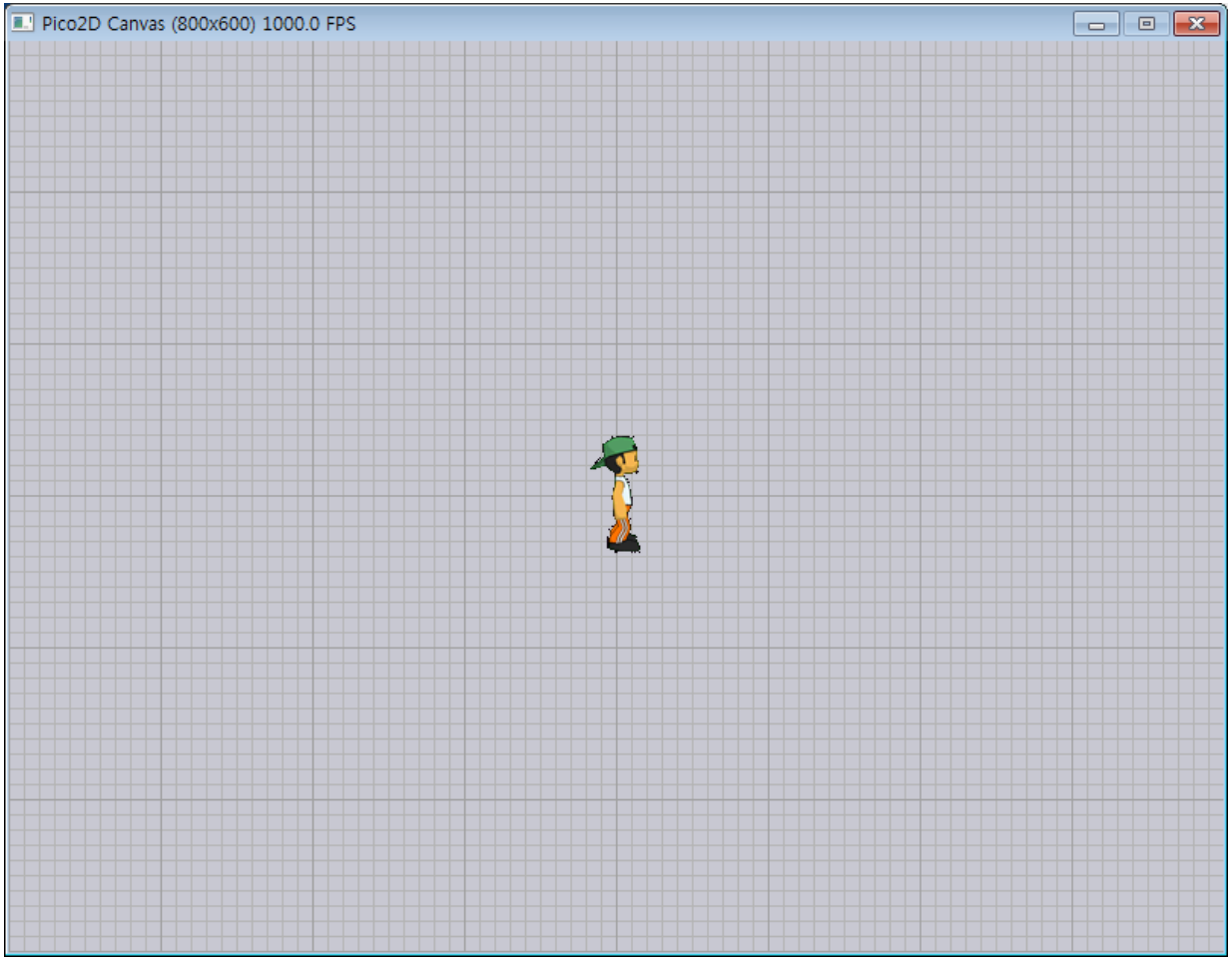
# 캔버스의 좌표계



# JPG vs PNG

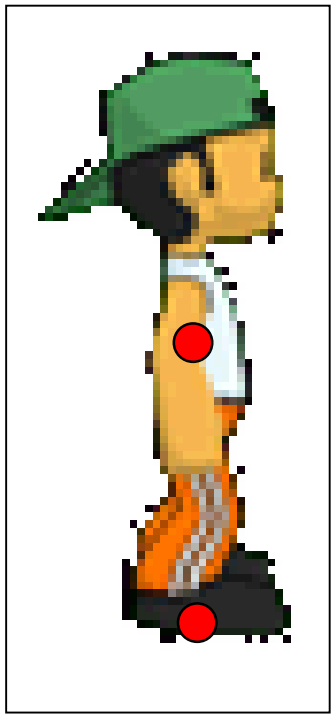
# 우리의 주인공

```
