

# 什么是Windows API编程

- 应用编程接口
- Application Programming Interface

## WinMain函数

当Windows操作系统启动一个程序时，它调用的就是该程序的WinMain函数（实际是由插入到可执行文件中的启动代码调用的）。WinMain是Windows程序的入口点函数，与DOS程序的入口点函数main的作用相同，当WinMain 函数结束或返回时，Windows应用程序结束。

```
1  int WINAPI WinMain(  
2      HINSTANCE hInstance,           //应用程序实例  
3      HINSTANCE hPrevInstance,       //上一个应用程序实例  
4      LPSTR lpCmdLine,               //命令行参数  
5      int nShowCmd                   //窗口显示的样式  
6  );
```

- **WINAPI**：是一个宏，它代表的是 `__stdcall`（注意是两个下划线），表示的是参数传递的顺序：从右往左入栈，同时在函数返回前自动清空堆栈。
- **hInstance**：表示该程序当前运行的实例的句柄，这是一个数值。当程序在Windows下运行时，它唯一标识运行中的实例（注意，只有运行中的程序实例，才有实例句柄）。一个应用程序可以运行多个实例，每运行一个实例，系统都会给该实例分配一个句柄值，并通过hInstance参数传递给WinMain 函数。
- **hPrevInstance**：表示当前实例的前一个实例的句柄。在Win32环境下，这个参数总是NULL，即在Win32环境下，这个参数不再起作用。
- **lpCmdLine**：是一个以空终止的字符串，指定传递给应用程序的命令行参数，相当于C或C++中的main函数中的参数 `char *argv[]`。
- **nShowCmd**：表示一个窗口的显示，表示它是要最大化显示、最小化显示、正常大小显示还是隐藏显示。

```
1  #include <windows.h>  
2  #include <cstdio>  
3  #pragma comment(linker, "/subsystem:\"console\""  
    "/entry:\"WinMainCRTStartup\"")  
4  
5  int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR  
    lpCmdLine, int nCmdShow) {  
6  
7      printf("%p %p %p\n", hInstance, hPrevInstance, GetModuleHandleA(NULL));  
8      printf("%d %d\n", nCmdShow, SW_SHOWNORMAL);  
9  
10     printf("%s\n", lpCmdLine);  
11  
12     MessageBoxA(NULL, TEXT("第一个 windows API 程序"), TEXT("黑猫编程"),  
        MB_OK);
```

```
13  
14     return 0;  
15 }
```

## 字符集

---

```
1  #include <windows.h>  
2  #include <stdio>  
3  #include <cstring>  
4  #include <locale>  
5  
6  int main() {  
7  
8      const char* str1 = "Abc中国";  
9      printf("%s %d\n", str1, strlen(str1));  
10  
11     _wsetlocale(LC_ALL, L"chs");  
12     const wchar_t* str2 = L"ABC中国文字";  
13     wprintf(L"%s %d\n", str2, wcslen(str2));  
14  
15     MessageBoxW(NULL, TEXT("hello cat."), L"coding", MB_OK);  
16  
17     return 0;  
18 }
```