# Bash tips: Colors and formatting (ANSI/VT100 Control sequences)

The **ANSI/VT100** terminals and terminal emulators are not just able to display black and white text ; they can display **colors** and formatted texts thanks to **escape sequences**. Those sequences are composed of the **Escape character** (often represented by "^[" or "<Esc>") followed by some other characters: "<Esc>[*FormatCode*m".

In Bash, the <Esc> character can be obtained with the following syntaxes:

- \e
- \033
- \x1B

*Examples:*

| Code (Bash) | Preview |
|---|---|
| `echo -e "\e[31mHello World\e[0m"` | Hello World |
| `echo -e "\033[31mHello\e[0m World"` | Hello World |

*NOTE[1]:* The -e option of the echo command enable the parsing of the escape sequences.

*NOTE[2]:* The "\e[0m" sequence removes all attributes (formatting and colors). It can be a good idea to add it at the end of each colored text. ;)

*NOTE[3]:* The examples in this page are in **Bash** but the **ANSI/VT100** escape sequences can be used in every programming languages.

## Formatting

Here are the most commonly supported control sequences for formatting text. Their support depends on the used terminal (see the compatibility list).

### Set

| Code | Description | Example | Preview |
|---|---|---|---|
| 1 | Bold/Bright | `echo -e "Normal \e[1mBold"` | Normal **Bold** |
| 2 | Dim | `echo -e "Normal \e[2mDim"` | Normal Dim |

| | | | |
|---|---|---|---|
| 4 | Underlined | `echo -e "Normal \e[4mUnderlined"` | Normal Underlined |
| 5 | Blink [1] | `echo -e "Normal \e[5mBlink"` | Normal |
| 7 | Reverse (invert the foreground and background colors) | `echo -e "Normal \e[7minverted"` | Normal inverted |
| 8 | Hidden (usefull for passwords) | `echo -e "Normal \e[8mHidden"` | Normal |

## Reset

| Code | Description | Example | Preview |
|---|---|---|---|
| 0 | Reset all attributes | `echo -e "\e[0mNormal Text"` | Normal Text |
| 21 | Reset bold/bright | `echo -e "Normal \e[1mBold \e[21mNormal"` | Normal **Bold** Normal |
| 22 | Reset dim | `echo -e "Normal \e[2mDim \e[22mNormal"` | Normal Dim Normal |
| 24 | Reset underlined | `echo -e "Normal \e[4mUnderlined \e[24mNormal"` | Normal Underlined Normal |
| 25 | Reset blink | `echo -e "Normal \e[5mBlink \e[25mNormal"` | Normal Blink Normal |
| 27 | Reset reverse | `echo -e "Normal \e[7minverted \e[27mNormal"` | Normal inverted Normal |
| 28 | Reset hidden | `echo -e "Normal \e[8mHidden \e[28mNormal"` | Normal Normal |

# 8/16 Colors

The following colors works with most terminals and terminals emulators [2], see the compatibility list for more informations.

*NOTE:* The colors can vary depending of the terminal configuration.

## Foreground (text)

| Code | Color | Example | Preview |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 39 | Default foreground color | `echo -e "Default \e[39mDefault"` | Default Default |
| 30 | Black | `echo -e "Default \e[30mBlack"` | Default Black |
| 31 | Red | `echo -e "Default \e[31mRed"` | Default Red |
| 32 | Green | `echo -e "Default \e[32mGreen"` | Default Green |
| 33 | Yellow | `echo -e "Default \e[33mYellow"` | Default Yellow |
| 34 | Blue | `echo -e "Default \e[34mBlue"` | Default Blue |
| 35 | Magenta | `echo -e "Default \e[35mMagenta"` | Default Magenta |
| 36 | Cyan | `echo -e "Default \e[36mCyan"` | Default Cyan |
| 37 | Light gray | `echo -e "Default \e[37mLight gray"` | Default Light gray |
| 90 | Dark gray | `echo -e "Default \e[90mDark gray"` | Default Dark gray |
| 91 | Light red | `echo -e "Default \e[91mLight red"` | Default Light red |
| 92 | Light green | `echo -e "Default \e[92mLight green"` | Default Light green |
| 93 | Light yellow | `echo -e "Default \e[93mLight yellow"` | Default Light yellow |
| 94 | Light blue | `echo -e "Default \e[94mLight blue"` | Default Light blue |
| 95 | Light magenta | `echo -e "Default \e[95mLight magenta"` | Default Light magenta |
| 96 | Light cyan | `echo -e "Default \e[96mLight cyan"` | Default Light cyan |

| | | | |
|---|---|---|---|
| 97 | White | `echo -e "Default \e[97mWhite"` | Default White |