>>> image = load_image('character.png')
```




```
>>> image.draw_now(400,300)
```

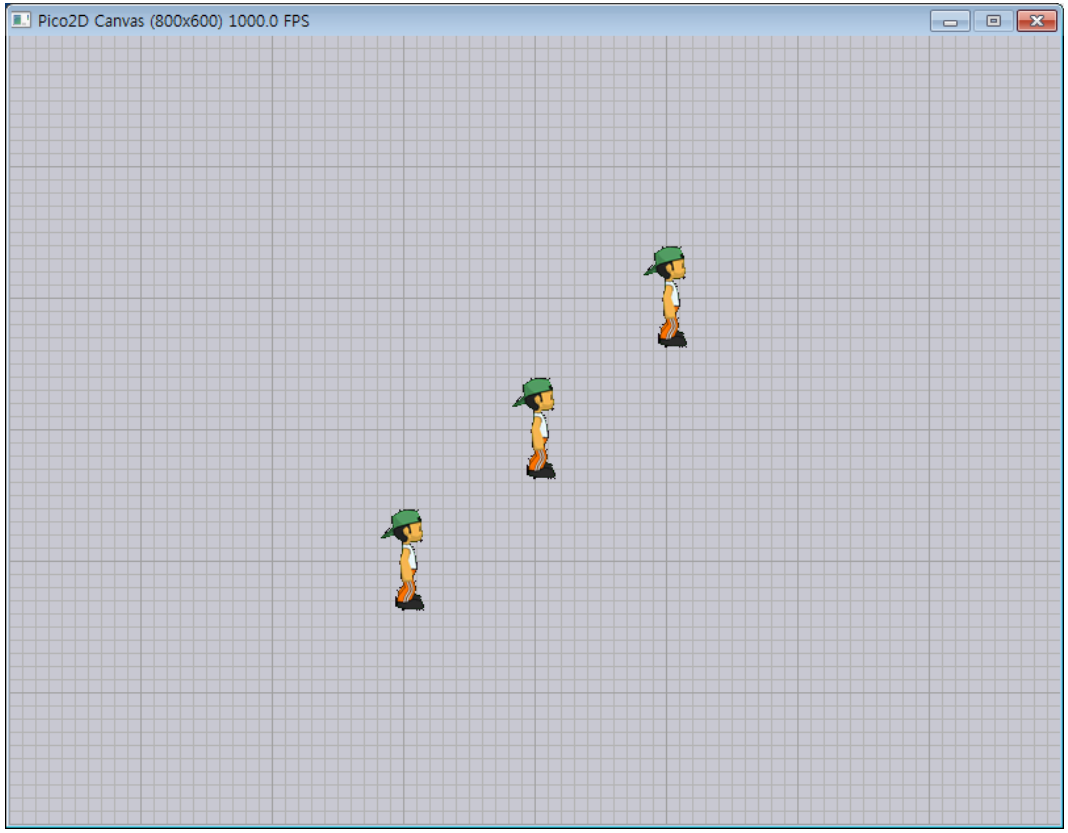
# 피봇(Pivot)



여기가 피봇입니다.

