GetStdHandle是一个Windows [API](https://so.csdn.net/so/search?q=API&spm=1001.2101.3001.7020)函数

HANDLE WINAPI GetStdHandle( \_In\_ DWORD nStdHandle);



可以返回一个HANDLE

包含在头文件 <windows.h>之中

#include <windows.h>

#include <stdio.h>

#include <string.h>

int main(int argc,char \* argv[])

{

HANDLE fd = GetStdHandle(STD\_OUTPUT\_HANDLE);//这里，获取标准输出设备的句柄，并将其赋给句柄变量fd

CONSOLE\_CURSOR\_INFO cinfo;

char str[4];

while(1)

{

cinfo.dwSize = 1;

printf("请输入on/off开启/关闭光标显示:\n");

scanf("%s",str);

if(!strcmp(str,"off"))

{

cinfo.bVisible = 0;

SetConsoleCursorInfo(fd,&cinfo);

}

else if(!strcmp(str,"on"))

{

cinfo.bVisible = 1;

SetConsoleCursorInfo(fd,&cinfo);

}

else

printf("指令错误，请重新输入，输入on或者off打开或者关闭光标显示\n");

}

return 0;

}

（1）HANDLE是一个句柄类 用来控制api的信息

（2）CONSOLE\_CURSOR\_INFO是一个光标信息类

typedef struct \_CONSOLE\_CURSOR\_INFO {

DWORD dwSize; 用来控制光标的大小 从1 到 100

BOOL bVisible;用来控制光标的有无 0 为无

} CONSOLE\_CURSOR\_INFO, \*PCONSOLE\_CURSOR\_INFO;

（１）设置光标的属性函数 SetConsoleCursorInfo

BOOL WINAPI SetConsoleCursorInfo( \_In\_ HANDLE hConsoleOutput, \_In\_ const CONSOLE\_CURSOR\_INFO \*lpConsoleCursorInfo ); 功能： 设置光标的属性 参数： hConsoleOutput 控制台屏幕缓冲区的句柄。句柄必须具有GENERIC\_READ访问权限。 lpConsoleCursorInfo 指向CONSOLE\_CURSOR\_INFO结构的指针，该结构为控制台屏幕缓冲区的游标提供新规范。 返回值： 如果函数成功，则返回值为非零值。 如果函数失败，则返回值为零。要获取扩展错误信息，请调用GetLastError。

（２）设置光标的位置函数 SetConsoleCursorPosition

BOOL WINAPI SetConsoleCursorPosition( \_In\_ HANDLE hConsoleOutput, \_In\_ COORD dwCursorPosition ); 功能： 设置光标的位置 参数： hConsoleOutput 控制台屏幕缓冲区的句柄。句柄必须具有GENERIC\_READ访问权限。 dwCursorPosition 用于指定新的光标位置（以字符为单位）。坐标是屏幕缓冲区字符单元格的列和行。坐标必须位于控制台屏幕缓冲区的边界内。 返回值： 如果函数成功，则返回值为非零值。 如果函数失败，则返回值为零。要获取扩展错误信息，请调用GetLastError。

参考代码

int main(void)

{

//定义句柄变量

HANDLE hOut = NULL;

COORD pos;

//获取标准输出句柄

hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

pos.X = 8;

pos.Y = 8;

//设置光标的位置

SetConsoleCursorPosition(hOut, pos);

printf("hello itcast1");

pos.X = 18;

pos.Y = 16;

//设置光标的位置

SetConsoleCursorPosition(hOut, pos);

printf("hello itcast2");

//关闭句柄

CloseHandle(hOut);

system("pause");

return 0;

}

设置控制台的字体颜色

（３）SetConsoleTextAttribute()函数是一个API设置字体颜色和背景色的函数。参数表中使用两个属性（属性之间用，隔开）。GetStdHandle()和FOREGROUND\_\*或BACKGROUND\_\*。\*值为INTENSITY或RED或GREEN或BLUE。第一个属性获得句柄（即要设置颜色的地方），第二个属性设置颜色。属性相加是在属性值中间加“|”隔开即可。

调用例子：

HANDLE hOut;

hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hOut,

FOREGROUND\_RED);

cout << "Red " << flush;

SetConsoleTextAttribute(hOut,

FOREGROUND\_RED |

FOREGROUND\_INTENSITY);

cout << "Red" << endl;

颜色的设置包括

ＦＯＲＥＧＲＯＵＮＤ＿ＣＯＬＯＲ　前景颜色　也就是字体

ＢＡＣＫＧＲＯＵＮＤ＿ＣＯＬＯＲ　后景颜色　也就是背景

ＦＯＲＥＧＲＯＵＮＤ＿ＩＮＴＥＮＳＩＴＹ　前景高光　将字体颜色拔亮

ＢＡＣＫＧＲＯＵＮＤ＿ＩＮＴＥＮＳＩＳＴ　后景高光



也可以直接带入数值　但代码不清晰