Colorizing text in the console with C++

Asked 13 years, 2 months ago Modified 3 months ago Viewed 455k times

156

How can I write colored text to the console with C++? That is, how can I write different text with different colors?

c++ colors colorize

Share Edit Follow

edited May 2, 2016 at 12:09
anon

asked Oct 29, 2010 at 16:20
Sudantha
15.8k 44 105 161

Did but no resources to find *different colors in the C++ console* :) – Sudantha Oct 29, 2010 at 16:24

6 What's the C++ console?? – Edward Strange Oct 29, 2010 at 16:30

Possible duplicate of: stackoverflow.com/questions/3585846/... – karlphillip Oct 29, 2010 at 16:38

Be careful of these answers. Your console will have the changes after the program exits if you don't revert them yourself. – kayleeFrye_onDeck Jul 11, 2017 at 19:21

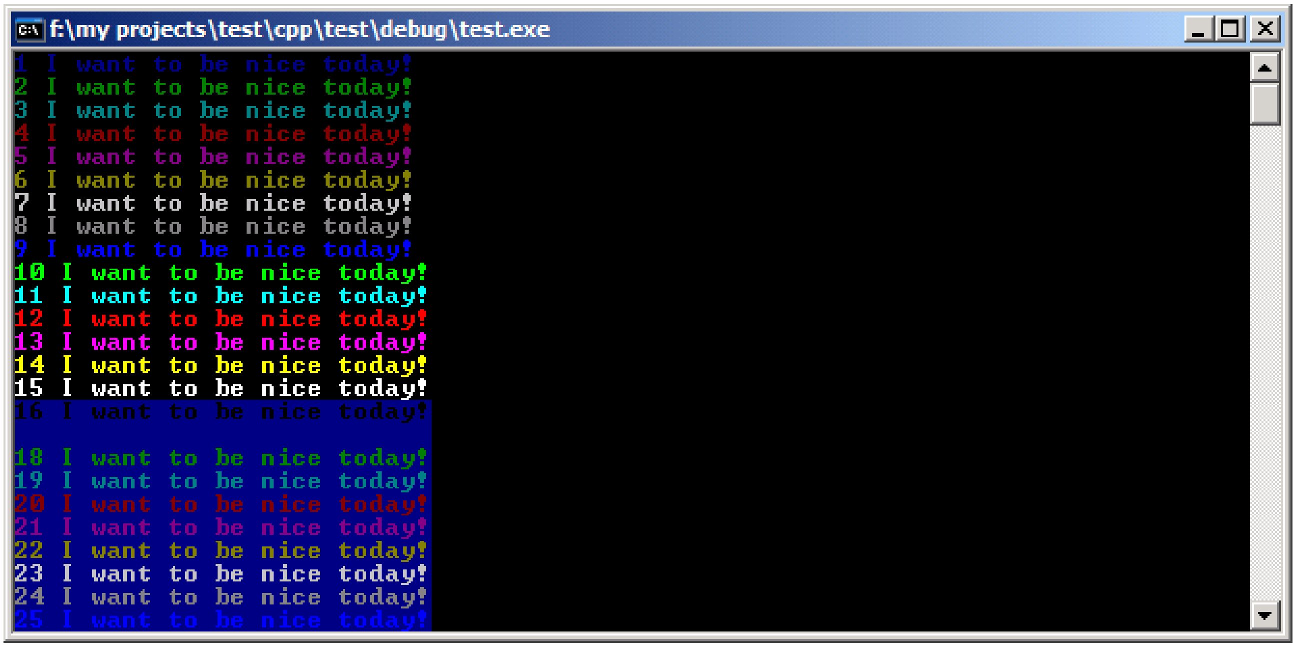
15 Answers

Sorted by: Highest score (default)

179

Add a little Color to your Console Text

```
HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
// you can loop k higher to see more color choices
for(int k = 1; k < 255; k++)
{
  // pick the colorattribute k you want
  SetConsoleTextAttribute(hConsole, k);
  cout << k << " I want to be nice today!" << endl;
}
```



```
1 I want to be nice today!
2 I want to be nice today!
3 I want to be nice today!
4 I want to be nice today!
5 I want to be nice today!
6 I want to be nice today!
7 I want to be nice today!
8 I want to be nice today!
9 I want to be nice today!
10 I want to be nice today!
11 I want to be nice today!
12 I want to be nice today!
13 I want to be nice today!
14 I want to be nice today!
15 I want to be nice today!
16 I want to be nice today!

18 I want to be nice today!
19 I want to be nice today!
20 I want to be nice today!
21 I want to be nice today!
22 I want to be nice today!
23 I want to be nice today!
24 I want to be nice today!
25 I want to be nice today!
```

Character Attributes Here is how the "k" value be interpreted.