이 점을 피봇으로 삼기도 합니다

# 몇 명 더 그려 봅시다~

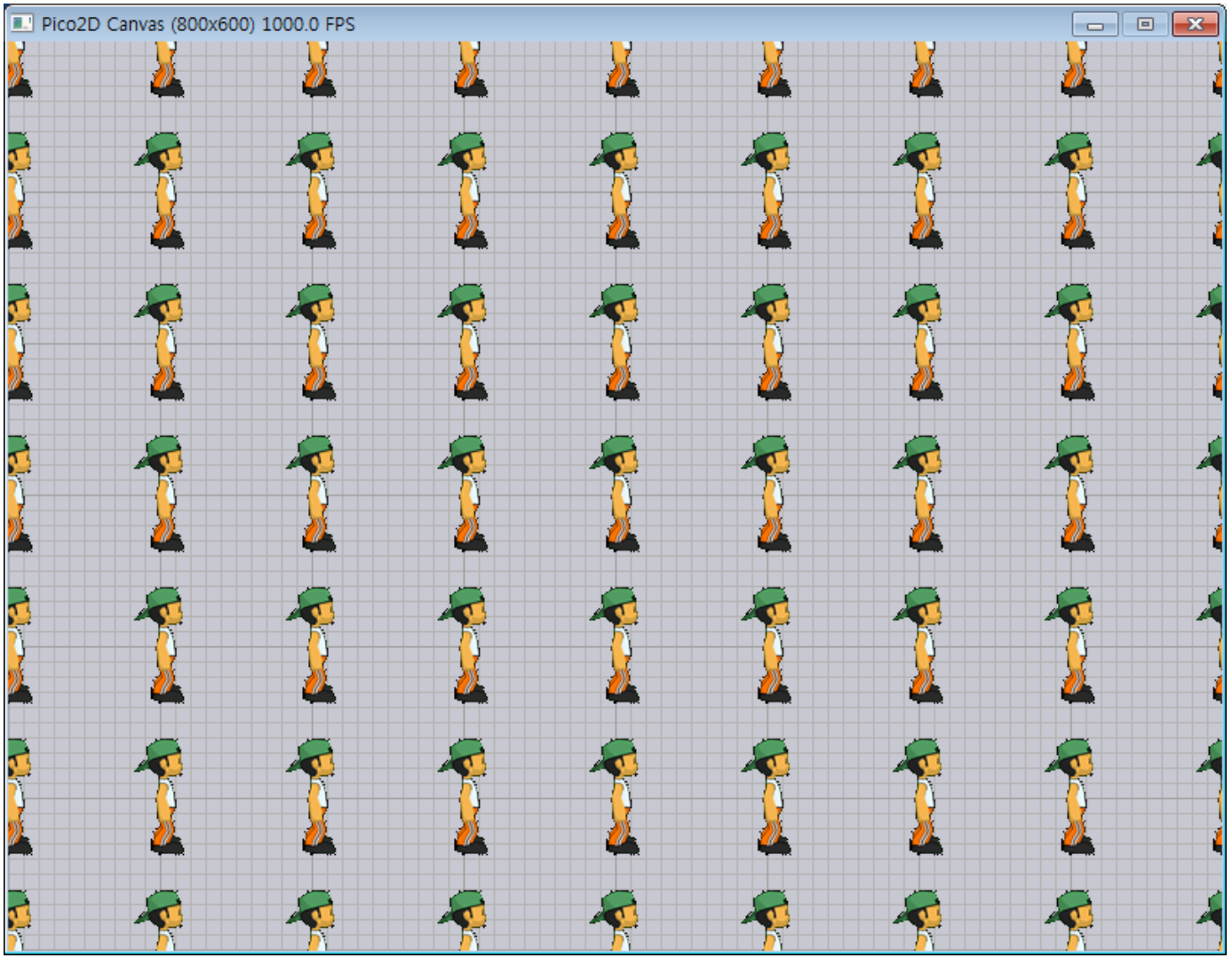


