

2D 게임 프로그래밍

- ZOOM 라이브 수업 시, 반드시 카메라를 ON 하고 !
- 입장 이름은 "학번 이름"으로 설정 !
- 미리 수업 git 서버에서 자료를 Pull 해서 준비 !

Lecture #5. 애니메이션

2D 게임 프로그래밍

이대현 교수

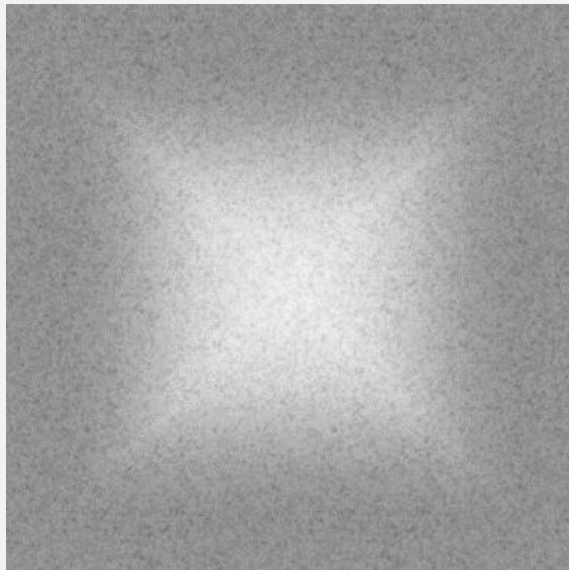
학습 내용

- 더블 버퍼링

- 플리핑

- 스프라이트 애니메이션

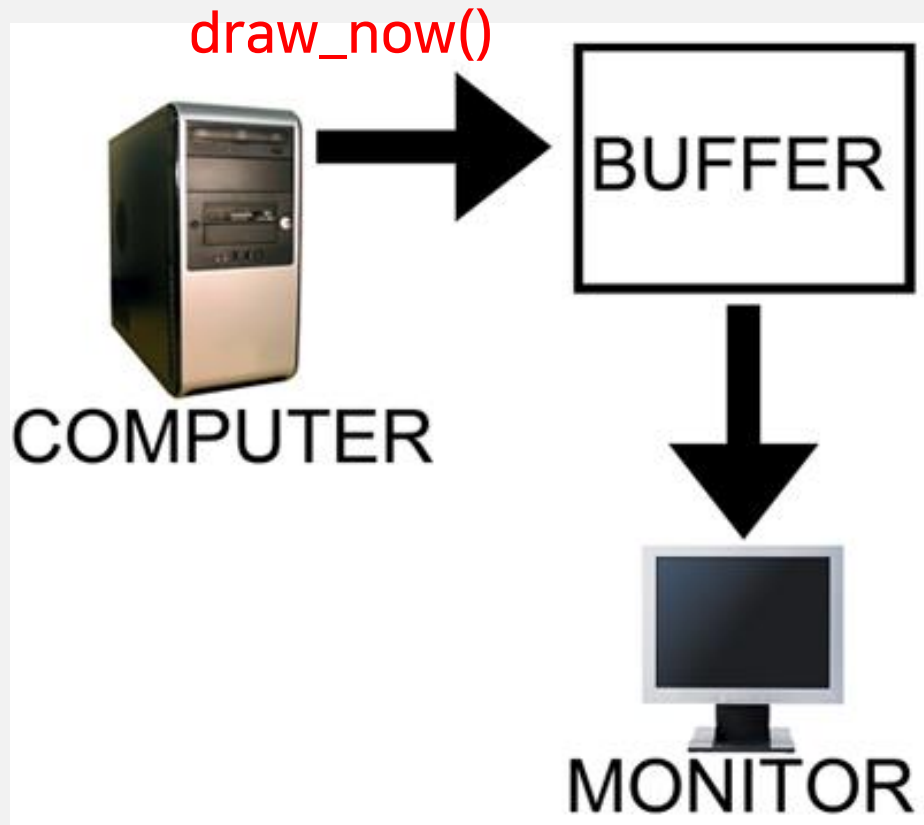
지난 번 실습의 문제점? - 화면 플리커링



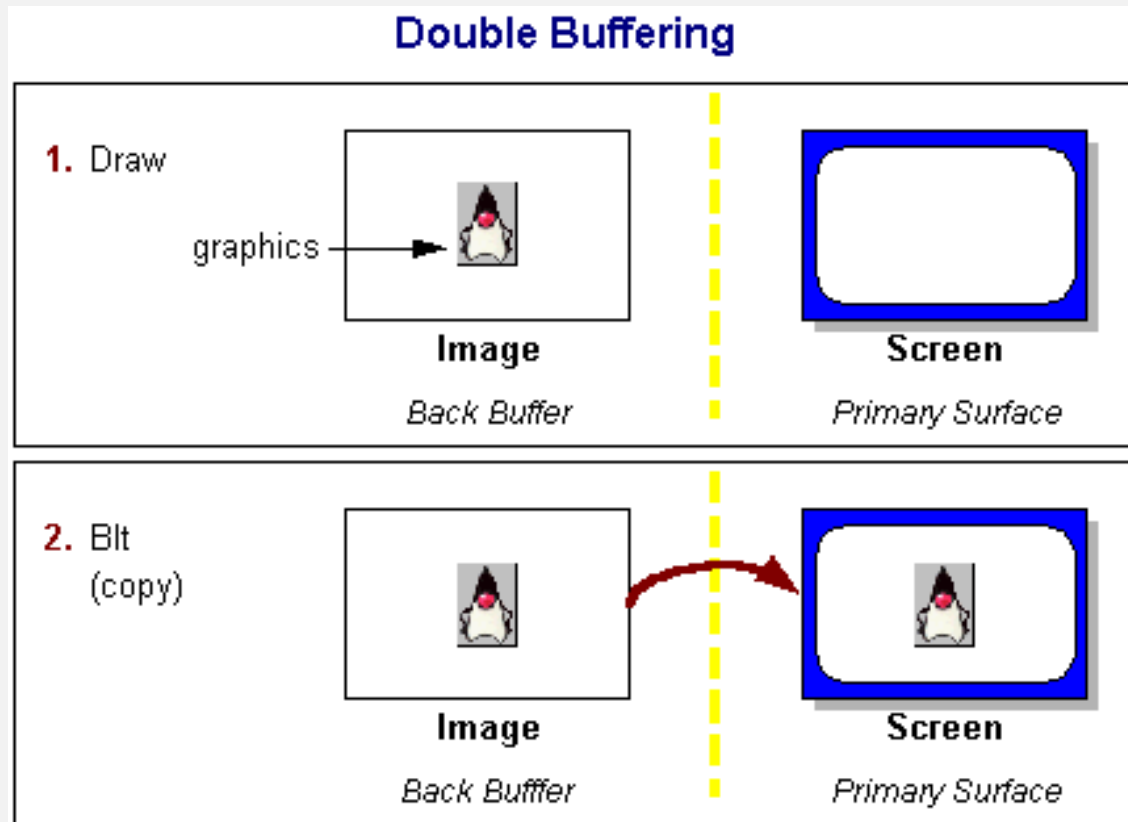
무대의 커튼은 왜 있을까?



