

OpenGL 소개와 구현환경 구축

동명대학교 게임공학과







강영민

OpenGL

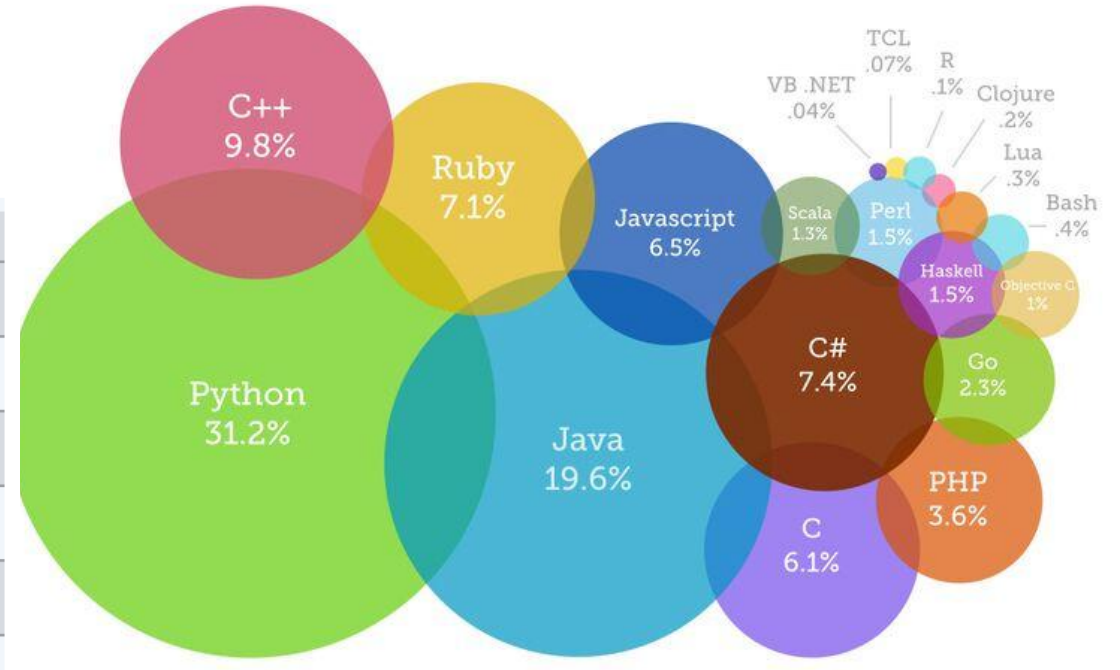
- 플랫폼 독립적인 그래픽스 라이브러리
 - 하드웨어나 운영체제 독립
 - GPU 활용 작업 지원
- 실시간 그래픽스 환경에서의 콘텐츠 생성이 주요 목적
 - 실사 수준 고품질 렌더링을 목표로 하지 않음
 - 최근의 하드웨어 발전에 따라 상당히 높은 수준의 렌더링 가능

프로그래밍 환경 - 언어

- 이 수업에서 사용할 언어
 - 파이썬(Python)

Jun 2023	Jun 2022	Change	Programming Language	Ratings	Change
1	1		 Python	12.46%	+0.26%
2	2		 C	12.37%	+0.46%
3	4	▲	 C++	11.36%	+1.73%
4	3	▼	 Java	11.28%	+0.81%
5	5		 C#	6.71%	+0.59%
6	6		 Visual Basic	3.34%	-2.08%