## Background

| Code | Color | Example | Preview |
|---|---|---|---|
| 49 | Default background color | `echo -e "Default \e[49mDefault"` | Default Default |
| 40 | Black | `echo -e "Default \e[40mBlack"` | Default Black |
| 41 | Red | `echo -e "Default \e[41mRed"` | Default Red |
| 42 | Green | `echo -e "Default \e[42mGreen"` | Default Green |
| 43 | Yellow | `echo -e "Default \e[43mYellow"` | Default Yellow |
| 44 | Blue | `echo -e "Default \e[44mBlue"` | Default Blue |
| 45 | Magenta | `echo -e "Default \e[45mMagenta"` | Default Magenta |
| 46 | Cyan | `echo -e "Default \e[46mCyan"` | Default Cyan |
| 47 | Light gray | `echo -e "Default \e[47mLight gray"` | Default Light gray |
| 100 | Dark gray | `echo -e "Default \e[100mDark gray"` | Default Dark gray |
| 101 | Light red | `echo -e "Default \e[101mLight red"` | Default Light red |
| 102 | Light green | `echo -e "Default \e[102mLight green"` | Default Light green |
| 103 | Light yellow | `echo -e "Default \e[103mLight yellow"` | Default Light yellow |

| | | | |
|---|---|---|---|
| 104 | Light blue | `echo -e "Default \e[104mLight blue"` | Default Light blue |
| 105 | Light magenta | `echo -e "Default \e[105mLight magenta"` | Default Light magenta |
| 106 | Light cyan | `echo -e "Default \e[106mLight cyan"` | Default Light cyan |
| 107 | White | `echo -e "Default \e[107mWhite"` | Default |

## 88/256 Colors

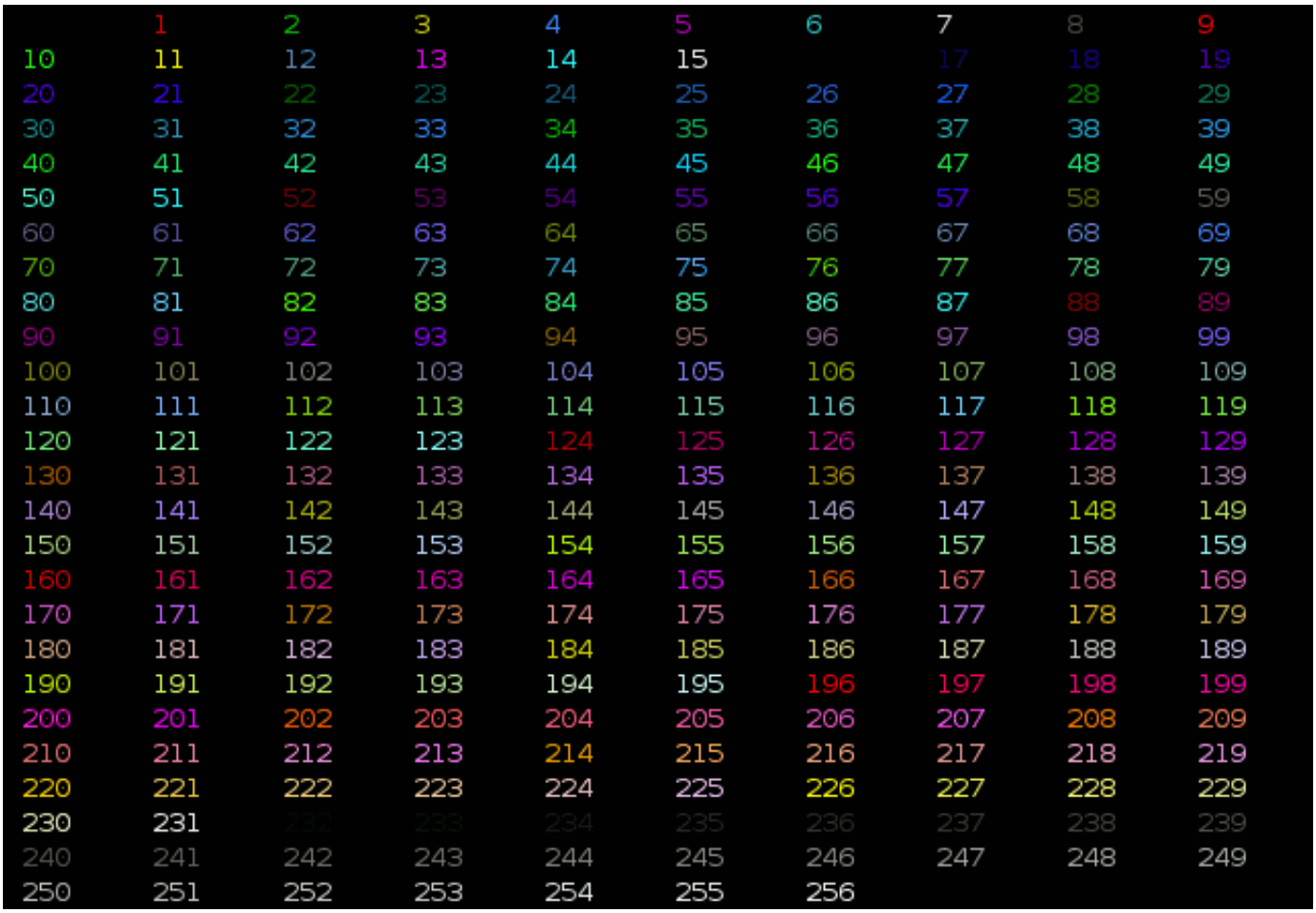
Some terminals (see the compatibility list) can support 88 or 256 colors. Here are the control sequences that permit you to use them.