Share Edit Follow

edited May 11, 2018 at 22:21

answered Oct 29, 2010 at 16:26

Sheen

3,343 5 27 47

85 Note that this is Windows-only. – DarkDust Oct 29, 2010 at 16:27

4 @Misaki I haven't tested but can you try removing the 'endl' bit? – Sheen Jan 10, 2017 at 13:59

3 What library I need to include for knowing HANDLE class? – STF Mar 16, 2017 at 5:54

4 @STF just use <windows.h> – Sheen Mar 16, 2017 at 10:06

3 how to set it back to default? – Beyondo Apr 27, 2018 at 12:49

101

ANSI escape color codes :

```
Name            FG  BG
Black           30  40
Red             31  41
Green           32  42
Yellow          33  43
Blue            34  44
Magenta         35  45
Cyan            36  46
White           37  47
Bright Black    90  100
Bright Red      91  101
Bright Green    92  102
Bright Yellow   93  103
Bright Blue     94  104
Bright Magenta  95  105
Bright Cyan     96  106
Bright White    97  107
```

Sample code for C/C++ :

```
#include <iostream>
#include <string>

int main(int argc, char ** argv){

    printf("\n");
    printf("\x1B[31mTexting\033[0m\t\t");
    printf("\x1B[32mTexting\033[0m\t\t");
    printf("\x1B[33mTexting\033[0m\t\t");
    printf("\x1B[34mTexting\033[0m\t\t");
    printf("\x1B[35mTexting\033[0m\n");

    printf("\x1B[36mTexting\033[0m\t\t");
    printf("\x1B[36mTexting\033[0m\t\t");
    printf("\x1B[36mTexting\033[0m\t\t");
    printf("\x1B[37mTexting\033[0m\t\t");
    printf("\x1B[93mTexting\033[0m\n");

    printf("\033[3;42;30mTexting\033[0m\t\t");
    printf("\033[3;43;30mTexting\033[0m\t\t");
    printf("\033[3;44;30mTexting\033[0m\t\t");
    printf("\033[3;104;30mTexting\033[0m\t\t");
    printf("\033[3;100;30mTexting\033[0m\n");

    printf("\033[3;47;35mTexting\033[0m\t\t");
    printf("\033[2;47;35mTexting\033[0m\t\t");
    printf("\033[1;47;35mTexting\033[0m\t\t");
    printf("\t\t");
    printf("\n");

    return 0;
}
```

GCC :

```
g++ cpp_interactive_terminal.cpp -o cpp_interactive_terminal.cgi
chmod +x cpp_interactive_terminal.cgi
./cpp_interactive_terminal.cgi
```

Share Edit Follow

edited Aug 25, 2020 at 9:17
thetek
3 4

answered Jan 6, 2019 at 15:06
Me Mehdi Mohammadpour
Moha **1,070** 1 7 11

14 @Mehdi Mohammadpour I have Windows 10, and your escape codes *are* working for me, but it appears to me that you've got the Foreground and Background codes reversed. Can you please reverse the

headings 'FG' and 'BG' please? – yamex5 Mar 20, 2019 at 1:32

I have to agree with @yamex5 – smoothware Jan 2, 2020 at 22:19

1 Thanks, Worked with Visual C++ in Win10 – Prabath Jan 1, 2022 at 15:48

4 Espace sequences are basically for Linux. This will mostly not work with Windows – MrFox Mar 6, 2022 at 16:50 ✎

1 In windows, you might need to enable the Virtual Terminal mode to use those color codes, checkout learn.microsoft.com/en-us/windows/console/... – EdgeNeko Dec 1, 2022 at 16:42

35

Standard C++ has no notion of 'colors'. So what you are asking depends on the operating system.

For Windows, you can check out the SetConsoleTextAttribute function.

On *nix, you have to use the ANSI escape sequences.