TIOBE 지수의 2023년 6월 순위표

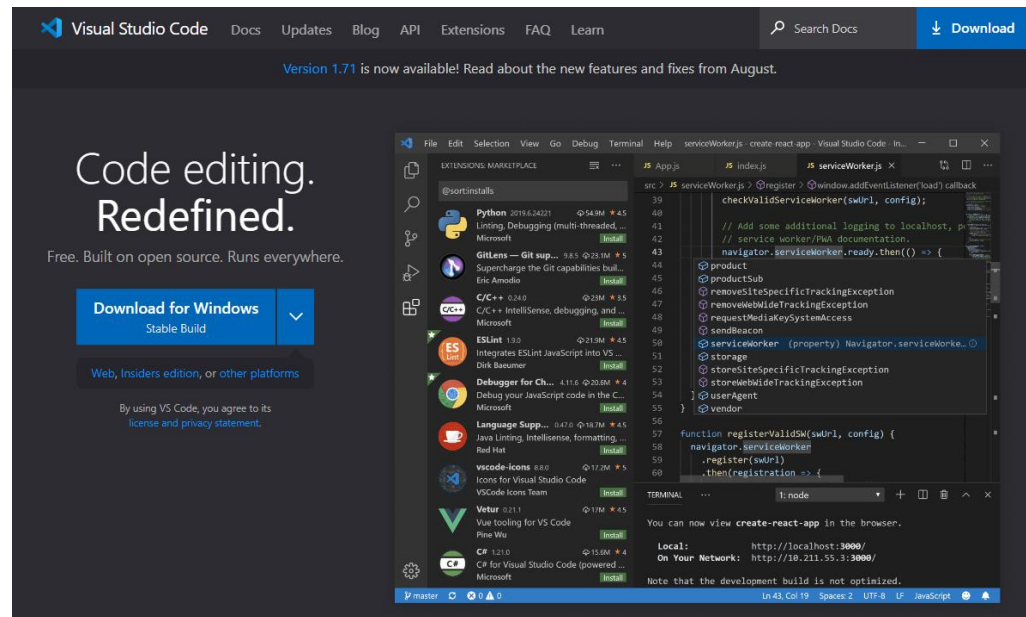


2019년 가장 트렌디한 프로그래밍 언어, 자료출처: <https://learnworthy.net/>

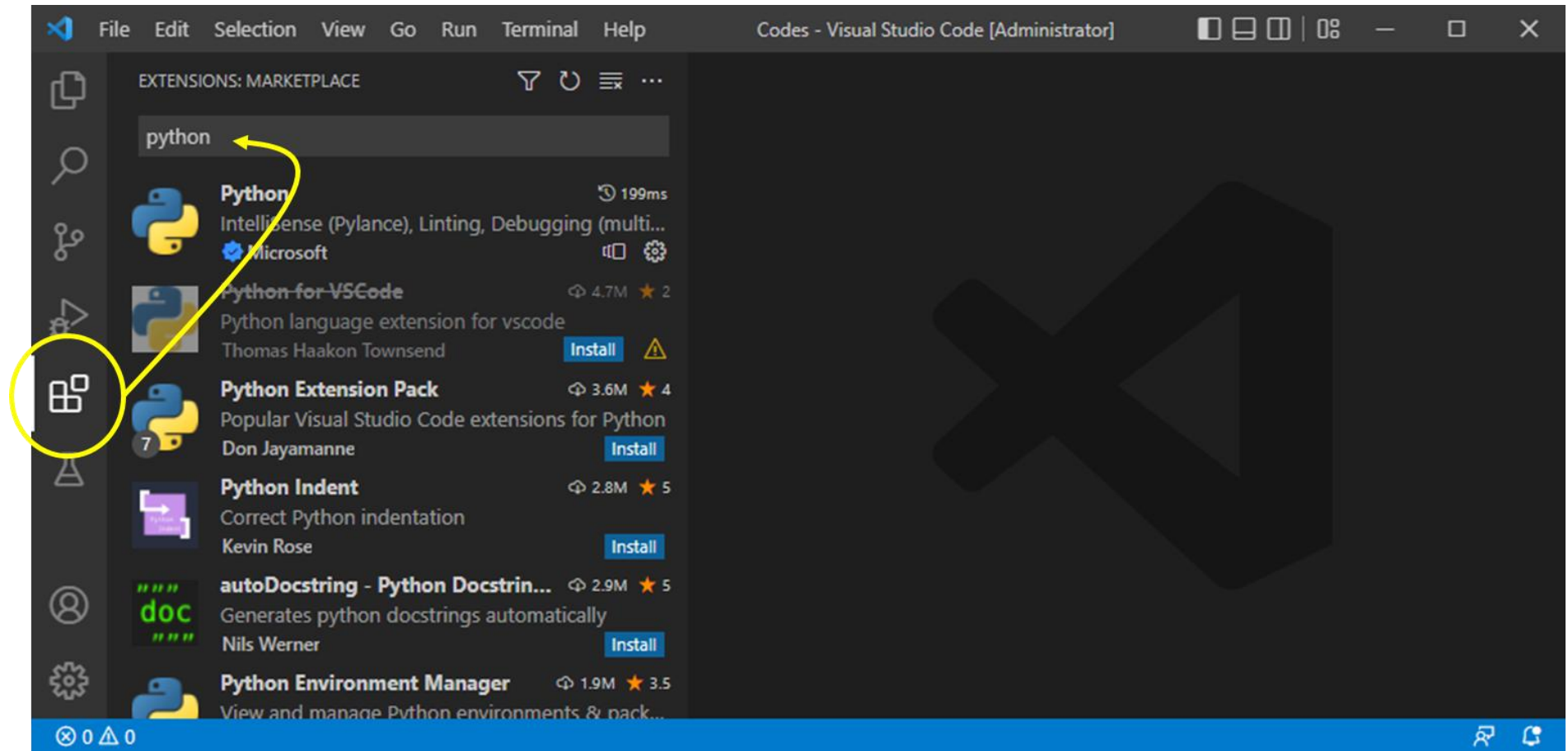
TIOBE is an acronym that stands for "The Importance Of Being Earnest." However, in the context of software development and programming, TIOBE typically refers to the TIOBE Index, which is a well-known index that ranks the popularity of programming languages based on their usage and availability.