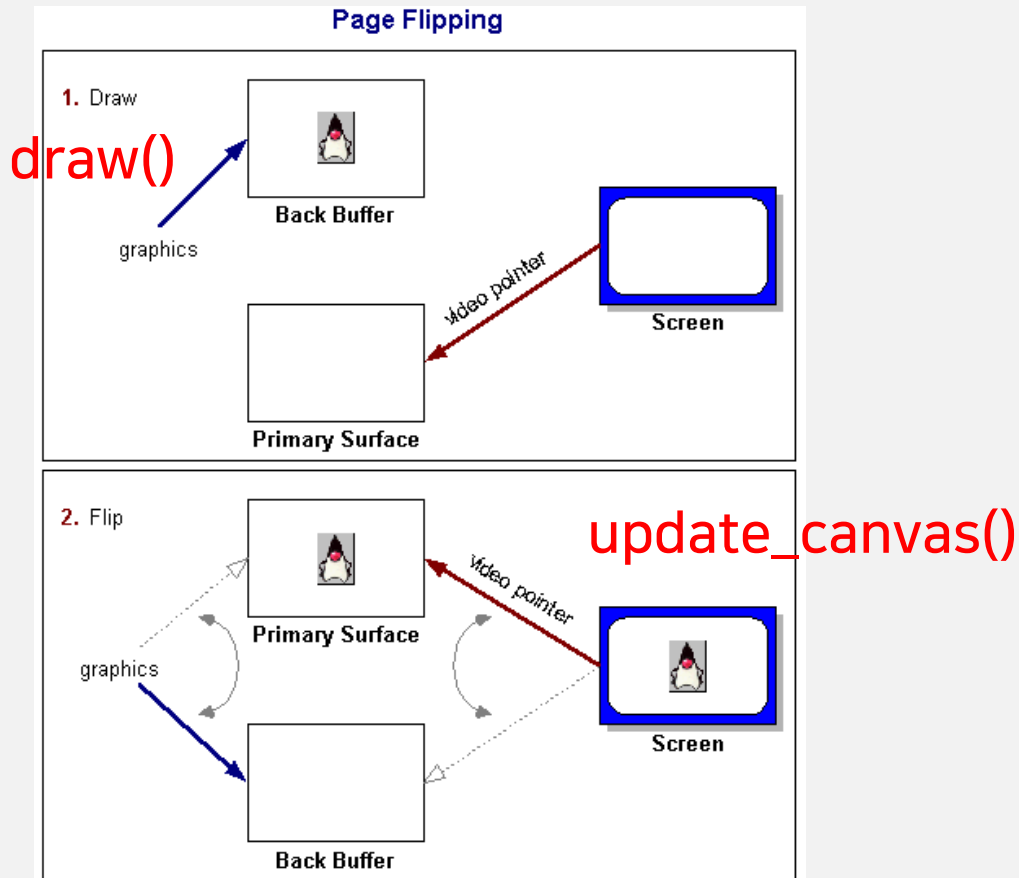
프레임 버퍼(Frame Buffer)



더블 버퍼링(Double Buffering)



페이지 플리핑(Page Flipping)



후면 버퍼(Back Buffer)에 그리기

```
>>> from pico2d import *  
>>> open_canvas()  
>>> character = load_image('character.png')  
>>> character.draw(100,100)  
>>> character.draw(200,200)
```

```
>>> update_canvas()
```