*NOTE*[1]: The colors number `256` is only supported by **vte** (GNOME Terminal, XFCE4 Terminal, Nautilus Terminal, Terminator,...).

*NOTE*[2]: The 88-colors terminals (like **rxvt**) does not have the same color map that the 256-colors terminals. For showing the 88-colors terminals color map, run the "256-colors.sh" script in a 88-colors terminal.

### Foreground (text)

For using one of the 256 colors on the foreground (text color), the control sequence is "`<Esc>[38;5;`*`ColorNumber`*`m`" where `ColorNumber` is one of the following colors:



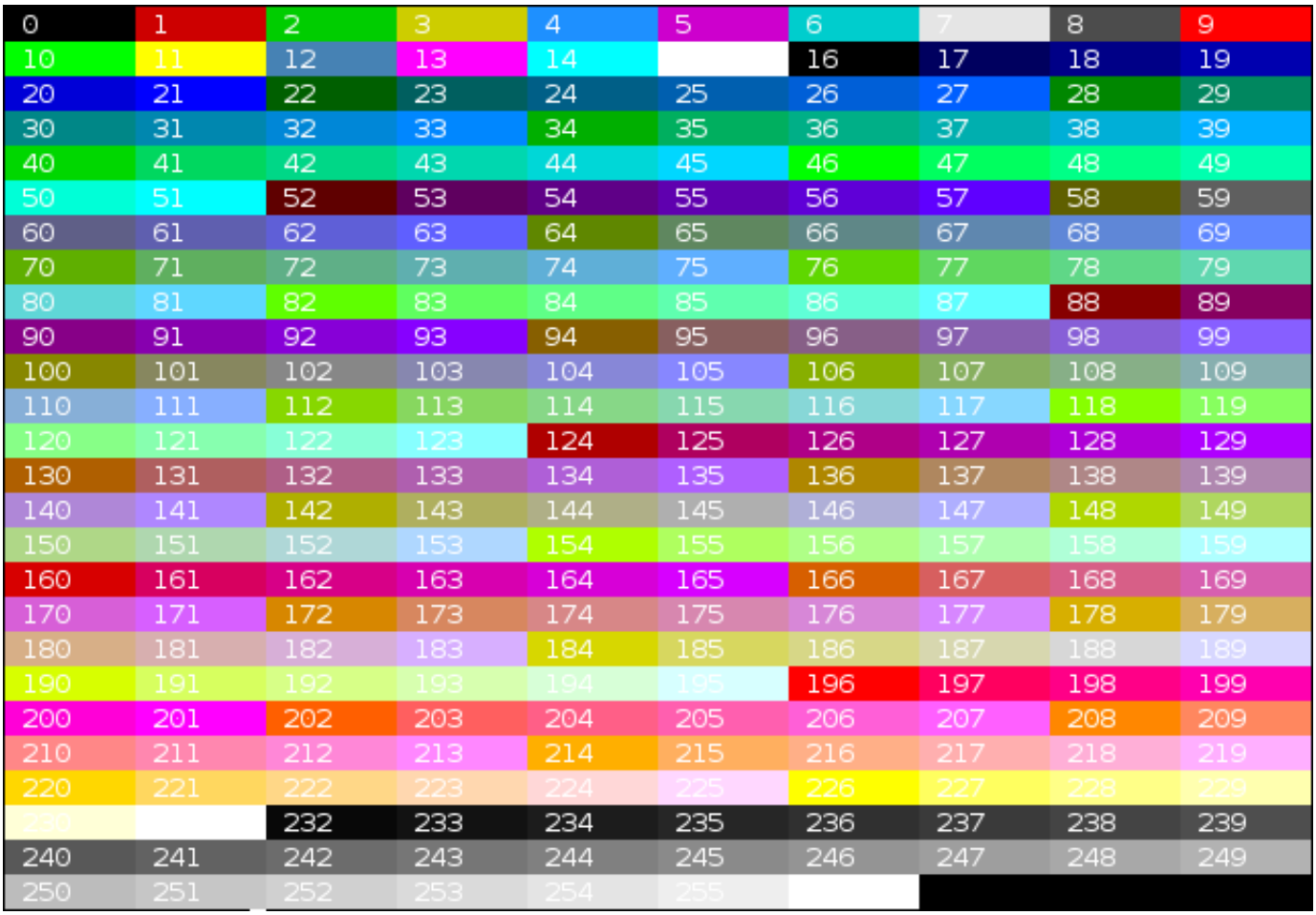
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
| 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 |
| 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 |
| 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 |
| 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 |
| 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 |
| 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 |
| 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 |
| 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 |
| 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 |
| 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 |
| 220 | 221 | 222 | 223 | 224 | 225 | 226 | 227 | 228 | 229 |
| 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 |
| 250 | 251 | 252 | 253 | 254 | 255 | 256 | | | |