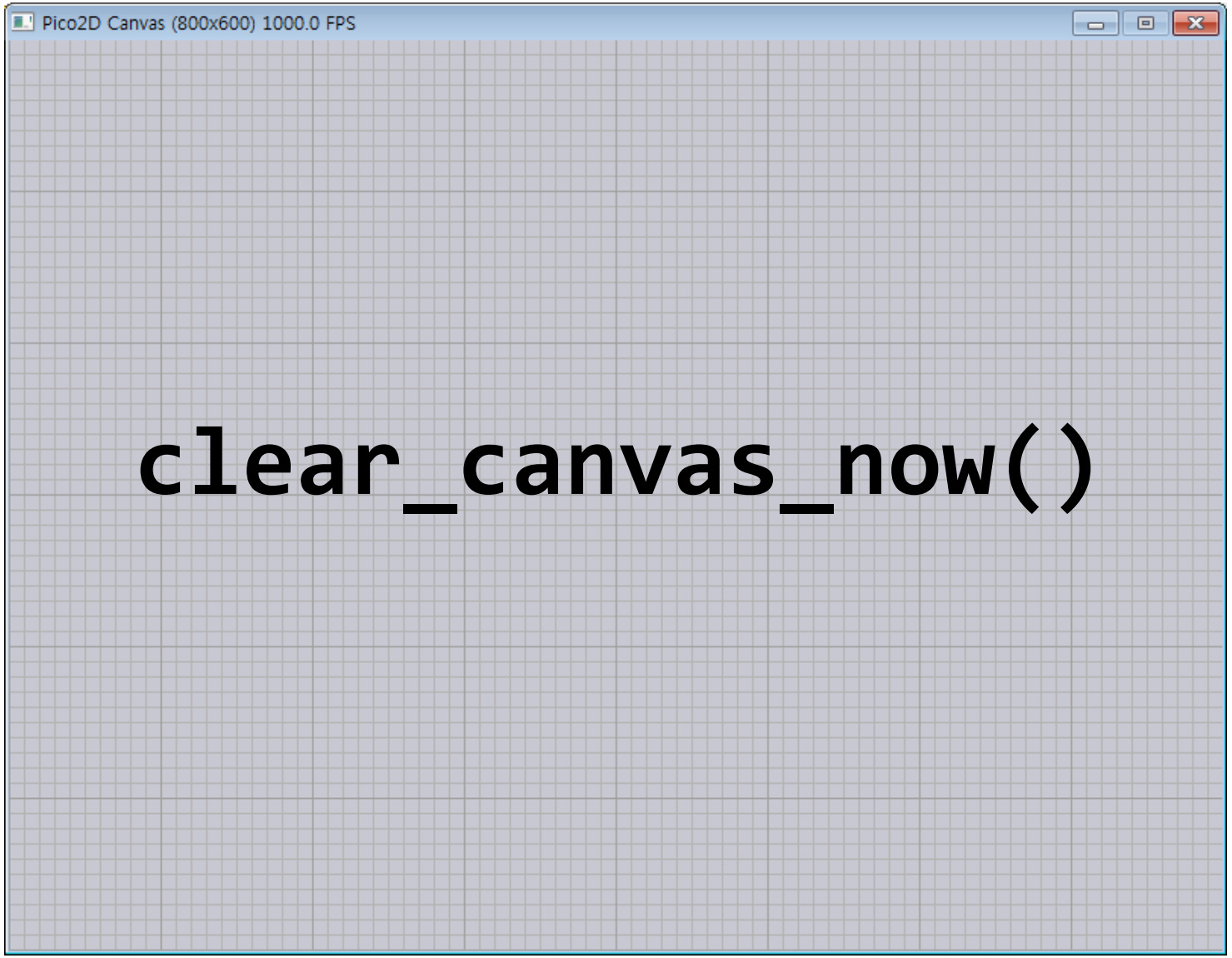

```
>>> image.draw_now(300,200)
>>> image.draw_now(500,400)
```