Python 환경 구축

- 수업에서 사용할 환경
 - Python 재단의 인터프리터 설치
 - <https://www.python.org>
 - 코드 편집
 - Visual Studio Code
 - <https://code.visualstudio.com>
 - Microsoft의 무료 통합 개발 환경(IDE)
 - 다양한 언어 지원 - 강력한 코드 편집 - 디버깅
 - 확장 가능성
 - 버전 관리 지원
 - 테마 및 사용자 정의
 - 무료 및 오픈 소스

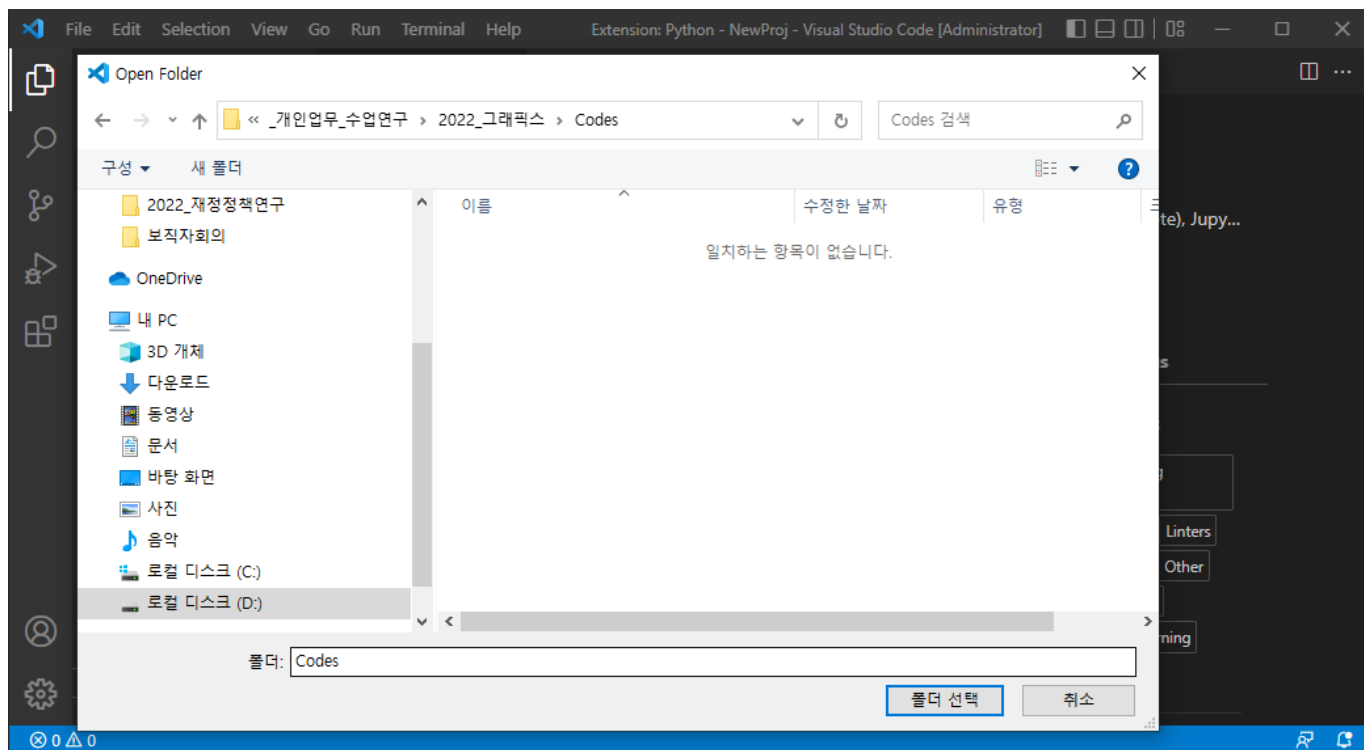


Python 확장 설치



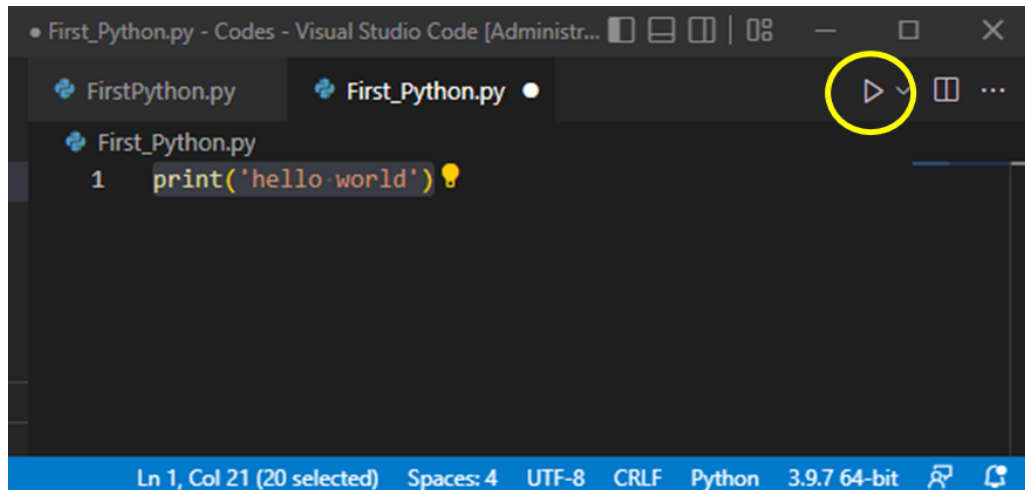
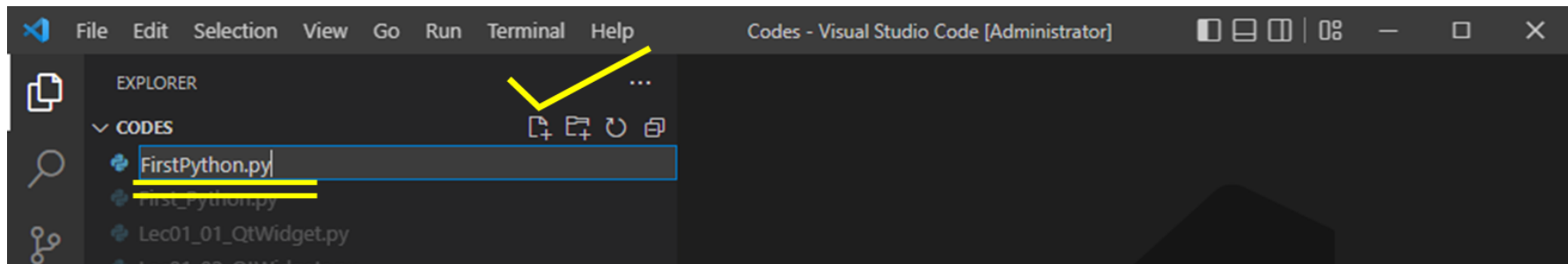
Python 연습