화면 지우기

```
>>> clear_canvas()  
>>> update_canvas()
```

Python IDE

■PyCharm

- 파이썬용으로 가장 많이 사용되는 IDE
- PC, 리눅스, 맥 등 다양한 환경에서 사용 가능

■Visual Studio

- C/C++/C# 이외에 파이썬 개발도 가능

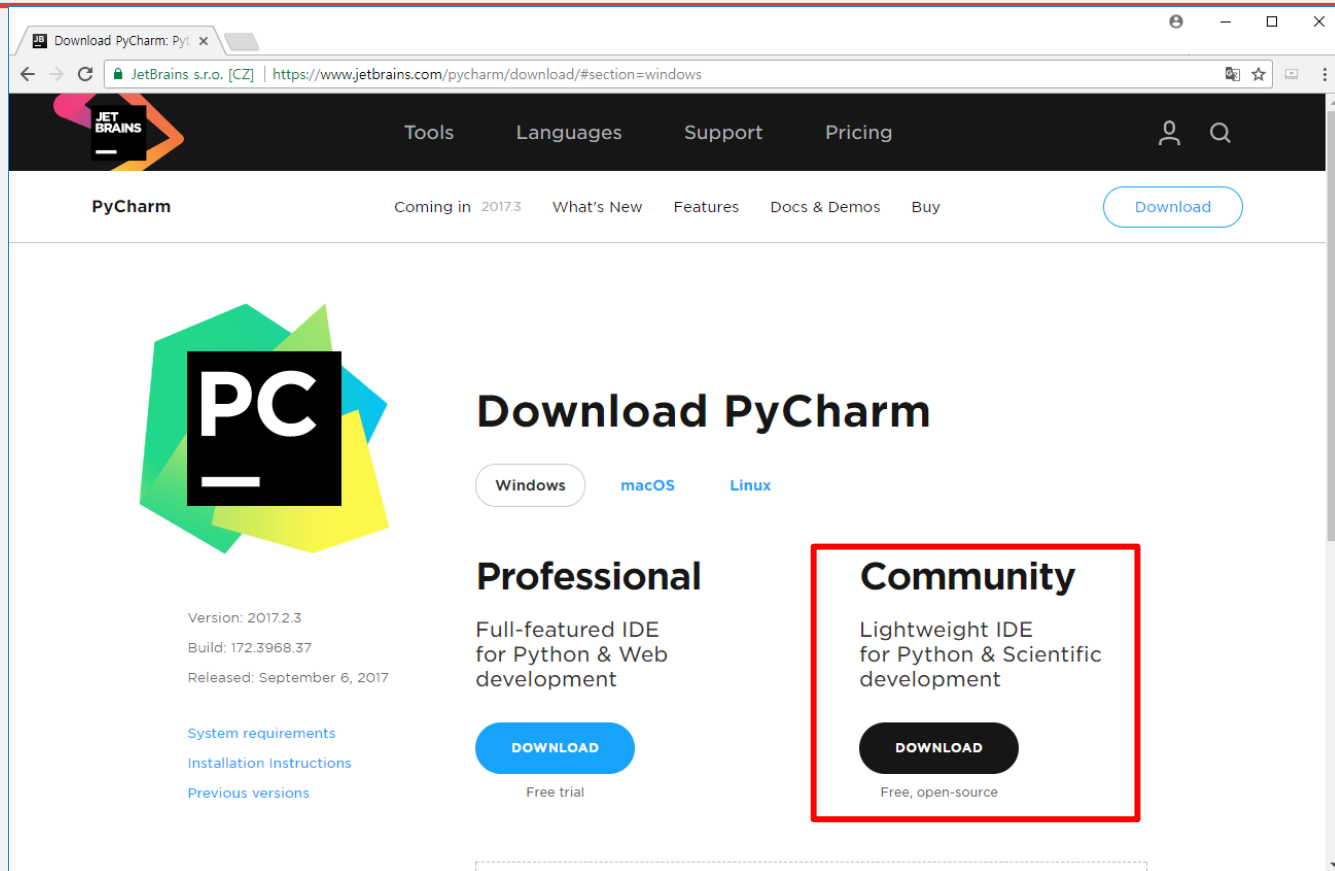
■Visual Studio Code

- 파이썬을 연계 사용 가능한 소스 코드 도구
- PC, Linux, 맥 등에서 공통적으로 사용 가능

PyCharm (<https://www.jetbrains.com/pycharm/>)



PyCharm 설치 – community version



The screenshot shows the JetBrains website's download page for PyCharm. The browser address bar shows the URL <https://www.jetbrains.com/pycharm/download/#section=windows>. The page features a dark navigation bar with the JetBrains logo and links for Tools, Languages, Support, Pricing, and a user profile icon. Below this, a secondary navigation bar includes 'PyCharm', 'Coming in 2017.3', 'What's New', 'Features', 'Docs & Demos', 'Buy', and a prominent 'Download' button. The main content area is titled 'Download PyCharm' and includes tabs for 'Windows', 'macOS', and 'Linux'. On the left, the PyCharm logo is displayed alongside version information: 'Version: 2017.2.3', 'Build: 172.3968.37', and 'Released: September 6, 2017'. Below this are links for 'System requirements', 'Installation instructions', and 'Previous versions'. Two main download options are presented: 'Professional' (Full-featured IDE for Python & Web development) and 'Community' (Lightweight IDE for Python & Scientific development). The 'Community' option is highlighted with a red rectangular border. Each option has a 'DOWNLOAD' button and a note: 'Free trial' for Professional and 'Free, open-source' for Community.

Download PyCharm: Pyi x

JetBrains s.r.o. [CZ] | <https://www.jetbrains.com/pycharm/download/#section=windows>

Tools Languages Support Pricing

PyCharm Coming in 2017.3 What's New Features Docs & Demos Buy [Download](#)