Share Edit Follow

answered Oct 29, 2010 at 16:26

user225312
128k 69 174 181

6 Note that ASNI sequences (or at least some of them) work on Windows' CMD.EXE as well. – anon May 4, 2016 at 18:26

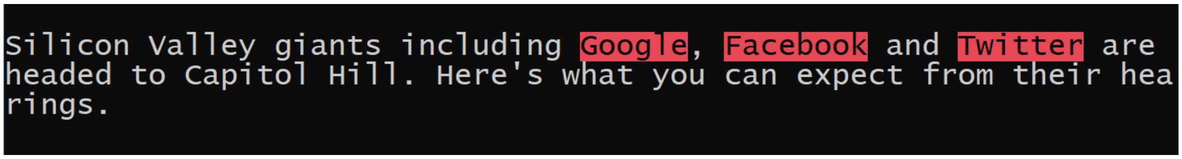
4 @Asu, they don't before Win10 Anniversary Edition. If your Win10 us up-to-date, you have it. – kayleeFrye_onDeck Jul 11, 2017 at 18:30

22

I've found header-only open-source C++ library working for Windows https://github.com/imfl/color-console

Color Console:

> A lightweight header-only C++ library to bring colors to your Windows console with a very-easy-to-use API that frees you from the burden of setting and resetting screen colors every time you make a call.

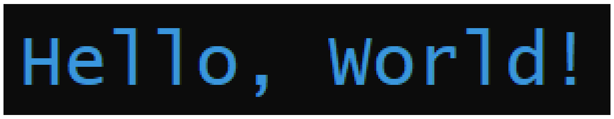


```
Silicon Valley giants including Google, Facebook and Twitter are
headed to Capitol Hill. Here's what you can expect from their hea
rings.
```

```
#include "../include/color.hpp"
#include <iostream>
```

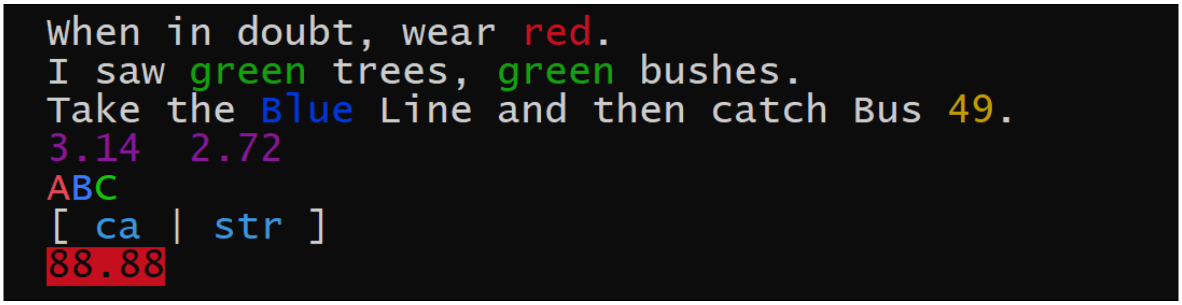
```
int main() {
    std::cout << dye::aqua("Hello, World!") << std::endl;
    return 0; }
```

You are seeing Hello, World! in aqua.



```
Hello, World!
```

Why Use It?



```
When in doubt, wear red.
I saw green trees, green bushes.
Take the Blue Line and then catch Bus 49.
3.14  2.72
ABC
[ ca | str ]
88.88
```