- 폴더를 만들어 프로그램 코드를 만들어 보자
 - File 메뉴의 Open Folder로 자신의 작업공간 지정



Python 연습

- 폴더 내에 파일을 만들어 코딩 연습



Python 연습

- 변수와 연산, 출력

```
# 파이썬 연습 - 이건 코멘트
```

```
# 변수를 사용하자
```

```
a = 3 # 처음으로 변수에 값이 저장될 때 선언이 알아서 이루어짐
```

```
b = 2 # 마찬가지로
```

```
c = a + b**3 ; # 세미콜론은 있든지 없든지...
```

```
print(c) # print는 내장함수
```


Python 연습

- 변수와 연산, 출력

```
a = '나는 정말 이상한 변수' # 변수에 다른 자료형이 들어가면 새롭게 선언됨
b = '나도 나도'

c = a + b
print(c)

a = 123434323
b = 234233423
c = a * b ** 100
print(c)
```

Python 연습

- 흐름제어

```
##### 프로그램의 흐름 제어 #####

age = int (input('나이를 입력하세요: '))
print(age , '세')

# 만약 18세 미만이면 출입금지를 출력하고
# 18세 이상이면 어떤 음료를 마실 것인지 묻는다

if age < 18 :
    print('출입 금지입니다')
elif age < 21:
    print('신분증을 보여주세요')
else :
    beverage = input('어떤 음료를 주문하시겠습니까? ')
    print('네, 알겠습니다.', beverage, '를 준비하겠습니다.')

### 반복 - for / while

for i in range(1, 10, 2) :
    print(i)

for i in [3, 4, 8, 9, 'hello'] :
    print(i)
```

Python 연습

- 리스트 데이터

```
presidents = ['이승만', '윤보선', '박정희', '최규하', '전두환', '노태우', '김영삼', '김대중', '노무현', '이명박', '박근혜', '문재인', '윤석열']
```

```
print(presidents)
```

```
for president in presidents:  
    print(president)
```

```
i = 0  
while i < 5 :  
    print(presidents[i])  
    i += 1
```

```
for i in range(5):  
    print(presidents[i])
```

Python 연습

- 함수

```
import math

def myFunction() :
    print('hahaha')

for i in range(10):
    myFunction()

def 빗변계산(밑변, 높이) :
    # 빗변을 계산하여 반환하자
    빗변 = math.sqrt(밑변**2 + 높이**2)
    return 빗변

result = 빗변계산(3, 4)
print(result)

while True: # 무한 루프
    밑변 = float (input('밑변은 얼마입니까?: '))
    높이 = float (input('높이는 얼마입니까?: '))

    if 밑변 < 0 or 높이 < 0 :
        print('종료합니다.')
        break

    빗변 = 빗변계산(밑변, 높이)
    print('이 삼각형의 빗변은 ', 빗변, '입니다.')
```