PC

Version: 2017.2.3
Build: 172.3968.37
Released: September 6, 2017

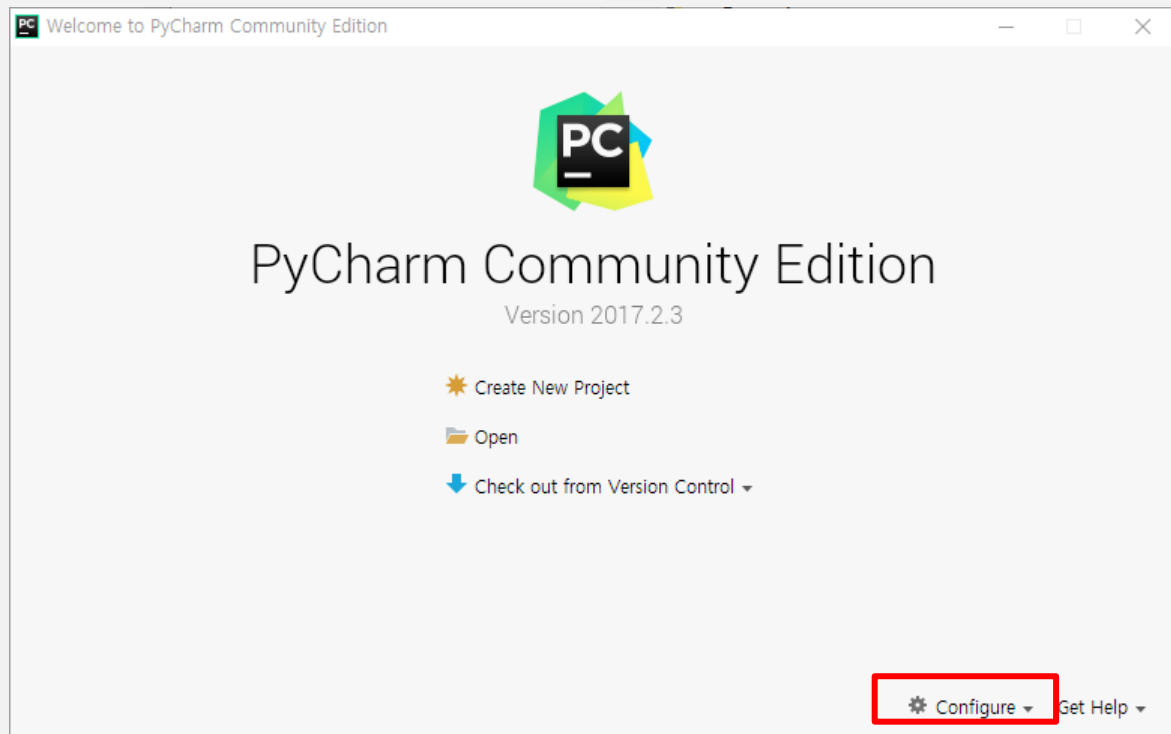
[System requirements](#)
[Installation instructions](#)
[Previous versions](#)

Download PyCharm

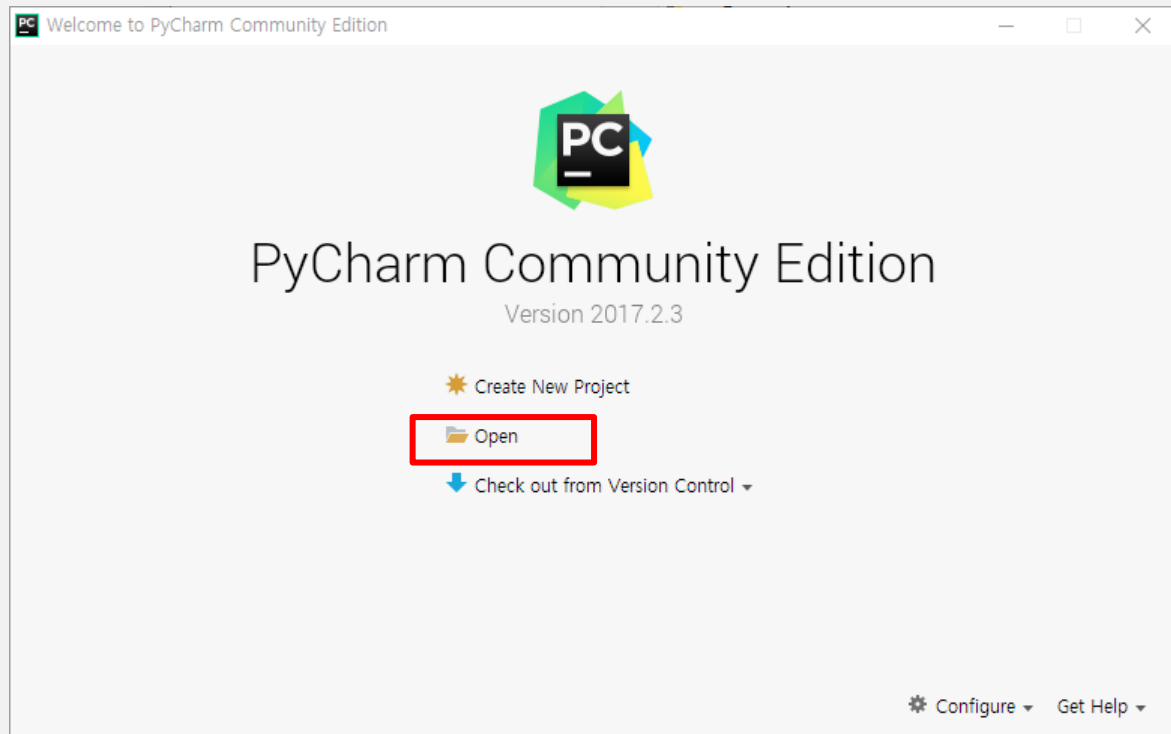
Windows macOS Linux

Professional
Full-featured IDE for Python & Web development
[DOWNLOAD](#)
Free trial

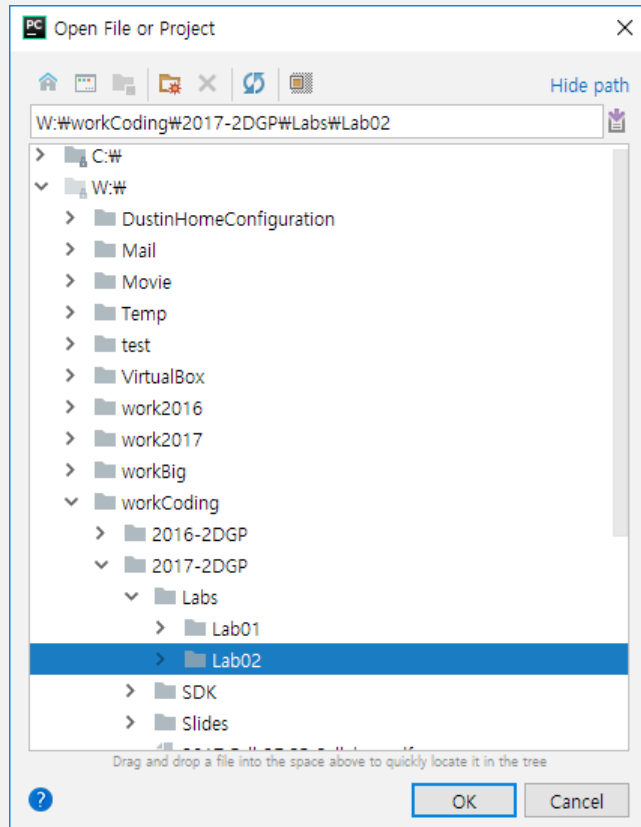
Community
Lightweight IDE for Python & Scientific development
[DOWNLOAD](#)
Free, open-source