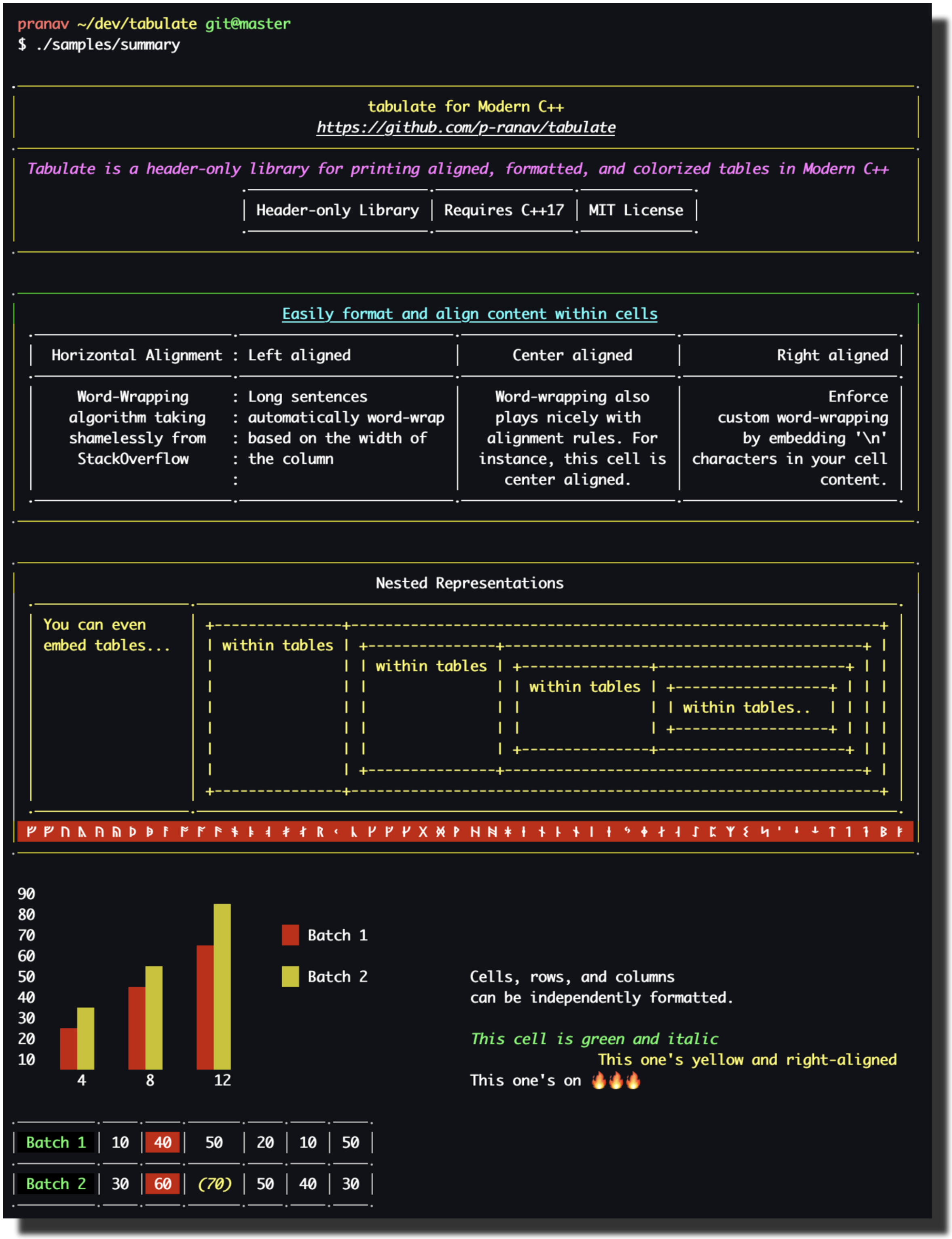
> No need to reset: most solutions on the market work like manipulators, which constantly require you to reset the screen color after you set it. While this traditional approach is also offered in this library in the hue namespace ...

tabulate:

If You want not only to change colors but print text in more readable form (e.g. in form of tabular) it is also https://github.com/p-ranav/tabulate which can change colors and draw tables in console.

> tabulate is a header-only library. Just add include/ to your include_directories and you should be good to go. A single header file version is also available in single_include/.

