Colorize linux programs output

Colorize linux program output

You can colorize linux bash shell or any linux program like **echo**, **sed**, **grep** and another with this simple steps:

1. You need type ASCII `ESC` character before your colorized text
2. **ESC** character has to be followed by **[**
3. After that you can write one or two numbers separated by **;**

- First number is one of:
  - **color-mode**: It modifies the style of color NOT text. For example make the color bright or darker.
    - **0** - normal (default)
    - **1** - lighter than normal
    - **2** - darker than normal
  - **text-mode**: This mode is for modifying the style of text NOT color.
    - **3** - italic
    - **4** - underline
    - **5** - blinking (slow)
    - **6** - blinking (fast)
    - **7** - reverse
    - **8** - hide
    - **9** - cross-out
- Second number is **color-code** separated by **;** from first number is ANSI color code . eg. 31 for red color, 34 for blue color for 3/4 bit ANSI color code . Color is one of:
  - **foreground mode** - This mode is for colorizing the foreground.
  - **background mode** - This mode is for colorizing the background.

> second number - **color-code** in step 3 is **3/4 bit** or **8 bit** or **24 bit ANSI color code**

4. Then you have to write **m** char

5. You can get back to standard output with **ESC[0m**

0 is appended if you omit it in step 5 - e.g. you can write **ESC[m** in bash

ESC - you can type escape character to your console or linux program with some of ASCII code :

Decimal	Octal	Hex	Binary	Value	Description	Caret notation	Escape sequence in bash or C
027	033	1B	0001 1011	ESC	escape	^[	\e

3/4bit ANSI Color

The below table shows a summary of **3/4 bit** version of ANSI color.

```
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
| color-mode | octal   | hex     | bash  | description      | example (ESC in octal)     | NOTE                                 |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
|          0 | \033[0m | \x1b[0m | \e[0m | reset any affect | echo -e "\033[0m"          | 0m equals to m                       |
|          1 | \033[1m |         |       | light (= bright) | echo -e "\033[1m####\033[m"  | -                                    |
|          2 | \033[2m |         |       | dark (= fade)    | echo -e "\033[2m####\033[m"  | -                                    |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
|  text-mode | ~       |         |       | ~                | ~                          | ~                                    |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
|          3 | \033[3m |         |       | italic           | echo -e "\033[3m####\033[m"  |                                      |
|          4 | \033[4m |         |       | underline        | echo -e "\033[4m####\033[m"  |                                      |
|          5 | \033[5m |         |       | blink (slow)     | echo -e "\033[5m####\033[m"  |                                      |
|          6 | \033[6m |         |       | blink (fast)     | ?                          | not wildly support                   |
|          7 | \003[7m |         |       | reverse          | echo -e "\033[7m####\033[m"  | it affects the background/foreground |
|          8 | \033[8m |         |       | hide             | echo -e "\033[8m####\033[m"  | it affects the background/foreground |
|          9 | \033[9m |         |       | cross            | echo -e "\033[9m####\033[m"  |                                      |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
| foreground | all below examples are with ommited color/text mode - default is 0                                                 |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
|         30 | \033[30m |         |       | black            | echo -e "\033[30m####\033[m" |                                      |
|         31 | \033[31m |         |       | red              | echo -e "\033[31m####\033[m" |                                      |
|         32 | \033[32m |         |       | green            | echo -e "\033[32m####\033[m" |                                      |
|         33 | \033[33m |         |       | yellow           | echo -e "\033[33m####\033[m" |                                      |
|         34 | \033[34m |         |       | blue             | echo -e "\033[34m####\033[m" |                                      |
|         35 | \033[35m |         |       | purple           | echo -e "\033[35m####\033[m" | real name: magenta = reddish-purple  |
|         36 | \033[36m |         |       | cyan             | echo -e "\033[36m####\033[m" |                                      |
|         37 | \033[37m |         |       | white            | echo -e "\033[37m####\033[m" |                                      |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
|         38 | 8/24    |                    This is for special use of 8-bit or 24-bit                                          |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
| background | all below examples are with ommited color/text mode - default is 0                                                 |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
|         40 | \033[40m |         |       | black            | echo -e "\033[40m####\033[m" |                                      |
|         41 | \033[41m |         |       | red              | echo -e "\033[41m####\033[m" |                                      |
|         42 | \033[42m |         |       | green            | echo -e "\033[42m####\033[m" |                                      |
|         43 | \033[43m |         |       | yellow           | echo -e "\033[43m####\033[m" |                                      |
|         44 | \033[44m |         |       | blue             | echo -e "\033[44m####\033[m" |                                      |
|         45 | \033[45m |         |       | purple           | echo -e "\033[45m####\033[m" | real name: magenta = reddish-purple  |
|         46 | \033[46m |         |       | cyan             | echo -e "\033[46m####\033[m" |                                      |
|         47 | \033[47m |         |       | white            | echo -e "\033[47m####\033[m" |                                      |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
|         48 | 8/24    |                    This is for special use of 8-bit or 24-bit                                          |
|------------+---------+---------+-------+------------------+----------------------------+--------------------------------------|
```