PyCharm의 실행



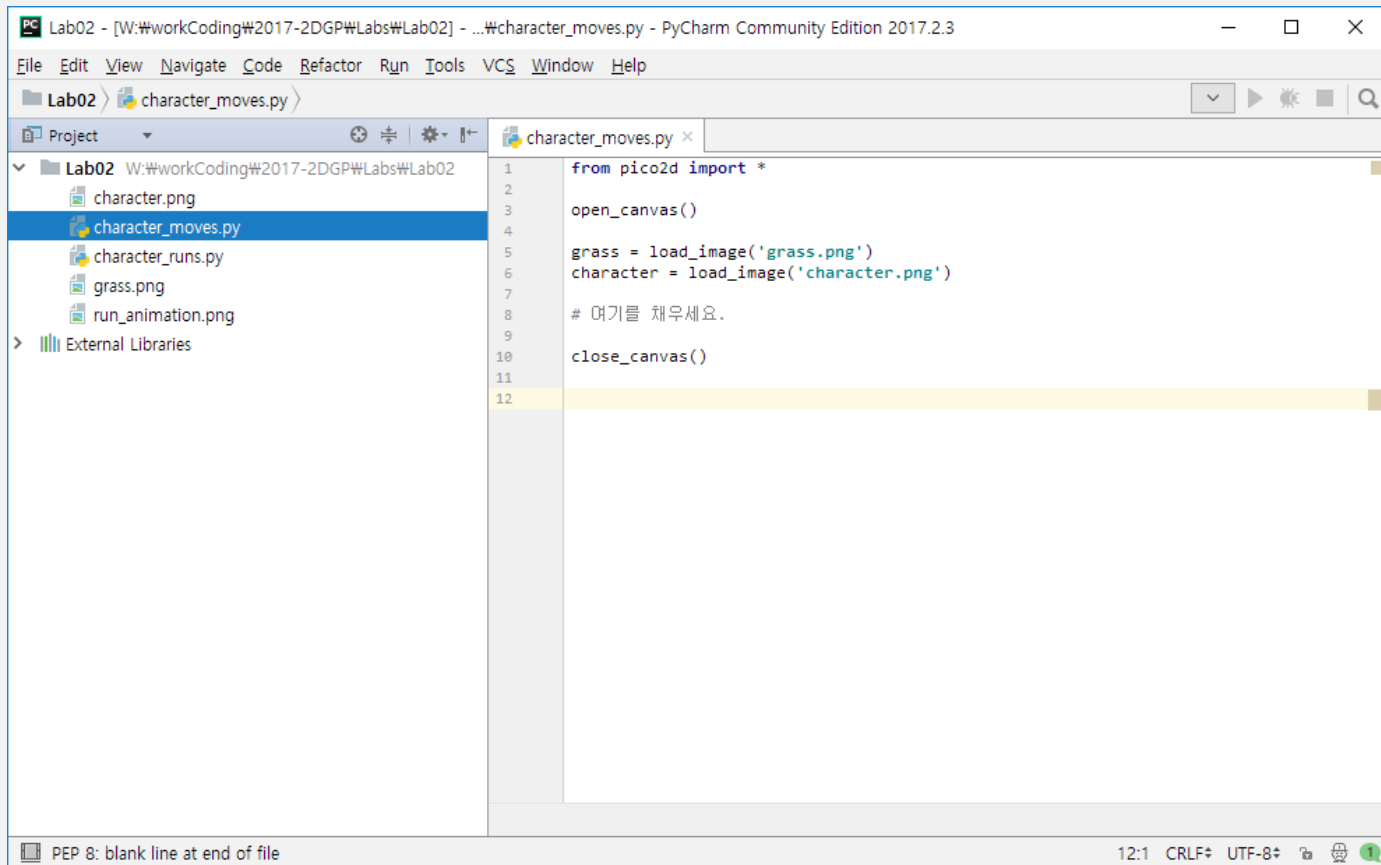
폴더 선택





부드러운 캐릭터 이동

character_moves.py 선택 및 코드 입력



character_moves.py



```
from pico2d import *

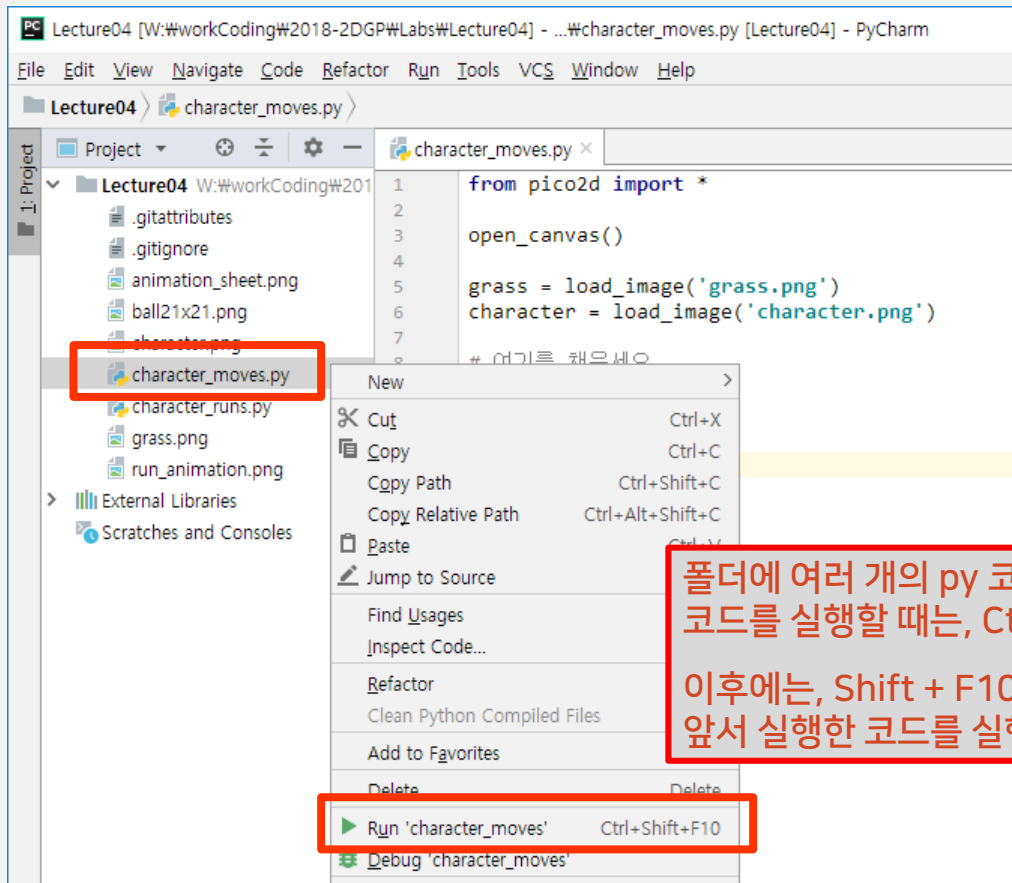
open_canvas()

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

x = 0
while (x < 800):
    clear_canvas()