Python 학습

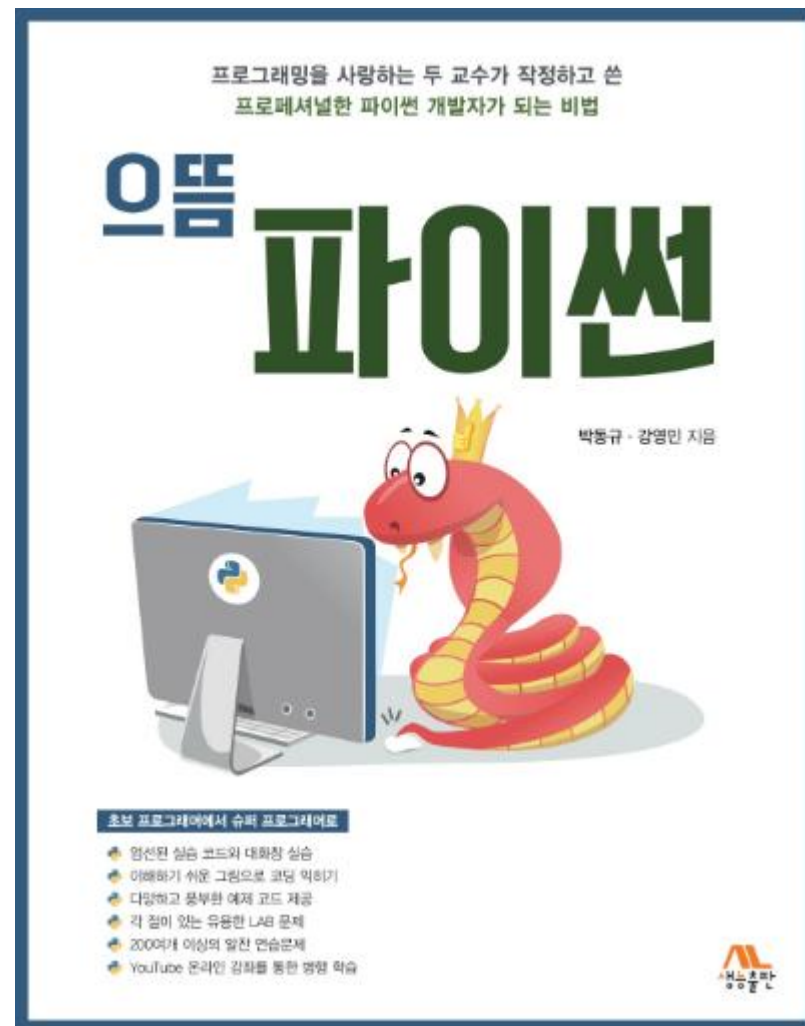
• 파이썬 자율학습

<https://github.com/dongupak/Prime-Python>



도움이 될만한 동영상 강좌

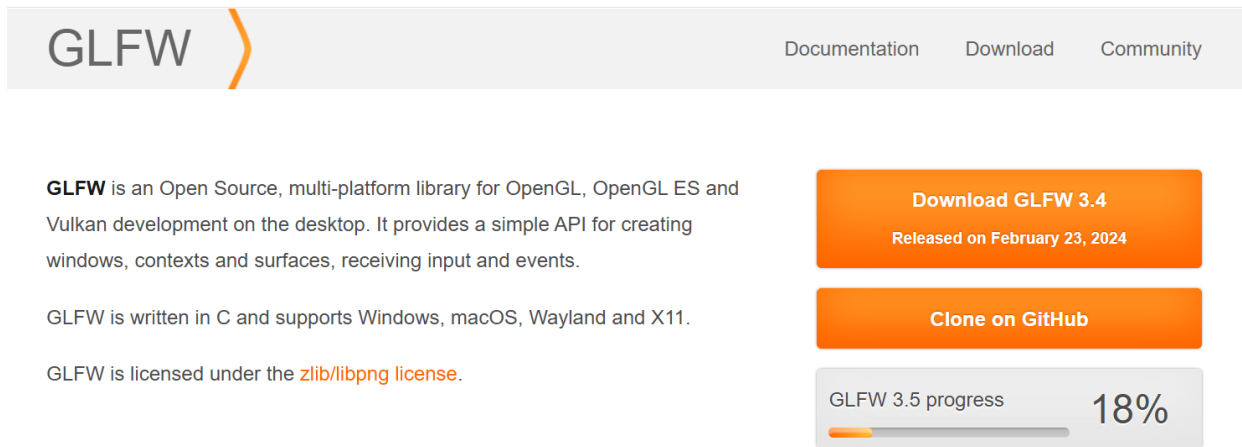
- 널널한 교수의 으뜸 파이썬 Youtube 강좌
 - <https://www.youtube.com/playlist?list=PL2P1Vm9k53HOflCAdyYaOx7hfBHydWNcL>
- 널널한 교수의 기초 파이썬 Youtube 강좌
 - <https://www.youtube.com/watch?v=TK6RLGaM3Ao&list=PL2P1Vm9k53HOEn-QYXAfbIQWkR0OW9mKX>
- 널널한 교수의 고급 파이썬 Youtube 강좌
 - https://www.youtube.com/watch?v=HYg_Zp1186M&list=PL2P1Vm9k53HMUI3i_UKRfWc3ePzAVerhc
- "따라하며 배우는 파이썬과 데이터 과학(2020 출판)" Youtube 강좌
 - https://www.youtube.com/playlist?list=PL2P1Vm9k53HOD_lesbnzAS51t71QeOPms



