Python 모듈을 c++에서 Import하기

1. Visual Studio에서 Pybind11과 Python 프로젝트 설정

새 프로젝트 → CMake Project

Pybind11 설치

```
pip install pybind11
```

CMake 설정

```
# 최소 CMake 버전 설정
cmake_minimum_required (VERSION 3.8)

# 프로젝트 이름 설정
project(CMakeProject1)

# C++ 표준 설정
set(CMAKE_CXX_STANDARD 20)

# Pybind11 헤더 경로 추가
include_directories(
    C:/Users/<사용자>/AppData/Local/Programs/Python/Python312/I
    C:/Users/<사용자>/AppData/Local/Programs/Python/Python312/L
)

# Python 라이브러리 경로 추가
link_directories(C:/Users/<사용자>/AppData/Local/Programs/Pytho

# 실행 파일 생성
add_executable(CMakeProject1 "CMakeProject1.cpp")

# Python 라이브러리 링크
target_link_libraries(CMakeProject1 python312)
```

C++ 코드 작성

```
#include <pybind11/embed.h>
namespace py = pybind11;

#include <iostream>

int main() {
    py::scoped_interpreter guard{};  // Python 인터프리터 초기화
```

```
    try {
        py::module_ my_script = py::module_::import("my_scrip

        // add 함수 호출
        int result = my_script.attr("add")(10, 20).cast<int>(
        std::cout << "Result of add: " << result << std::endl

        // greet 함수 호출
        std::string greeting = my_script.attr("greet")("World
        std::cout << "Result of greet: " << greeting << std::

    } catch (const py::error_already_set& e) {
        std::cerr << "Python error: " << e.what() << std::end
    }

    return 0;
}
```

Python 모듈

```
def add(a, b):
    return a + b

def greet(name):
    return f"Hello, {name}!"
```

최종 디렉토리 구조 예시

```
CMakeProject1/
  |- CMakeLists.txt
  |- CMakeProject1.cpp
  |- out/
      |- build/
          |- x64-Debug/
              |- CMakeProject1.exe
              |- my_script.py
```

실행 결과


Result of add: 30
Result of greet: Hello,World
C:\Users\MINKYU\source\repos\CMakeProject1\out\build\x64-Debug\CMakeProject1\CMakeProject1.exe(프로세스 31180개)이(가)
종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...