    grass.draw(400, 30)
    character.draw(x, 90)
    x = x + 2
    update_canvas()
    delay(0.01)
    get_events()

close_canvas()
```

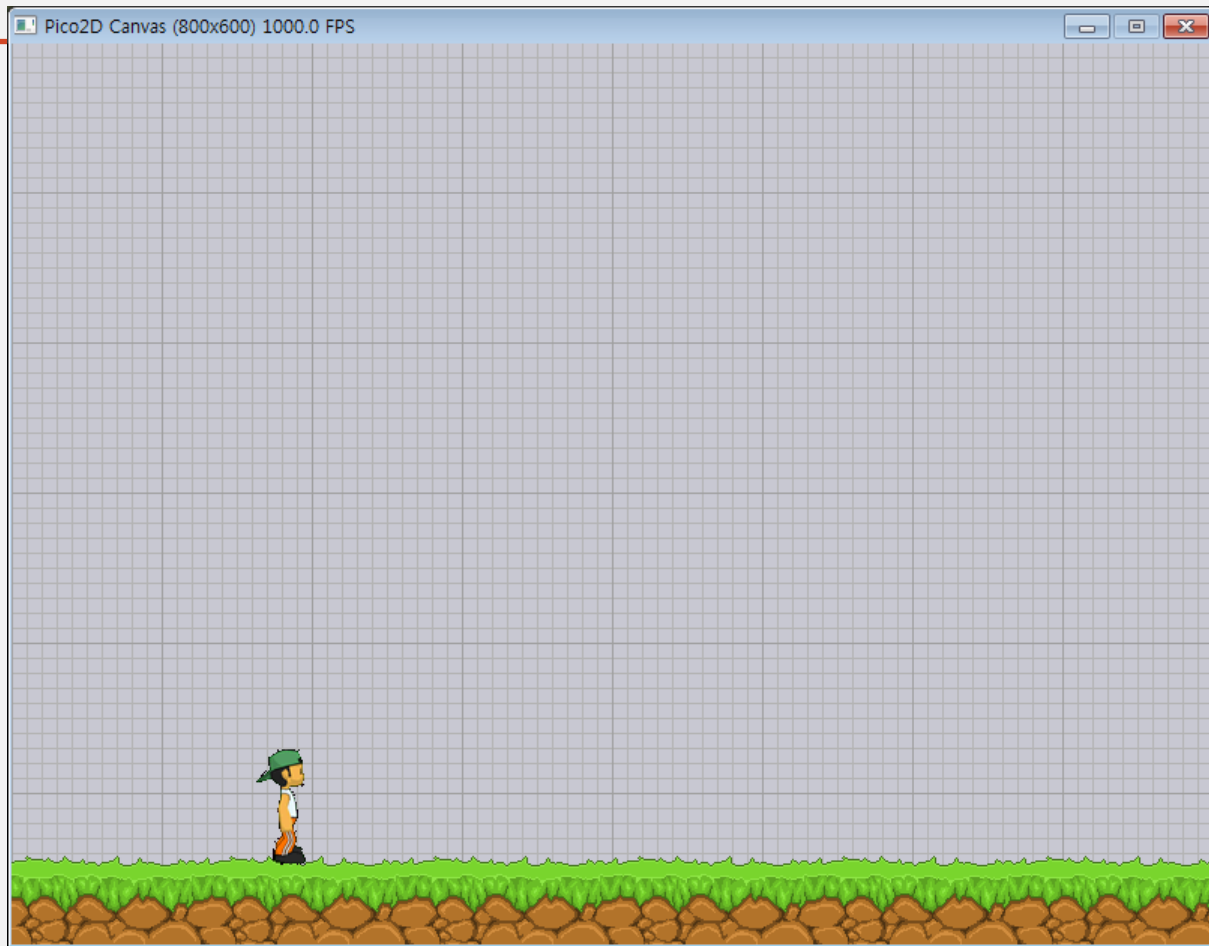
선택한 코드의 실행(Ctrl + Shift + F10)



폴더에 여러 개의 py 코드가 있을 경우, 선택한 코드를 실행할 때는, Ctrl + Shift + F10

이후에는, Shift + F10 을 하면, 지속적으로 앞서 실행한 코드를 실행할 수 있음.