*Examples:*

| Code (Bash) | Preview |
|---|---|
| `echo -e "\e[38;5;82mHello \e[38;5;198mWorld"` | Hello World |
| `for i in {16..21} {21..16} ; do echo -en "\e[38;5;${i}m#\e[0m" ; done ; echo` | ########## |

## Background

For using one of the 256 colors on the background, the control sequence is "<Esc>[48;5;*ColorNumber*m" where ColorNumber is one of the following colors:




*Examples:*

| Code (Bash) | Preview |
|---|---|
| `echo -e "\e[40;38;5;82m Hello \e[30;48;5;82m World \e[0m"` | Hello World |
| `for i in {16..21} {21..16} ; do echo -en "\e[48;5;${i}m \e[0m" ; done ; echo` | |

## Attributes combination

Terminals allow attribute combinations. The attributes must be separated by a semicolon (”**;**”).

*Examples:*

| Description | Code (Bash) | Preview |
|---|---|---|
| Bold + Underlined | `echo -e "\e[1;4mBold and Underlined"` | Bold and Underlined |
| Bold + Red forground + Green background | `echo -e "\e[1;31;42m Yes it is awful \e[0m"` | Yes it is awful |

# Terminals compatibility

| Terminal | Formatting | | | | | | Colors | | | | Comment |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Bold | Dim | Underlined | Blink | invert | Hidden | 8 | 16 | 88 | 256 | |
| aTerm [http://www.afterstep.org/aterm.php] | ok | – | ok | – | ok | – | ok | ~ | – | – | Lighter background instead of blink. |
| Eterm [http://www.eterm.org/] | ~ | – | ok | – | ok | – | ok | ~ | – | ok | Lighter color instead of Bold. Lighter background instead of blink. Can overline a text with the "^[[6m" sequence. |
| GNOME Terminal [http://library.gnome.org/users/gnome-terminal/] | ok | ok | ok | – | ok | ok | ok | ok | – | ok | Strikeout with the "^[[9m" sequence. |
| Guake [http://guake.org/] | ok | ok | ok | – | ok | ok | ok | ok | – | ok | Strikeout with the "^[[9m" sequence. |
| Konsole [http://konsole.kde.org/] | ok | – | ok | ok | ok | – | ok | ok | – | ok | |
| Nautilus Terminal [http://software.flogisoft.com/nautilus-terminal/] | ok | ok | ok | – | ok | ok | ok | ok | – | ok | Strikeout with the "^[[9m" sequence. |
| rxvt [http://rxvt.sourceforge.net/] | ok | – | ok | ~ | ok | – | ok | ok | ok | – | If the background is not set to the default color, Blink make it lighter instead of blinking. Support of italic text with the "^[[3m" sequence. |
| Terminator [http://www.tenshu.net/terminator/] | ok | ok | ok | – | ok | ok | ok | ok | – | ok | Strikeout with the "^[[9m" sequence. |
| Tilda [http://tilda.sourceforge.net/tildaabout.php] | ok | – | ok | – | ok | – | ok | ok | – | – | Underline instead of Dim. Convert 256-colors in 16-colors. |
| XFCE4 Terminal [http://www.xfce.org/projects/terminal] | ok | ok | ok | – | ok | ok | ok | ok | – | ok | Strikeout with the "^[[9m" sequence. |
| XTerm [http://invisible-island.net/xterm/xterm.html] | ok | – | ok | ok | ok | ok | ok | ok | – | ok | |
| xvt | ok | – | ok | – | ok | – | – | – | – | – | |
| Linux TTY | ok | – | – | – | ok | – | ok | ~ | – | – | Specials colors instead of Dim and Underlined. Lighter background instead of Blink, Bug with 88/256 colors. |
| VTE Terminal [http://developer.gnome.org/vte/] [3)] | ok | ok | ok | – | ok | ok | ok | ok | – | ok | Strikeout with the "^[[9m" sequence. |