NOTE Tabulate supports >=C++11.



```
pranav ~/dev/tabulate git@master
$ ./samples/summary

.------------------------------------------------------------------------------------------.
|                                 tabulate for Modern C++                                  |
|                          https://github.com/p-ranav/tabulate                             |
.------------------------------------------------------------------------------------------.
| Tabulate is a header-only library for printing aligned, formatted, and colorized tables in Modern C++ |
|                   .-----------------------.---------------.-------------.                 |
|                   | Header-only Library | Requires C++17 | MIT License |                  |
|                   .-----------------------.---------------.-------------.                 |
.------------------------------------------------------------------------------------------.


.------------------------------------------------------------------------------------------.
|                       Easily format and align content within cells                       |
| .----------------------.-----------------------.----------------------.---------------------. |
| | Horizontal Alignment : Left aligned          |    Center aligned    |       Right aligned | |
| .----------------------.-----------------------.----------------------.---------------------. |
| |    Word-Wrapping     : Long sentences        |  Word-wrapping also  |             Enforce | |
| |   algorithm taking   : automatically word-wrap |   plays nicely with  |  custom word-wrapping | |
| |   shamelessly from   : based on the width of |  alignment rules. For |   by embedding '\n' | |
| |    StackOverflow     : the column            | instance, this cell is | characters in your cell | |
| |                      :                       |    center aligned.   |            content. | |
| .----------------------.-----------------------.----------------------.---------------------. |
.------------------------------------------------------------------------------------------.


.------------------------------------------------------------------------------------------.
|                                  Nested Representations                                  |
| .-----------------.----------------------------------------------------------------------. |
| | You can even    | +---------------+------------------------------------------------------+ | |
| | embed tables... | | within tables | +---------------+-------------------------------------+ | | |
| |                 | |               | | within tables | +---------------+---------------------+ | | | |
| |                 | |               | |               | | within tables | +-----------------+ | | | |
| |                 | |               | |               | |               | | within tables.. | | | | |
| |                 | |               | |               | |               | +-----------------+ | | | |
| |                 | |               | |               | +---------------+---------------------+ | | |
| |                 | |               | +---------------+-------------------------------------+ | |
| |                 | +---------------+------------------------------------------------------+ |
| .-----------------.----------------------------------------------------------------------. |
| ᚠ ᚡ ᚢ ᚣ ᚤ ᚥ ᚦ ᚧ ᚨ ᚩ ᚪ ᚫ ᚬ ᚭ ᚮ ᚯ ᚰ ᚱ ᚲ ᚳ ᚴ ᚵ ᚶ ᚷ ᚸ ᚹ ᚺ ᚻ ᚼ ᚽ ᚾ ᚿ ᛀ ᛁ ᛂ ᛃ ᛄ ᛅ ᛆ ᛇ ᛈ ᛉ ᛊ ᛋ ᛌ ᛍ ᛎ ᛏ ᛐ ᛑ ᛒ ᛓ |
.------------------------------------------------------------------------------------------.

90
80
70                          🟥 Batch 1
60
50                          🟨 Batch 2            Cells, rows, and columns
40                                                can be independently formatted.
30
20                                                This cell is green and italic
10                                                             This one's yellow and right-aligned
       4        8        12                       This one's on 🔥🔥🔥

.---------.----.----.------.----.----.----.
| Batch 1 | 10 | 40 |  50  | 20 | 10 | 50 |
.---------.----.----.------.----.----.----.
| Batch 2 | 30 | 60 | (70) | 50 | 40 | 30 |
.---------.----.----.------.----.----.----.
```

Share Edit Follow

edited Sep 28, 2023 at 11:43

congusbongus
13.8k 8 74 100

answered Mar 6, 2021 at 22:44

baziorek
2,542 2 29 45

2 These are both very good. thanks a lot. the bad thing though is the first one is windows specific only.
– Hossein Apr 6, 2021 at 7:59

4 Last two days I've found better multi-platform library for colors: github.com/jupyter-xeus/cpp-terminal It is being developed by Jupyter-xeus - people from CERN. – baziorek Apr 6, 2021 at 8:51

13

You can write methods and call like this

```
HANDLE  hConsole;
hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
int col=12;

// color your text in Windows console mode
// colors are 0=black 1=blue 2=green and so on to 15=white
// colorattribute = foreground + background * 16
// to get red text on yellow use 4 + 14*16 = 228
// light red on yellow would be 12 + 14*16 = 236

FlushConsoleInputBuffer(hConsole);
SetConsoleTextAttribute(hConsole, col);

cout << "Color Text";

SetConsoleTextAttribute(hConsole, 15); //set back to black background and white
text
```

Share Edit Follow

edited May 19, 2016 at 13:50
Synxis
9,268 3 42 64

answered Oct 29, 2010 at 17:53
Sudantha
15.8k 44 105 161

4 Note that `SetConsoleTextAttribute(hConsole, 15);` sets color to *Bright White*, not to *White*. ***7 - White.*** and ***15 - Bright White*** – GooDeeJAY Apr 28, 2020 at 3:45

10

On Windows 10 you may use escape sequences this way:

```
#ifdef _WIN32
SetConsoleMode(GetStdHandle(STD_OUTPUT_HANDLE),
ENABLE_VIRTUAL_TERMINAL_PROCESSING);
#endif
// print in red and restore colors default
std::cout << "\033[32m" << "Error!" << "\033[0m" << std::endl;
```

Share Edit Follow

answered Jan 29, 2019 at 18:11
Krzysztof Buchacz
351 5 11

`ENABLE_VIRTUAL_TERMINAL_PROCESSING` somehow does not work for me on Windows 10. I get `[32mError![0m` as output. Am I missing something? – ProjectPhysX Mar 18, 2021 at 7:46

Your code resets any other output console flags – Alexey Biryukov May 6, 2021 at 18:32