윈도우 관리

- glfw 사용

- GLFW(Graphics Library Framework)는 오픈 소스, 크로스 플랫폼 라이브러리
- OpenGL, OpenGL ES, Vulkan을 사용한 애플리케이션 개발을 위해 설계
- OpenGL 컨텍스트가 있는 창을 생성하고, 키보드, 마우스, 조이스틱 등 이벤트 처리
- 렌더링 관리를 하고, 감마 보정이나 클립보드 처리 같은 기능을 지원
- 간단한 API



The image shows the GLFW website header and main content area. The header has the GLFW logo on the left and navigation links for Documentation, Download, and Community on the right. The main content area contains a description of GLFW as an open-source, multi-platform library for OpenGL, OpenGL ES, and Vulkan development. It also includes two orange buttons: 'Download GLFW 3.4' (with a release date of February 23, 2024) and 'Clone on GitHub'. At the bottom, there is a progress bar for 'GLFW 3.5 progress' showing 18% completion.

GLFW

Documentation Download Community

GLFW is an Open Source, multi-platform library for OpenGL, OpenGL ES and Vulkan development on the desktop. It provides a simple API for creating windows, contexts and surfaces, receiving input and events.

GLFW is written in C and supports Windows, macOS, Wayland and X11.

GLFW is licensed under the [zlib/libpng license](#).

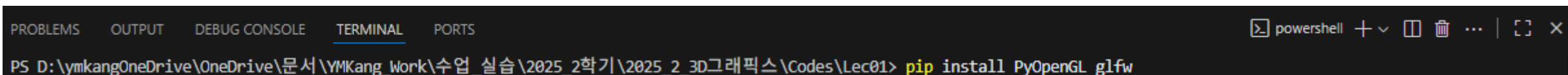
Download GLFW 3.4
Released on February 23, 2024