*Notations used in the table:*

- "ok": Supported by the terminal.
- "~": Supported in a special way by the terminal.
- "–": Not supported at all by the terminal.

# Demonstration programs

## Colors and formatting (16 colors)

The following shell script displays a lot of possible combination of the attributes (but not all, because it uses only one formatting attribute at a time).

colors_and_formatting.sh

```
#!/bin/bash

# This program is free software. It comes wit
# the extent permitted by applicable law. You
# and/or modify it under the terms of the Do
# To Public License, Version 2, as published
# http://sam.zoy.org/wtfpl/COPYING for more d

#Background
for clbg in {40..47} {100..107} 49 ; do
        #Foreground
        for clfg in {30..37} {90..97} 39 ; do
                #Formatting
                for attr in 0 1 2 4 5 7 ; do
                        #Print the result
                        echo -en "\e[${attr};${clbg};${clfg}m ^[${attr};${clbg};${clfg}m \e[0m"
                done
                echo #Newline
        done
done

exit 0
```

## 256 colors

The following script display the 256 colors available on some terminals and terminals emulators like **XTerm** and **GNOME Terminal**.

256-colors.sh

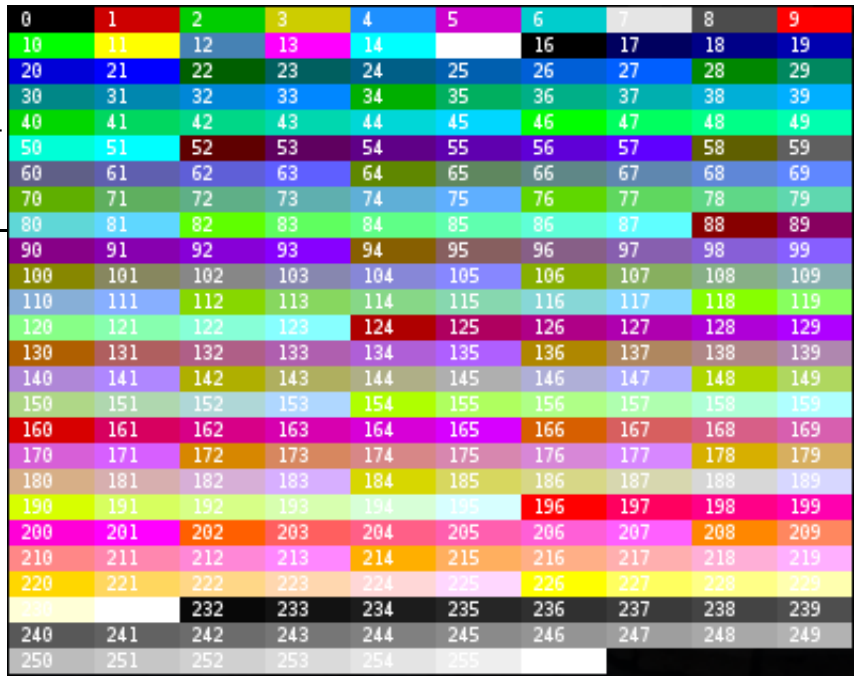
```
#!/bin/bash

# This program is free software. It comes without any warranty, to
# the extent permitted by applicable law. You can redistribute it
# and/or modify it under the terms of the Do What The Fuck You Want
# To Public License, Version 2, as published by Sam Hocevar. See
# http://sam.zoy.org/wtfpl/COPYING for more details.

for fgbg in 38 48 ; do #Foreground/Background
        for color in {0..256} ; do #Colors
                #Display the color
                echo -en "\e[${fgbg};5;${color}m ${color}\t\e[0m"
                #Display 10 colors per lines
                if [ $((($color + 1) % 10)) == 0 ] ; then
                        echo #New line
                fi
        done
```

```
        echo #New line
done

exit 0
```