5 @ProjectPhysX I find a solution, we should save the original console mode first. ``` DWORD dwMode; HANDLE hOutput = GetStdHandle(STD_OUTPUT_HANDLE); GetConsoleMode(hOutput, &dwMode); dwMode |= ENABLE_PROCESSED_OUTPUT | ENABLE_VIRTUAL_TERMINAL_PROCESSING; SetConsoleMode(hOutput, dwMode)); ``` This works for me on Win10. – chenmo Sep 28, 2021 at 3:48

@chenmo This is the correct usage for SetConsoleMode to enable virtual terminal processing. link – Cem Polat Jun 3, 2022 at 11:00

usefuul codes student.cs.uwaterloo.ca/~cs452/terminal.html – superbem Sep 14, 2022 at 18:22

8

The simplest way you can do is:

```
#include <stdlib.h>

system("Color F3");
```

Where "F" is the code for the background color and 3 is the code for the text color.

Mess around with it to see other color combinations:

```
system("Color 1A");
std::cout << "Hello, what is your name?" << std::endl;
system("Color 3B");
std::cout << "Hello, what is your name?" << std::endl;
system("Color 4c");
std::cout << "Hello, what is your name?" << std::endl;
```

Note: *I only tested on Windows. Works. As pointed out, this is not cross-platform, it will not work on Linux systems.*

Share Edit Follow

edited Nov 6, 2020 at 7:08

answered Jan 1, 2017 at 11:35

Taufiq Rahman
5,660 2 37 45

5 This is a bad approach! Code is not cross-platform, it will not work in Linux systems – GooDeeJAY Apr 28, 2020 at 3:40

1 Not only that, you're also invoking system commands. This is terribly inefficient. – Noud Zandbergen Dec 25, 2020 at 10:16

8

Here is my solution that is lightweight and works with both Windows and Linux:

```
#include <iostream>
#include <string>
#ifdef _WIN32
```

```
#define WIN32_LEAN_AND_MEAN
#define VC_EXTRALEAN
#include <Windows.h> // for displaying colors
#endif // Windows
using namespace std;

#define color_black      0
#define color_dark_blue  1
#define color_dark_green 2
#define color_light_blue 3
#define color_dark_red   4
#define color_magenta    5
#define color_orange     6
#define color_light_gray 7
#define color_gray       8
#define color_blue       9
#define color_green     10
#define color_cyan      11
#define color_red       12
#define color_pink      13
#define color_yellow    14
#define color_white     15

string get_textcolor_code(const int textcolor) { // Linux only
    switch(textcolor) {
        case  0: return "30"; // color_black      0
        case  1: return "34"; // color_dark_blue  1
        case  2: return "32"; // color_dark_green 2
        case  3: return "36"; // color_light_blue 3
        case  4: return "31"; // color_dark_red   4
        case  5: return "35"; // color_magenta    5
        case  6: return "33"; // color_orange     6
        case  7: return "37"; // color_light_gray 7
        case  8: return "90"; // color_gray       8
        case  9: return "94"; // color_blue       9
        case 10: return "92"; // color_green     10
        case 11: return "96"; // color_cyan      11
        case 12: return "91"; // color_red       12
        case 13: return "95"; // color_pink      13
        case 14: return "93"; // color_yellow    14
        case 15: return "97"; // color_white     15
        default: return "37";
    }
}
string get_backgroundcolor_code(const int backgroundcolor) { // Linux only
    switch(backgroundcolor) {
        case  0: return  "40"; // color_black      0
        case  1: return  "44"; // color_dark_blue  1
        case  2: return  "42"; // color_dark_green 2
        case  3: return  "46"; // color_light_blue 3
        case  4: return  "41"; // color_dark_red   4
        case  5: return  "45"; // color_magenta    5
        case  6: return  "43"; // color_orange     6
        case  7: return  "47"; // color_light_gray 7
        case  8: return "100"; // color_gray       8
        case  9: return "104"; // color_blue       9
        case 10: return "102"; // color_green     10
        case 11: return "106"; // color_cyan      11
        case 12: return "101"; // color_red       12
        case 13: return "105"; // color_pink      13
        case 14: return "103"; // color_yellow    14
        case 15: return "107"; // color_white     15
        default: return  "40";
    }
}
string get_print_color(const int textcolor) { // Linux only
    return "\033["+get_textcolor_code(textcolor)+"m";
}
string get_print_color(const int textcolor, const int backgroundcolor) { // Linux
only
    return
"\033["+get_textcolor_code(textcolor)+";"+get_backgroundcolor_code(backgroundcolor)+
}
void print_color(const int textcolor) {
#if defined(_WIN32)
    static const HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(handle, textcolor);
#elif defined(__linux__)
    cout << get_print_color(textcolor);
#endif // Windows/Linux
}
void print_color(const int textcolor, const int backgroundcolor) {
#if defined(_WIN32)
    static const HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(handle, backgroundcolor<<4|textcolor);
#elif defined(__linux__)
    cout << get_print_color(textcolor, backgroundcolor);
#endif // Windows/Linux
}
void print_color_reset() {
#if defined(_WIN32)
    static const HANDLE handle = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(handle, 7); // reset color
#elif defined(__linux__)
    cout << "\033[0m"; // reset color
#endif // Windows/Linux
}

void println(const string& s="") {
    cout << s << endl;
}
void print(const string& s="") {
    cout << s;
}
void print(const string& s, const int textcolor) {
    print_color(textcolor);
    cout << s;
    print_color_reset();
}
void print(const string& s, const int textcolor, const int backgroundcolor) {
    print_color(textcolor, backgroundcolor);
    cout << s;
    print_color_reset();
}
void print_no_reset(const string& s, const int textcolor) { // print with color,
but don't reset color afterwards (faster)
    print_color(textcolor);
    cout << s;
}
void print_no_reset(const string& s, const int textcolor, const int
backgroundcolor) { // print with color, but don't reset color afterwards (faster)
    print_color(textcolor, backgroundcolor);
    cout << s;
}
```