# 떼로 그리기

```
>>> for x in range(0,9):
        for y in range (0, 7):
            image.draw_now(x * 100, y * 100)
```

# 캐릭터 떼!

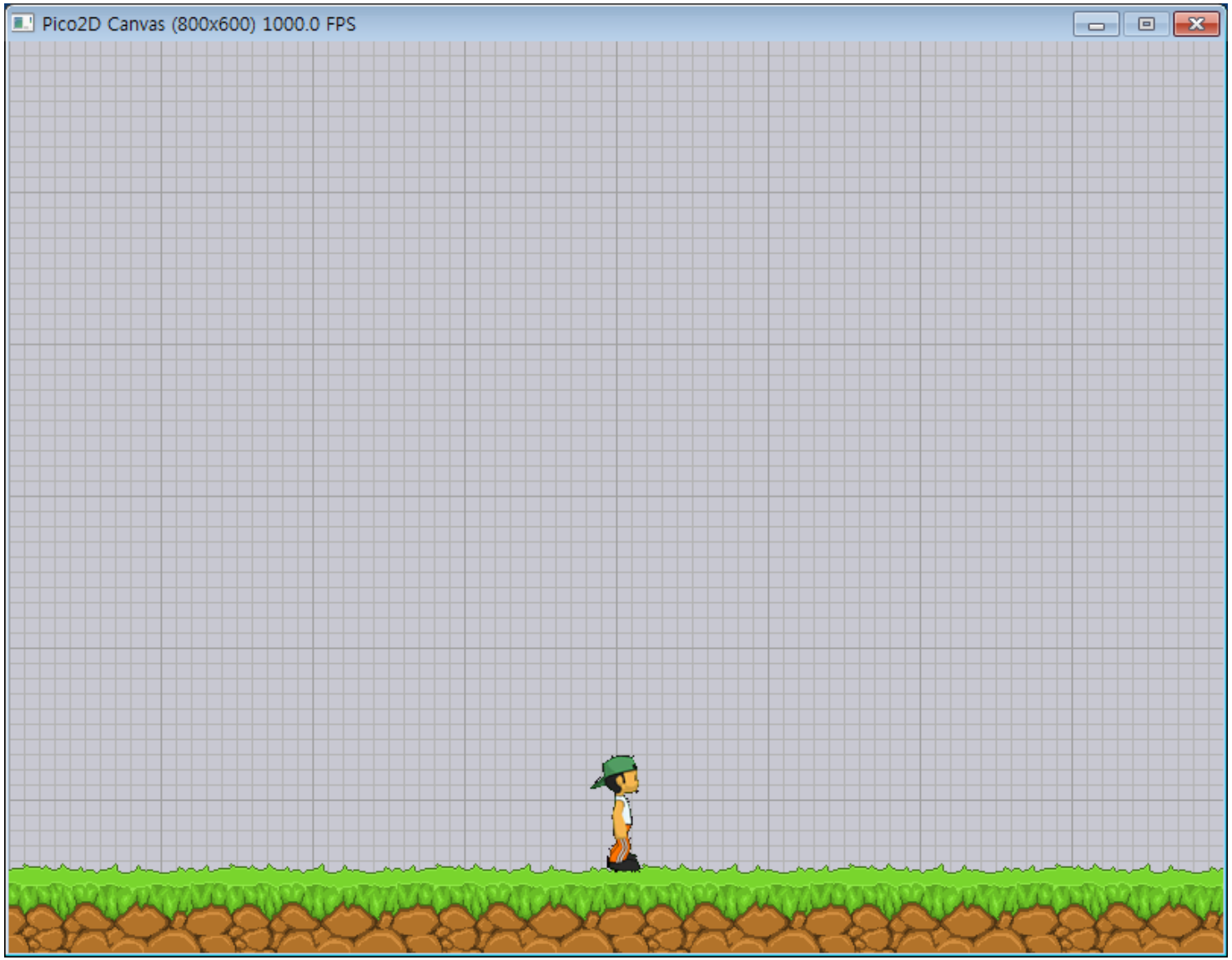





# Character_moves.py
# 캐릭터 이동

# character_grass.py


```
from pico2d import *

open_canvas()

grass = load_image('grass.png')
character = load_image('character.png')

grass.draw_now(400, 30)
character.draw_now(400, 90)

delay(5)

close_canvas()
```



# character_moves.py


```
from pico2d import *

open_canvas()

grass = load_image('grass.png')
character = load_image('character.png')

x = 0
while (x < 800):
    clear_canvas_now()
    grass.draw_now(400, 30)
    character.draw_now(x, 90)
    x = x + 2
    delay(0.01)

close_canvas()
```

# 게임 루프

```
x = 0
while (x < 800):
    clear_canvas_now()
    grass.draw_now(400, 30)
    character.draw_now(x, 90)
    x = x + 2
    delay(0.01)
```