## Links

- Linux console codes manual ("man console_codes") [http://linux.die.net/man/4/console_codes]
- XTerm Control Sequences [http://invisible-island.net/xterm/ctlseqs/ctlseqs.html]
- Compilation of all escape sequences [http://bjh21.me.uk/all-escapes/all-escapes.txt]
- ANSI escape code (Wikipedia) [http://en.wikipedia.org/wiki/ANSI_escape_code]

---

[1] Does not work with most of the terminal emulators, works in the tty and XTerm.
[2] Some terminals supports only the first 8 colors (30..37 and 40..47), and some others does not support any color at all.
[3] GTK Widget used in GNOME Terminal, Nautilus Terminal, XFCE4 Terminal...

## Discussion

William C Grisaitis, 2011/11/13 02:00
Thanks! This was invaluable in customizing my PS1's:

```
if [[ ${EUID} == 0 ]] ; then
PS1='\e[1;31;48;5;234m\u \e[38;5;240mon \e[1;38;5;28;48;5;234m\h \e[38;5;54m\d
\@\e[0m\n\e[0;31;48;5;234m[\w] \e[1m\$\e[0m '
else
PS1='\e[1;38;5;56;48;5;234m\u \e[38;5;240mon \e[1;38;5;28;48;5;234m\h \e[38;5;54m\d
\@\e[0m\n\e[0;38;5;56;48;5;234m[\w] \e[1m\$\e[0m '
fi
```

@caravaggisto

Barry Scott, 2012/06/14 21:41
Great work on terminal compatibility.I have been trying to get blinking text on a Linux tty(at the console). Do you have any idea if it's possible?

Fabien LOISON, 2012/06/14 21:54
I Think it is possible, but I haven't found how to do that

Barry Scott, 2012/06/14 22:11
I have looked at infocmp for linux the terminal(TERM=linux) I use and I see blink referenced

in it but I'm having a hard time understanding the file format. The cursor blinks why disable blinking text.

Warron, 2013/04/04 19:09
Great page on bash coloring and attributes.

I was actually looking to find out if there is a way to combine attributes {BOLD, Blink, etc} around the same subset of text in doing a bash echo command with the -e option.

Can you help with this matter?

\\War

Fabien LOISON, 2013/04/04 21:54
Hello,

You can combine attributes with a semicolon:

```
echo -e "\e[1;5m Bold+Blink \e[0m"
echo -e "\e[1;4;31m Bold+Underline+Red \e[0m"
```

Note that the blink attribute is supported only by few terminals (XTerm, tty).

Regards,

Warron, 2013/04/11 21:49
Thank you Fabian.

That worked splendidly! You are the man!

Konrad, 2015/07/25 21:51
Thank you!

konrad@vps1 ~/web/abc▌▌▌ mkdir xyz

```
PS1='\[\e[0m\]\[\e[48;5;236m\]\[\e[38;5;105m\]\u\[\e[38;5;105m\]@\[\e[38;5;105m\]\h\
[\e[38;5;105m\]                    \[\e[38;5;221m\]\w\[\e[38;5;221m\]\[\e[38;5;105m\]\[\e[0m\]\
[\e[38;5;236m\]\342\226\214\342\226\214\342\226\214\[\e[0m\]'
```

root:

```
PS1='\[\e[0m\]\[\e[48;5;236m\]\[\e[38;5;197m\]\u\[\e[38;5;197m\]@\[\e[38;5;197m\]\h\
[\e[38;5;105m\]                    \[\e[38;5;221m\]\w\[\e[38;5;221m\]\[\e[38;5;105m\]\[\e[0m\]\
[\e[38;5;236m\]\342\226\214\342\226\214\342\226\214\[\e[0m\]'
```

Per Bothner, 2015/11/28 20:03
Note that the 256-colors.sh script uses a tab character, which has different behavior on different emulators.