And here is an example how to use it:

```
int main() {
    print("Hello ", color_red, color_blue);
    print("World!\n", color_black, color_yellow);
    println();
    return 0;
}
```

Share Edit Follow

edited Aug 23, 2023 at 5:31

answered Apr 21, 2021 at 12:04

ProjectPhysX

4,978 2 17 35

1 Hi, I've got a question. Am I right that `reset` color for Windows is just the `color_light_gray` , as far as you use `7` to reset color? – hazer_hazer Jun 11, 2021 at 14:49

1 @hazer_hazer yes exactly. Light gray text and black background is the default color for the Windows console. If you have set a different default color, you can make a global variable defaultcolor, initialize it in the beginning with GetConsoleTextAtteibute(handle, defaultcolor); and use that for print_color_reset();. The Linux variant in contrast always goes back to whatever you have set as default colors, not necessarily light gray on black. – ProjectPhysX Jun 12, 2021 at 5:14

wait, hows that going to compile in linux with windows.h? – j0h Jul 30, 2021 at 1:01

1 @j0h there is an `#ifdef _WIN32 ... #endif` switch for the preprocessor. On Linux, `_WIN32` is not defined, so it does not include `<Windows.h>` on a Linux system. The `_WIN32` and `__linux__` defines are very useful as they allow for detecting on which operating system the code runs. – ProjectPhysX Jul 30, 2021 at 6:04

1 This solution is ok – DevWolf31 Nov 12, 2022 at 22:45

7

You can use ANSI escape sequences to colorizing the console text, it works for windows and Linux. For Windows, you need to activate the virtual terminal.

```
#include <iostream>

#ifdef _WIN32
#include <windows.h>

#define ENABLE_VIRTUAL_TERMINAL_PROCESSING 0x0004
#define DISABLE_NEWLINE_AUTO_RETURN  0x0008

void activateVirtualTerminal()
{
    HANDLE handleOut = GetStdHandle(STD_OUTPUT_HANDLE);
    DWORD consoleMode;
    GetConsoleMode( handleOut , &consoleMode);
    consoleMode |= ENABLE_VIRTUAL_TERMINAL_PROCESSING;
    consoleMode |= DISABLE_NEWLINE_AUTO_RETURN;
    SetConsoleMode( handleOut , consoleMode );
}
#endif

using namespace std;

enum COLORS {
    NC=-1,
    BLACK,
    RED,
    GREEN,
    YELLOW,
    BLUE,
    MAGENTA,
    CYAN,
    WHITE,
};

/**
* Colorize terminal colors ANSI escape sequences.
*
* @param font font color (-1 to 7), see COLORS enum
* @param back background color (-1 to 7), see COLORS enum
* @param style font style (1==bold, 4==underline)
**/
const char *colorize(int font, int back = -1, int style = -1) {
    static char code[20];

    if (font >= 0)
        font += 30;
    else
        font = 0;
    if (back >= 0)
        back += 40;
    else
        back = 0;

    if (back > 0 && style > 0) {
        sprintf(code, "\033[%d;%d;%dm", font, back, style);
    } else if (back > 0) {
        sprintf(code, "\033[%d;%dm", font, back);
    } else {

        sprintf(code, "\033[%dm", font);
    }

    return code;
}


int main()
{
#ifdef _WIN32
    activateVirtualTerminal();
#endif

    cout << colorize(RED) << "trying red" << colorize(NC) << endl;
    cout << colorize(RED, BLACK) << "red and black background" << colorize(NC) <<
endl;
    cout << colorize(YELLOW, BLUE, 1) << "yellow blue bold" << colorize(NC) <<
endl;
```

```
    cout << colorize(BLACK, WHITE) << "Black white" << colorize(NC) << endl;
    cout << colorize(MAGENTA, CYAN) << "Magenta cyan" << colorize(NC) << endl;

    return 1;
}
```

