

第2集: CMakeLists.txt 基本结构与项目设置**

目标:

- 了解 CMakeLists.txt 文件的基本结构和常见指令
- 设置并构建一个简单的 CMake 项目

1. CMakeLists.txt 文件结构:

1. 最低版本要求:

```
cmake_minimum_required(VERSION 3.10)
```

- 指定 CMake 的最低版本要求，确保项目使用合适的 CMake 版本。

2. 项目声明:

```
project(MyProject)
```

- 使用 `project()` 来声明项目的名称，通常用于后续的配置。

3. 添加可执行文件:

```
add_executable(my_program main.cpp)
```

- `add_executable()` 是核心命令之一，用于定义可执行文件并指定其源文件。

2. 设置一个简单的 CMake 项目:

1. 项目结构:

```
MyProject/  
├─ CMakeLists.txt  
└─ main.cpp
```

2. 示例 CMakeLists.txt 文件:

```
cmake_minimum_required(VERSION 3.10)
project(MyProject)

# 添加一个可执行文件
add_executable(my_program main.cpp)
```

3. 示例 **main.cpp** 文件:

```
#include <iostream>

int main() {
    std::cout << "Hello, CMake!" << std::endl;
    return 0;
}
```

4. 使用 CMake 构建项目:

- 在项目目录下执行以下命令生成构建文件:

```
cmake .
```

- 然后使用 `make` 进行编译:

```
make
```

- 运行生成的可执行文件:

```
./my_program
```

3. 小结:

- **CMakeLists.txt** 文件是 CMake 项目的核心，用于描述如何编译和链接项目。
- 学会了使用基本的 CMake 指令创建并构建一个简单的 C++ 项目。