실행 결과



스프라이트(Sprite)

■스프라이트란?

- 게임 장면안에서 보여지는 이미지 또는 애니메이션되는 오브젝트
- 2D 게임에서는 게임의 모든 캐릭터들과 이동하는 물체들을 표현하는 데 사용됨.
- 3D 게임에서는 2D로 표현될 수 있는 각종 오브젝트에 사용됨.
 - 불, 연기, 작은 물체들, UI 표시 등등.

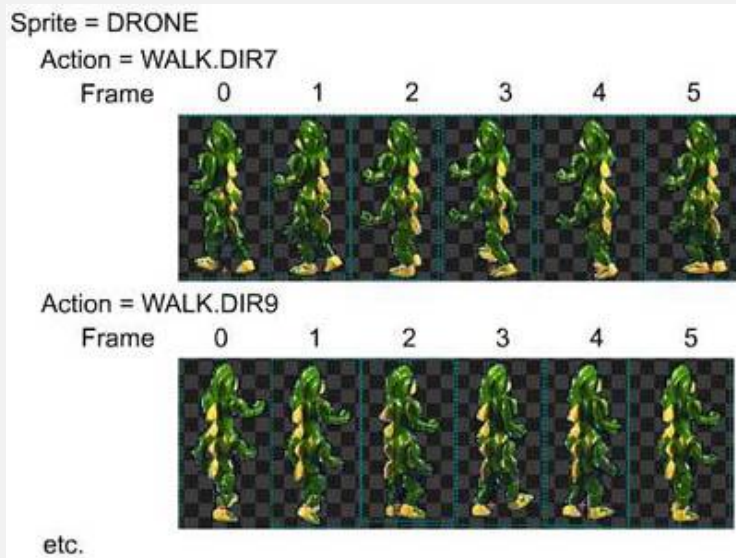


Metal Slug 3

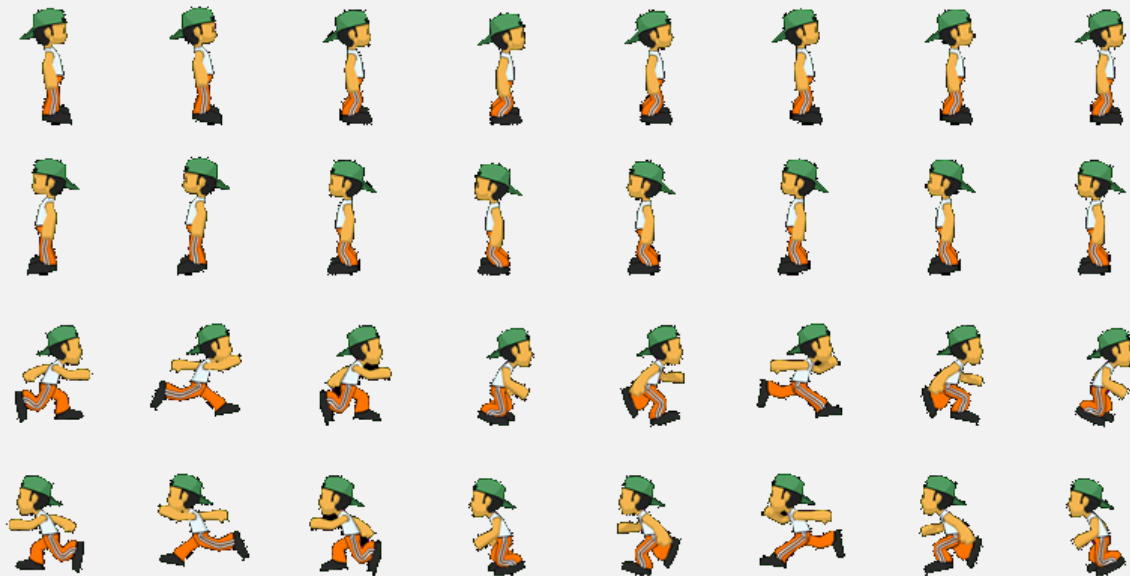
애니메이션(Animation)

■애니메이션이란?

- 여러 개의 이미지를 일정한 시간 간격을 통해서 화면에 뿌림으로써, 물체가 움직이는 효과를 주는 것.
- 스프라이트는 여러 개의 action으로 구성됨.
 - Action: 달리기, 걷기, 제자리 동작 등과 같이 캐릭터의 움직임을 나타냄.
 - Action은 여러 개의 Frame으로 구성됨.
 - Frame은 한 개의 이미지