foreground 4-bit summary in a .gif:

```
\033[30m#####\033[m
```

3/4 bit foreground bash color output

background 4-bit summary in a .gif

```
\033[40m#####\033[m
```

3/4 bit background bash color output

8 bit ANSI Colors

The below table shows a summary of **8 bit** version of ANSI color. where **$** is **color/text mode**

```
|------------+-----------+-----------+---------+------------------+------------------------------------+-------------------------|
| foreground | octal     | hex       | bash    | description      | example                            | NOTE                    |
|------------+-----------+-----------+---------+------------------+------------------------------------+-------------------------|
|        0-7 | \033[$;38 | \x1b[$;38 | \e[$;38 | standard. normal | echo -e '\033[0;38;1m####\033[m'   | red color               |
|       8-15 |           |           |         | underline        | echo -e '\033[4;38;10m####\033[m'  | light green             |
|     16-231 |           |           |         | reverse          | echo -e '\033[7;38;226m####\033[m' | yellow plus green       |
|    232-255 |           |           |         |                  | echo -e '\033[0;38;242m####\033[m' | from black to white     |
|------------+-----------+-----------+---------+------------------+------------------------------------+-------------------------|
| background | octal     | hex       | bash    | description      | example                            | NOTE                    |
|------------+-----------+-----------+---------+------------------+------------------------------------+-------------------------|
|        0-7 |           |           |         | standard. normal | echo -e '\033[0;48;1m####\033[m'   |                         |
|       8-15 |           |           |         |                  | echo -e '\033[0;48;9m####\033[m'   |                         |
|     16-231 |           |           |         |                  | echo -e '\033[0;48;45m####\033[m'  |                         |
|    232-255 |           |           |         |                  | echo -e '\033[0;48;242m####\033[m' | from black to white     |
|------------+-----------+-----------+---------+------------------+------------------------------------+-------------------------|
```

Here is quick test with 8bit color and underline text mode:

```
~] for code in {0..255}; do echo -e "\e[4;38;${code}m $code: Test"; done
```

linux bash in color

You can colorize output from echo , printf programs or you can colorize special bash varibles like PS0, PS1, PS2 (prompt) and PS4 also.

Echo and printf use a **octal ascii code** or **C escape sequence** for **ESC** char.

Here is end sequence with `octal` ***ESC*** escape sequence:

```
end="\033[0m"
```

> Remeber that `\033` is ASCII octal escape sequence. Maybe in another program you has to use another escape sequence for **ESC** char.

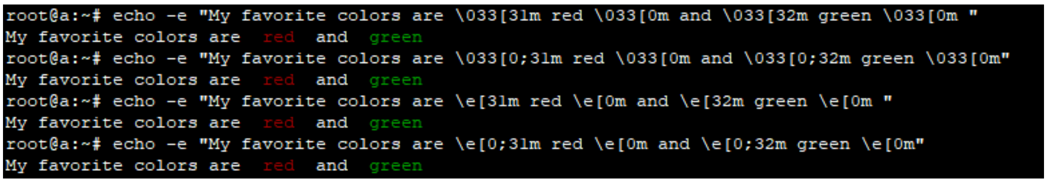
Examples

This 4 examples do same colored output:

```
echo -e "My favorite colors are \033[31m red \033[0m and \033[32m green \033[0m "
echo -e "My favorite colors are \033[0;31m red \033[0m and \033[0;32m green \033[0m"
echo -e "My favorite colors are \e[31m red \e[0m and \e[32m green \e[0m "
echo -e "My favorite colors are \e[0;31m red \e[0m and \e[0;32m green \e[0m"
```

- **\033** or **\e** - escape code for **ESC** char
- **[** - ESC character has to be followed by **[** (step 2)
- color/text mode is ommited - default is 0 or we type exactly 0 color/text mode
- **31** - **color-code** from step 3 for 3/4 bit ansi red color, **32** for green color
- **m** - m char from step 4
- **\033[0m** or **\e[0m** - ending sequence for **ESC** char

produce:



```
root@a:~# echo -e "My favorite colors are \033[31m red \033[0m and \033[32m green \033[0m "
My favorite colors are  red  and  green
root@a:~# echo -e "My favorite colors are \033[0;31m red \033[0m and \033[0;32m green \033[0m"
My favorite colors are  red  and  green
root@a:~# echo -e "My favorite colors are \e[31m red \e[0m and \e[32m green \e[0m "
My favorite colors are  red  and  green
root@a:~# echo -e "My favorite colors are \e[0;31m red \e[0m and \e[0;32m green \e[0m"
My favorite colors are  red  and  green
```