Share Edit Follow

edited Sep 28, 2021 at 22:53

answered Sep 28, 2021 at 4:11
Derzu
7,057 3 58 61

4

Assuming you're talking about a Windows console window, look up the console functions in the MSDN Library documentation.

Otherwise, or more generally, it depends on the console. Colors are not supported by the C++ library. But a library for console handling may/will support colors. E.g. google "ncurses colors".

For connected serial terminals and terminal emulators you can control things by outputting "escape sequences". These typically start with ASCII 27 (the escape character in ASCII). There is an ANSI standard and a lot of custom schemes.

Share Edit Follow

edited Sep 16, 2018 at 15:03
Matthias
4,539 12 46 84

answered Oct 29, 2010 at 16:26
Cheers and hth. - Alf
144k 15 212 334

Since that answer was written, Windows 10 gained support for ANSI escape sequences for the narrow console streams. – Cheers and hth. - Alf Oct 3, 2016 at 18:27

4

In Windows, you can use any combination of red green and blue on the foreground (text) and the background.

```
/* you can use these constants
FOREGROUND_BLUE
FOREGROUND_GREEN
FOREGROUND_RED
FOREGROUND_INTENSITY
BACKGROUND_BLUE
BACKGROUND_GREEN
BACKGROUND_RED
BACKGROUND_INTENSITY
*/

HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
SetConsoleTextAttribute(hConsole, FOREGROUND_BLUE | FOREGROUND_GREEN |
FOREGROUND_INTENSITY);
std::cout << "I'm cyan! Who are you?" << std::endl;
```

Source: https://msdn.microsoft.com/en-us/library/windows/desktop/ms682088(v=vs.85).aspx#_win32_character_attributes

Share Edit Follow

answered Jun 28, 2016 at 20:53
cambunctious
8,780 5 35 55

3

I'm not sure what you really want to do, but my guess is you want your C++ program to output colored text in the console, right ? Don't know about Windows, but on all Unices (including Mac OS X), you'd simply use ANSI escape sequences for that.

Share Edit Follow

answered Oct 29, 2010 at 16:26
DarkDust
91.3k 19 191 225

0

Do not use "system("Color …")" if you don't want the entire screen to be filled up with color. This is the script needed to make colored text:

```
#include <iostream>
#include <windows.h>

int main()
{
const WORD colors[] =
{
0x1A, 0x2B, 0x3C, 0x4D, 0x5E, 0x6F,
0xA1, 0xB2, 0xC3, 0xD4, 0xE5, 0xF6
};

HANDLE hstdin = GetStdHandle(STD_INPUT_HANDLE);
HANDLE hstdout = GetStdHandle(STD_OUTPUT_HANDLE);
WORD   index = 0;


    SetConsoleTextAttribute(hstdout, colors[index]);
    std::cout << "Hello world" << std::endl;
FlushConsoleInputBuffer(hstdin);
return 0;
}
```

Share Edit Follow

answered May 4, 2019 at 10:16
user11441767

You don't need to use any library. Just only write system("color 4f");

0

Share Edit Follow

answered Dec 18, 2019 at 13:02
user12558829
9 1

8 This is windows only, and IIRC this will just make the entire screen one color – nathanfranke Apr 20, 2020 at 0:52

Here [cplusplus example] is an example how to use colors in console.

0

Share Edit Follow

answered Oct 29, 2010 at 16:26
Łukasz Milewski
1,915 14 14

The answer does not add anything to the existing ones – MrFox Mar 6, 2022 at 15:43

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.