# 1，可以在命令行程序里面定义windows API

## 1）新建一个控制台工程，取名：console-load-win-api，然后添加下面的代码

|  |
| --- |
|  |

### 编译运行，效果如下

|  |
| --- |
|  |

### 其实很多语言都可以调用windows API函数

## 2）我们把main.c改为main.cpp,那么她就是一个cpp程序，我们把它修改一下，代码不修改，只修改文件名

|  |
| --- |
|  |

### 编译运行程序，可以通过编译，但是有一个警告，说cpp的main函数的返回值是int而不是void，程序也是可以再次运行。

|  |
| --- |
|  |

## 3）我们把main函数的返回值改为int，并且让他符号0，并且修改运行头文件

|  |
| --- |
|  |

### 编译运行，效果如下：

|  |
| --- |
|  |

## 4）然后我们尝试把main函数改为WinMain函数

|  |
| --- |
|  |

### 编译程序，会出现一个链接错误，说找不到main

|  |
| --- |
|  |

## 5）在项目上面点击右键-》属性-》链接器-系统，把控制台 (/SUBSYSTEM:CONSOLE)改为：窗口 (/SUBSYSTEM:WINDOWS)，点击确定

|  |
| --- |
|  |

### 然后编译，就能够成功，运行程序，效果如下

|  |
| --- |
|  |

### 因为我们并没有创建窗口，它就没有窗口，但是它仍然是一个应用程序，仍然能够调用windows API

# 2.其实我们主要是学习Window应用程序开发，

## 1>我们可以直接创建一个win32项目，如我们新建一个windows sdk项目，如图

|  |
| --- |
|  |

## 2>.可以看到开发工具默认帮我们生成了很多基础代码，我们真正需要学习的就是对windows消息的响应操作，如这里我们添加对鼠标左键按下的响应代码

|  |
| --- |
|  |

### 编译程序程序，效果如下

|  |
| --- |
|  |

# 3.我们也可以创建MFC项目

## 1）我们新建一个mfc，项目如下，注意用较新版本的vs创建mfc项目，在用户界面功能选项卡里面有一些选项，不要勾选，而且在命令行(菜单/工具栏/功能区)选项里面选择经典菜单

|  |
| --- |
|  |

## 在高级选项里面有一些新窗口我们暂时用不到，可以取消勾选，

|  |
| --- |
|  |

### 然后点击完成，

|  |
| --- |
|  |

## 2）编译运行项目，效果如下，这就是经典应用界面

|  |
| --- |
|  |

## 注意：MFC里面的API都是类的成员函数，他们和win32 sdk的API的名字相同都是他们的参数个数都比sdk平台的简单，有时候参数个数少很多然后MessageBox比平台sdk少3个等等。在MFC中也可以使用平台sdk函数，但是响应在前面添加::,如图

|  |
| --- |
|  |

### 上面那个调用就是平台sdk的函数。还有一个Afx开头的是框架全局函数，也是一种包装。

|  |
| --- |
|  |

## 3）.我们给我们的view类添加一个鼠标左键按下的消息响应函数，在里面使用AfxMessageBox函数

|  |
| --- |
|  |

### 编译运行，效果如下

|  |
| --- |
|  |

# 4.window API有多少个？恐怕比你想象的还要多。有1万个以上，改变没有可能全部都去学习，用到了，可以去查询子类如msdn或者在线帮助