On xterm and Konsole, TAB moves the cursor, without touching the skipped-over positions (so the background color is unchanged), while Gnome Terminal appears to effectively write spaces (so the background color is changed). Your images show the latter, but note that is incompatible with xterm.

Per Bothner, 2015/11/28 20:14
A fix for 256-colors.sh that uses printf instead of tabs:

```
#Display the color
echo -en "\e[${fgbg};5;${color}m"
printf "%4d " ${color}
```

echo -en "\e[0m"

Also, the upper cbound should be 255, not 256:

for color in {0..255} ; do #Colors

Nga Nguyen Duy, 2015/12/06 21:55
I don't know what is the difference between the <ESC> characters:

\e

\033

\x1B

Can somebody explain for me?
Thank in advance.

Fabien LOISON, 2015/12/07 09:29
Hello,

This is only three ways to represent the same character. There will be no differences between using one representation or an other.

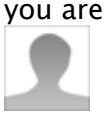
* \e is a convenient way provided by Bash to insert the Escape character.

* 33 is the position of the Escape character in the ASCII table expressed in octal (base 8, in decimal this is equal to 27)
* 1B is the position of the Escape character in the ASCII table expressed in hexadecimal (base 16)

→ So \0nn and \xNN are just a way to insert a character by providing its position in the ASCII table, in octal or hexadecimal format.

You can find an the ASCII table here → https://duckduckgo.com/?q=ascii+table&t=canonical

g alexander, 2016/02/12 22:13
you are a bash scripting color, ascii man among boys



g alexander, 2016/02/12 22:20
good work.

with that gradient,i was trying to work how to put text inside of it to get a gradient of of text but the text just repeats with the loop. how can i put this into a function something like gradient "some text" blue white or gradient "more text" blue white yellow, function gradient(){}?

Mohsen Pahlevanzadeh, 2016/03/15 02:33
blink code doesn't work.

for example:

echo -e "Normal \033[5mHello"

Normal Hello

##### It's normal print Normal Hello, Not blink.

Can you write truly blink text?
Fabien LOISON, 2016/03/15 09:13
Hello,

blink do not work on vte based terminals (most linux terminal, like gnome-terminal, tilda, guake, terminator, xfce4-terminal,...)

You can try with xterm, it should work on it.

See the compatibility table for more info: http://misc.flogisoft.com/bash/tip_colors_and_formatting?&#terminals_compatibility

Gerry, 2016/04/12 20:26
Here's a little more on resetting:

\e[0m resets all colors and attributes.
\e[20m resets only attributes (underline, etc.), leaving colors unchanged.
\e[39m resets only foreground color, leaving attributes unchanged.
\e[49m resets only background color, leaving attributes unchanged.

Ron, 2016/05/13 15:17
(Taken from http://makandracards.com/makandra/1090-customize-your-bash-prompt :)
\u: current username
\h: hostname up to the first ., \H: full hostname
\w: current working directory, \W: same, but only the basename
$(__git_ps1 "%s"): your current git branch if you're in a git directory, otherwise nothing

\$: if the effective UID is 0: #, otherwise $
\d: the date in "Weekday Month Date" format (e.g., "Tue May 26")
\t: the current time in 24-hour HH:MM:SS format, \T: same, but 12-hour format, \@: same, but in 12-hour am/pm format
\n: newline
\r: carriage return
\\: backslash

Fabien LOISON, 2016/05/13 15:21
@Ron: \u, \h &co are available only in prompts:

http://misc.flogisoft.com/bash/tip_customize_the_shell_prompt

Toby, 2016/06/13 15:11
Please, please, please DON'T encourage people to put the raw terminal codes into their message strings! That way lies madness, because not all the world is a VT100/VT220/etc. Instead, use the 'tput' program to generate the correct code (if one exists) for the user's terminal. That is much more portable, and doesn't clutter the poor user's screen with lots of escape character clutter when they run your program from a non-terminal environment.

Fabien LOISON, 2016/06/13 15:21
Of course it is better to use libs or programs that abstract all the things and make it works with almost any terminals. But it still usefull to know how it works behind :)

fujisan, 2016/06/14 11:05
On a mate terminal with a white background, the bold (echo -e "Default \e[1mDefault") is actually white so impossible to see the characters.

Fabien LOISON, 2016/06/14 14:05
In GNOME Terminal there is an option to set the color of the bold text (right click → Profiles → Profile Settings → Colors → Bold colors), there should be the same on mate-terminal.