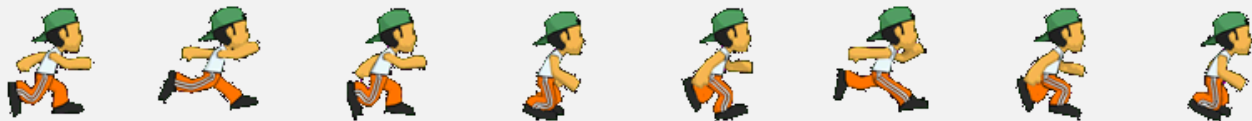
스프라이트 시트





캐릭터 애니메이션

run_animation.png



character_runs.py

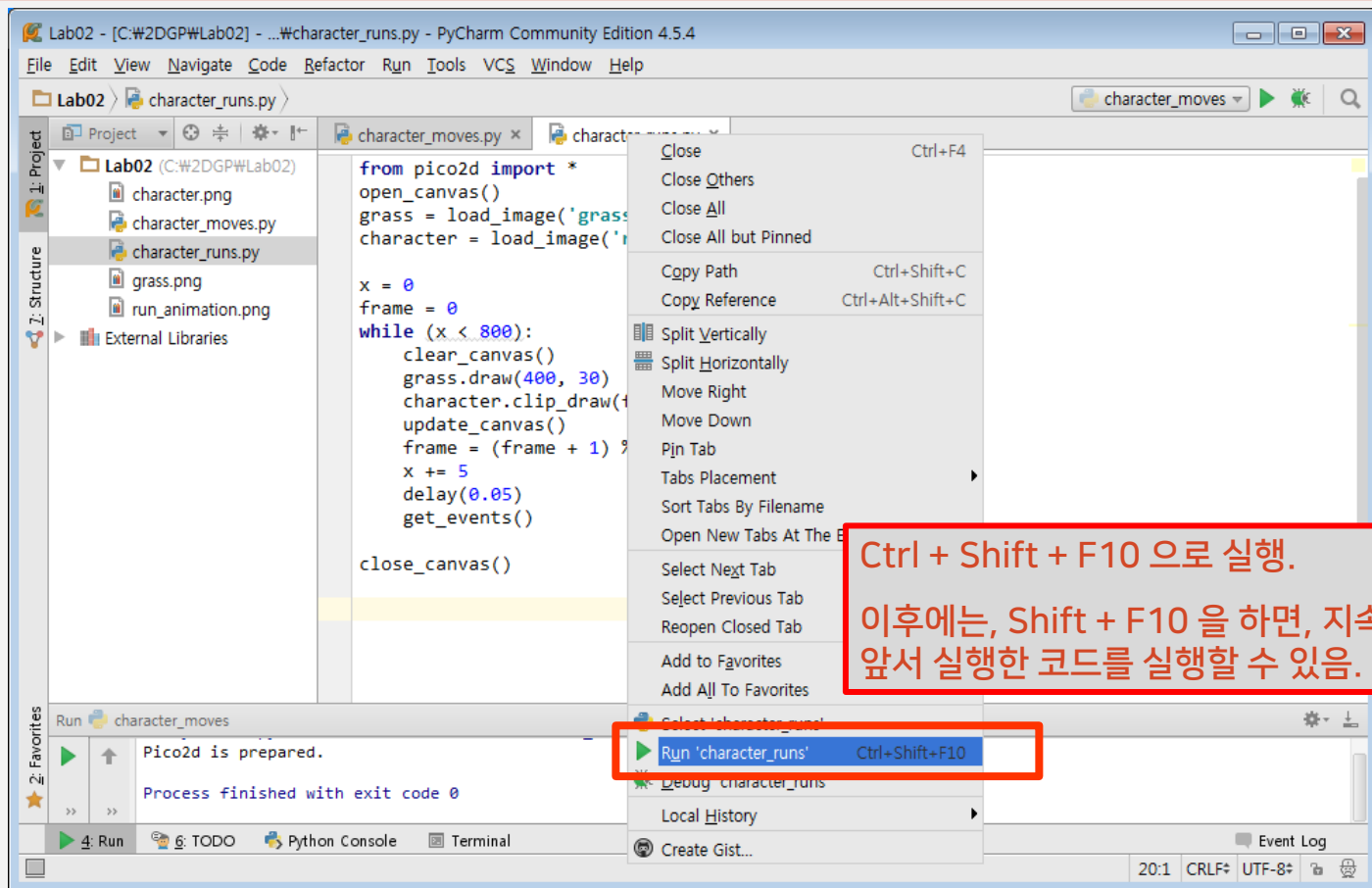


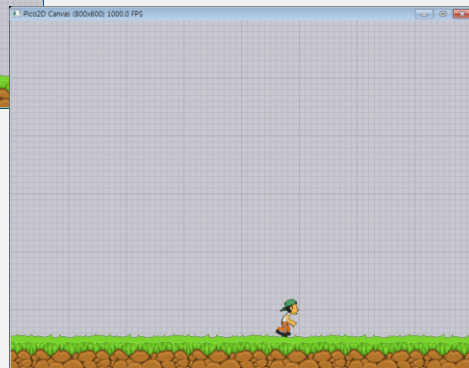
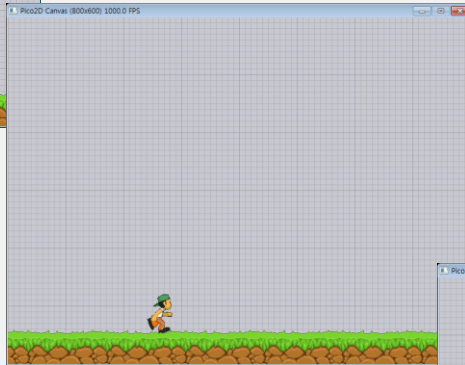
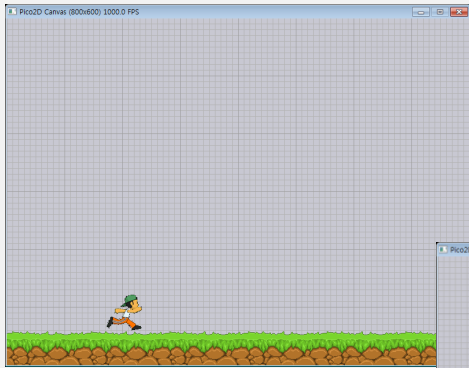
```
from pico2d import *
open_canvas()
grass = load_image('grass.png')
character = load_image('run_animation.png')

x = 0
frame = 0
while (x < 800):
    clear_canvas()
    grass.draw(400, 30)
    character.clip_draw(frame * 100, 0, 100, 100, x, 90)
    update_canvas()
    frame = (frame + 1) % 8
    x += 5
    delay(0.05)
    get_events()

close_canvas()
```

현재 Edit 중인 파일의 실행(Ctrl+Shift+F10)





clip_draw(left, bottom, width, height, x, y)