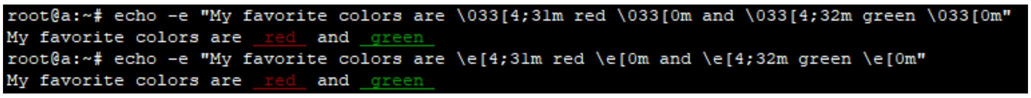
echo colored output

Example for red/green 3/4 bit ansi color code and underline text mode:

```
echo -e "My favorite colors are \033[4;31m red \033[0m and \033[4;32m green \033[0m"
echo -e "My favorite colors are \e[4;31m red \e[0m and \e[4;32m green \e[0m"
```

- **\033** or **\e** - escape code for **ESC** char
- **[** - ESC character has to be followed by **[** (step 2)
- **4** - color/text mode is number 4 - underline text
- **31** - **color-code** from step 3 for 3/4 bit ansi red color, **32** for green color
- **m** - m char from step 4
- **\033[0m** or **\e[0m** - ending sequence for **ESC** char

produce:



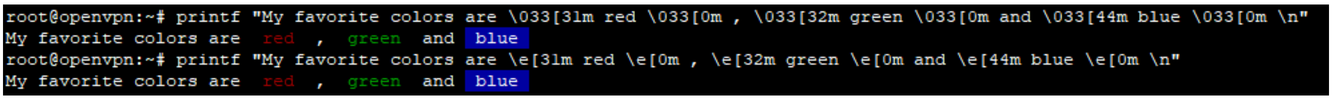
```
root@a:~# echo -e "My favorite colors are \033[4;31m red \033[0m and \033[4;32m green \033[0m"
My favorite colors are  red  and  green
root@a:~# echo -e "My favorite colors are \e[4;31m red \e[0m and \e[4;32m green \e[0m"
My favorite colors are  red  and  green
```

echo colored output 2

> If you are using the echo command, be sure to use the `-e` flag to allow backslash escapes.

The same is for **printf** utility:

```
~] printf "My favorite colors are \033[31m red \033[0m , \033[32m green \033[0m and \033[44m blue \033[0m \n"
~] printf "My favorite colors are \e[31m red \e[0m , \e[32m green \e[0m and \e[44m blue \e[0m \n"
```



```
root@openvpn:~# printf "My favorite colors are \033[31m red \033[0m , \033[32m green \033[0m and \033[44m blue \033[0m \n"
My favorite colors are  red  ,  green  and  blue
root@openvpn:~# printf "My favorite colors are \e[31m red \e[0m , \e[32m green \e[0m and \e[44m blue \e[0m \n"
My favorite colors are  red  ,  green  and  blue
```

colored printf output

bash exception

If you are going to use these colorized codes in your special bash variables

- PS0
- PS1
- PS2 - prompt
- PS4

you should add **extra escape characters** so that bash can interpret them correctly. You should add **[** before any starting ANSI code and add **]** after any ending ones.

example:

in regular usage: **\033[32mThis is in green\033[0m**

for PS0/1/2/4: **[\033[32m]This is in green[\033[m]**

- **[** is for start of a sequence of **non-printable** characters
- **]** is for end of a sequence of **non-printable** characters

sed colored output

Here is information from sed manual page:

```
\oxxx
```

```
     Produces or matches a character whose octal ASCII value is xxx.
```

So, for **ESC** char we have to this escape sequence: `\o033`

Here is another very important information for `&` char and `\1` through `\9` for sed substitution statement:

```
s/regexp/replacement/

     Attempt to match regexp against the pattern space. If successful, replace that portion matched with replacement. The replace
```

examples how colorize sed regex match

```
~] echo "one two three four five" | sed 's/two/\o033[1;31m&\o033[0m/g'
~] one two three four five

two - regex that we can try match
\o033 - octal ascci escape sequence for ESC char
[1;31m - color-mode 1 - lighter than normal and color-code 31 for red color in 3/4 bit ansi color code
& - special character & to refer to that portion of the regex pattern space which matched
[0m - end sequence
```

linux bash

« Previous article

Next article »

Architecture

Bash

Databases

Debian

SUBSCRIBE FOR NEW ARTICLES

@ Email Address

SUBSCRIBE

ALSO ON MYBLUELINUX-COM

Debian Linux and LSI MegaRAID SAS

3 years ago · 2 comments

napis description-o com je clanok

How Create Bootstrap Tables in Hugo

5 years ago · 2 comments

Do you know, how create very nice bootstrap tables in Hugo static …

NumPy I Visual

4 years

NumPy is library widely u

0 Comments Анатолий Суворин

Start the discussion…

♡ Share Best Newest Oldest

Be the first to comment.

Subscribe Privacy Do Not Sell My Data