PS: I translated the menu label from my french gnome-terminal. In yours, it can be slightly deferent.

Aakash Martand, 2016/09/23 10:30
Nice work.

would you please explain the control sequence of 8/16 Colors and 88/256 Colors

Fabien LOISON, 2016/09/26 12:47
what do you want I explain ?

Aakash Martand, 2016/09/26 15:14
Like in your example, \e[30;48;5;82m World you've used 4 parameters. Is there any specific sequence for that?

As I understand,

30 is for black text.

48 is for what?

5 is for blink which is not happening, not even in Xterm.

82 is background color.

please help.

Fabien LOISON, 2016/09/26 15:24
Ah ok,

In 8/16 color mode:

"3x" is for foreground color
"4x" is for background color

In 88/256 color mode:

"38;5" means "the next number is a foreground color in 88/256 color mode"
"48;5" means "the next number is a background color in 88/256 color mode"

so "38;5;XXX" and "48;5;XXX" allow you to select colors in 88/256 color mode.

In your example ("\e[30;48;5;82m"),
"30" is for back foreground (text in black)
"48;5;82" is for green background (in 88/256 color mode)

Aakash Martand, 2016/09/26 16:01
Now I clearly understand.

Thanx buddy.

keep rocking.

Joe, 2016/10/11 19:13
This is an awesome document! It is well written! Thanks for making it clear.

Cheers,

+ Joe

Mark, 2016/10/20 11:02

Perfect tips! One more question - how make colored backgroud to whole line?

Fabien LOISON, 2016/10/20 13:48

I do not know other solution than filling the line with spaces...

Eddie, 2016/11/15 13:48

Hi all,

In my shell script formating text (bold/colors) all works and the results look correct

if the output is sent to standard output. (Just calling the script ./myscript.sh

But, if i redirect the output into a file i only see original text including

statements such as ESC[90G ESC[1;32 and so on.

Any ideas?

1. Content of myscript.sh:

echo -e "OKAY TO BE PRINTED IN COLUMN 50 OF THIS LINE \e[20G OKAY"

2. ./myscript.sh &> output.txt 2>&1

3. Use Notepad++ to open output.txt: I see

OKAY TO BE PRINTED ON COLUMN 50 OF THIS LINE ESC[20G OKAY

If i use cat to show the content i see the correct results.

However, i want to see the same result in the text file as it is shown on default output.

Eddy, 2016/11/15 13:57

Hi all,

do you know how can i make this formating to be kept in the file if i redirect the output of my shell script?

1. Content of my shell script "myscript.sh"

echo -e "PRINT RED HELLO AT COLUMN POSITION 80 \e[80G \e[91m HELLO"

2. ./myscript.sh &> output.txt

3. Content of output.txt:

PRINT RED HELLO AT COLUMN POSITION 80 ESC[80G ESC[91m HELLO

Many thanks for your support in advance.

Regards, Eddy

Fabien LOISON, 2016/11/17 20:23
Hello,

You cannot see the formating in your text editor, because it is your terminal emulator (XTerm, GNOME Terminal, Konsole,...) that generates colors when there is some special byte sequence in the output. Your text editor will just display the content of the file, it will not interpret it.

Regards,

Emeric, 2016/11/24 18:59
Hey guys, here is another script to display 256 colors in a terminal.

To be honnest it's basically the same but the output is a bit more... readable.

```
for fgbg in 38 48 ; do

i=0

for color in {0..15} ; do

if [ $i -lt 10 ] ; then

echo "\x1B[${fgbg};5;${color}m"' '${color}' '"\x1B[0m" | tr -d '\n'

else

echo "\x1B[${fgbg};5;${color}m"' '${color}' '"\x1B[0m" | tr -d '\n'

fi

i=$(($i+1))

if [ $((i % 8)) == 0 ] ; then

echo

fi

done

i=0

for color in {16..255} ; do

if [ $i -lt 84 ] ; then

echo "\x1B[${fgbg};5;${color}m"' '${color}' '"\x1B[0m" | tr -d '\n'

else
```

```
echo "\x1B[${fgbg};5;${color}m"' '${color}' '"\x1B[0m" | tr -d '\n'
fi
i=$(($i+1))
if [ $((i % 6)) == 0 ] ; then
echo
fi
done
echo
echo
done
exit 0
```

(sorry for the horrible indentation, no way to fix this unfortunately)