关键字：

cmake的关键字对大小不关心，但用户自定义变量和cmake内置变量必须区分大小写

project(工程名 VERSION 1.2.3 LANGUAGES C CXX)

1.其中的工程名只是作为标识，不具有任何作用，

2.(可选)VESION 1.2.3为工程版本号，可自定义，定义后可用

message(${PROJECT\_VERSION})查看版本号A.B.C

message(${PROJECT\_VERSION\_MAJOR})查看A

message(${PROJECT\_VERSION\_MINOR})查看B

message(${PROJECT\_VERSION\_PATCH})查看C

3.(可选)C CXX 指定语言类型,前有 VERSION, DESCRIPTION 或 HOMEPAGE\_URL 时，必须加 LANGUAGES 指定语言

set(变量名 变量内容1 变量内容2 ...):

1.一个变量名支持包含多个变量内容

2.变量内容可以是字符串，符号，也可以是另一个变量，如${另一个变量名}

3.cmake有内置变量，不同内置变量有不同的作用，有控制类型，指示类型等，具体参看 https://cmake.org/cmake/help/latest/manual/cmake-variables.7.html

unset(变量名):

删除变量

#将路径下的左右文件添加到变量中

aux\_source\_directory(路径 变量名)

#生成库文件

add\_library(库名 [STATIC SHARE] 源文件)

# 添加子CMakeLists.txt路径(指包含CMakeLists.txt的路径)

add\_subdirectory(路径1 路径2 ...)

# 添加头文件路径

include\_directories(路径1 路径2 ...)

# 变量追加内容

list(APPEND 变量名 值)

常用系统定义变量：

CMAKE\_CURRENT\_SOURCE\_DIR : 当前CMakeLists.txt所在路径

message(输出类型 "输出字符串" ${SRC\_LIST})

输出类型：必须大写

(无) = 重要消息;

STATUS = 非重要消息；

WARNING = CMake 警告, 会继续执行；

AUTHOR\_WARNING = CMake 警告 (dev), 会继续执行；

SEND\_ERROR = CMake 错误, 继续执行，但是会跳过生成的步骤；

FATAL\_ERROR = CMake 错误, 终止所有处理过程；

configure\_file(输入文件.in 输出文件)

例：configure\_file(config.in config.h)，将.in文件按某种规则转换为.h文件

1.in文件中有 @变量名@ 或 ${变量名} 的位置，会被 CMakeLists.txt 定义的值替换，然后生成到.h

2.in文件中有 #cmakefefine 变量名 ${变量名}或@变量名@的地方，如果在CMakeLists.txt中未定义或定义为具有假含义的内容(0,no,false等)，则不会生成对应的宏，如果有定义，则替换为对应的值，然后生成到.h

${变量名}：为取变量内容，在if条件判断中，可不加${}

():小括号为指令参数列表，其中可用空格或分号分隔

add\_executable(生成的目标文件名 依赖文件1.c 依赖文件2.c .....)

1.目标文件名不带后缀

条件语句

if()

elseif()

else

endif()

函数语句

定义：

function(函数名 参数1 参数2)

......

endfunction()

调用：

函数名(${参数1} ${参数2})

https://www.bilibili.com/video/BV1nU4y1B7mJ/?p=3&spm\_id\_from=pageDriver&vd\_source=41b642be40ce1ad94fe112cebcb48a3c