Clone on GitHub

GLFW 3.5 progress 18%

PyQt 설치

pip install PyOpenGL glfw



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + v [ ] [ ] ... | [ ] [X]  
PS D:\ymkangOneDrive\OneDrive\문서\YMKang_Work\수업_실습\2025_2학기\2025_2_3D그래픽스\Codes\Lec01> pip install PyOpenGL glfw
```

그래픽 윈도우 생성 프로그램

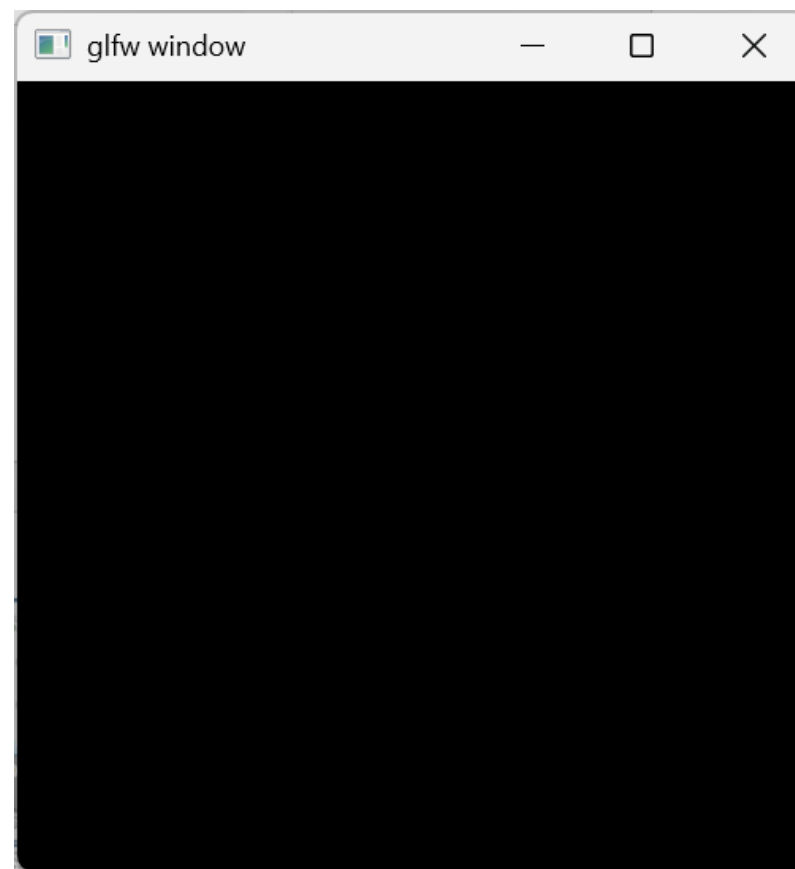
```
import glfw
from OpenGL.GL import *

def main():
    if not glfw.init():
        return
    window = glfw.create_window(500, 500, "glfw window", None, None)
    if not window:
        glfw.terminate()
        return
    glfw.make_context_current(window)

    while not glfw.window_should_close(window):
        glfw.swap_buffers(window)
        glfw.poll_events()

    glfw.destroy_window(window)
    glfw.terminate()

if __name__ == "__main__":
    main()
```



아무 문제가 발생하지 않는다고 생각하면

```
import glfw
from OpenGL.GL import *
```

```
def main():
    glfw.init():
    window = glfw.create_window(500, 500, "glfw window", None, None)
    glfw.make_context_current(window)
```

glfw.make_context_current(window)

makes the specified window's OpenGL context the current context for the calling thread. This means all subsequent OpenGL commands will affect this window. You must call this before issuing any OpenGL drawing commands.

```
while not glfw.window_should_close(window):
    # graphics code here
    glfw.swap_buffers(window)
    glfw.poll_events()
```

glfw.window_should_close(window)

is a function that returns True if the user has attempted to close the window, otherwise False.

```
glfw.destroy_window(window)
glfw.terminate()
```

```
if __name__ == "__main__":
    main()
```

glfw.poll_events()

processes all pending window events, such as keyboard input, mouse movement, and window close requests. It updates the window state and allows your program to respond to user actions. You should call it once per frame, usually inside your main loop.

그림을 그려볼까?

```
import glfw
from OpenGL.GL import *

def render_triangle():
    glClear(GL_COLOR_BUFFER_BIT)
    glBegin(GL_TRIANGLES)
    glColor3f(1.0, 0.0, 0.0) # Red
    glVertex2f(-0.5, -0.5)
    glColor3f(0.0, 1.0, 0.0) # Green
    glVertex2f(0.5, -0.5)
    glColor3f(0.0, 0.0, 1.0) # Blue
    glVertex2f(0.0, 0.5)
    glEnd()
```

OpenGL code

```
def main():
    if not glfw.init():
        return
    window = glfw.create_window(500, 500,
                                "OpenGL Triangle (Immediate Mode)", None, None)

    if not window:
        glfw.terminate()
        return
    glfw.make_context_current(window)

    while not glfw.window_should_close(window):
        render_triangle()
        glfw.swap_buffers(window)
        glfw.poll_events()

    glfw.destroy_window(window)
    glfw.terminate()

if __name__ == "__main__":
    main()
```

