

BASH Shell Change The Color of Shell Prompt on Linux or UNIX

Author: Vivek Gite
Last updated: October 29, 2020
[67 comments](#)

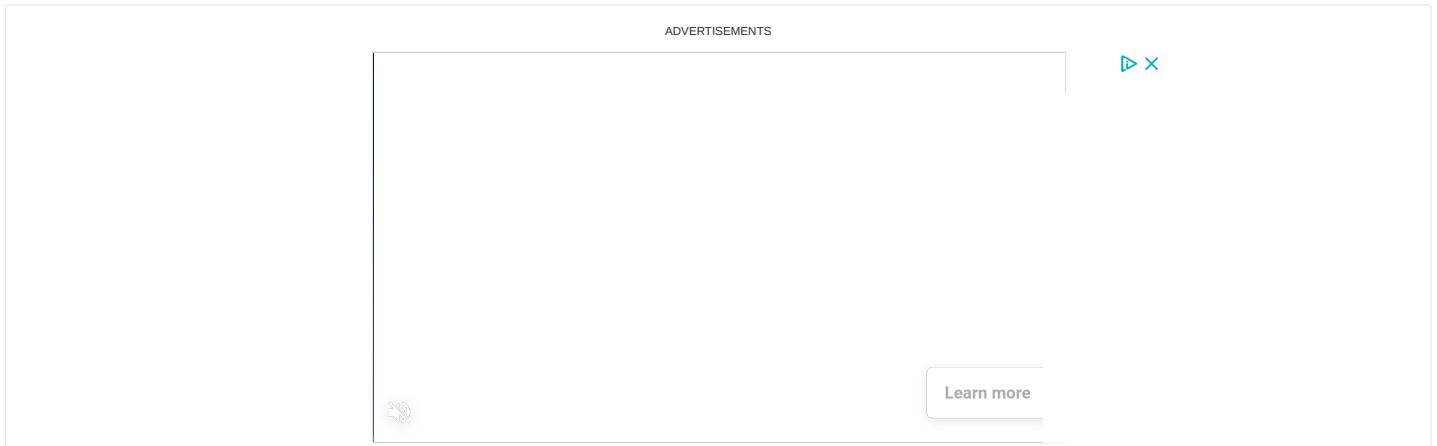
How do I change the color of my shell prompt under Linux or Unix operating systems? How do I customize and colorize my Bash prompt (PS1) on a Linux, macOS or Unix-like system?



You can change the color of your shell prompt to impress your friend or to make your own life quite easy while working at the command prompt. BASH shell is the default under Linux and Apple OS X. Your current prompt setting is stored in a shell variable called PS1. There are other variables too, like PS2, PS3 and PS4. Let us see how to change the color of shell prompt on a Linux or Unix system when using bash.

Tutorial details	
Difficulty	Easy (rss)
Root privileges	No
Requirements	Bash
Time	5m

Bash displays the primary prompt PS1 when it is ready to read a command, and the secondary prompt PS2 when it needs more input to complete a command. Bash allows these prompt strings to be customized by inserting a number of backslash-escaped special characters.



Task: Displaying current BASH prompt (PS1)

Use the [echo command](#)/[printf command](#) to display current BASH prompt settings:

```
$ echo "$PS1"
## OR ##
$ printf "%s\n" "$PS1"
```

Here is what I see:

```
[\\u@\\h \\w]\\$
```

Here is another output from my Debian based system:

```
$ echo $PS1
```

Ubuntu/Debian based Linux distro displayed the following:

```
\[\\e]0;\\u@\\h: \\w\\a\\}${debian_chroot:+($debian_chroot)}\\u@\\h:\\w\\$
```

By default the command prompt is set to `[\\u@\\h \\w]\\$` . The backslash-escaped special characters are decoded as follows:

- `\\u` : Display the current username .
- `\\h` : Display the hostname
- `\\w` : Print the base of current working directory.
- `\\s` : Display # (indicates root user) if the effective UID is 0, otherwise display a \$.

Task: Modifying current BASH prompt

Before you modify settings save your old prompt using the following command:

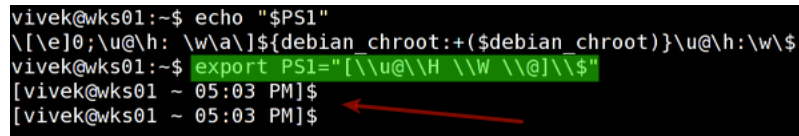
```
oldps1="$PS1"
```

So if you messed up, you can switch back easily using the following syntax:

```
PS1="$oldps1"
```

Use the export command to setup a new shell prompt:

```
$ export PS1="[\\u@\\H \\W \\@]\\$ "
```



```
vivek@wks01:~$ echo "$PS1"
\\[\\e]0;\\u@\\h: \\w\\a\\}${debian_chroot:+($debian_chroot)}\\u@\\h:~$
vivek@wks01:~$ export PS1="[\\u@\\H \\W \\@]\\$"
[vivek@wks01 ~ 05:03 PM]$
[vivek@wks01 ~ 05:03 PM]$
```

Fig.01: New prompt in action

Where,

- `\\H` : Display FQDN (fully qualified domain name) hostname.
- `\\@` : Display current time in 12-hour am/pm format.

Task: Adding colors to the prompt

To add colors to the shell prompt use the following export command syntax:

```
'\\e[x;ym $PS1 \\e[m'
```

Where,

- `\\e[` : Start color scheme.
- `x;y` : Color pair to use (x;y)
- `$PS1` : Your shell prompt variable.
- `\\e[m` : Stop color scheme.

Change the color of shell prompt by setting the PS1

To set a red color prompt, type the following export command:

```
$ export PS1="\\e[0;31m[\\u@\\h \\W]\\$ \\e[m "
```

```
vivek@wks01:~$ export PS1="\e[0;31m[\u@\h \W]\$ \e[m"
[vivek@wks01 ~]$
[vivek@wks01 ~]$
[vivek@wks01 ~]$ ## set to green ##
[vivek@wks01 ~]$ export PS1="\e[0;32m[\u@\h \W]\$ \e[m"
[vivek@wks01 ~]$
[vivek@wks01 ~]$
[vivek@wks01 ~]$ ## set to brown ##
[vivek@wks01 ~]$ export PS1="\e[1;33m[\u@\h \W]\$ \e[m"
[vivek@wks01 ~]$
[vivek@wks01 ~]$
```

Adding colors to the prompt

Fig.02: Adding the colors to the prompt

A list of color codes

Color	Code
Black	0;30
Blue	0;34
Green	0;32
Cyan	0;36
Red	0;31
Purple	0;35
Brown	0;33
Blue	0;34
Green	0;32
Cyan	0;36
Red	0;31
Purple	0;35
Brown	0;33

Note: You need to replace digit 0 with 1 to get light color version.

Task: How do I make the prompt setting permanent?

Your new shell prompt setting set by \$PS1 is temporary i.e. when you logout setting will be lost. To have it set every time you login to your workstation add above export command to your \$HOME/.bash_profile file or \$HOME/.bashrc file.

```
$ cd
$ vi .bash_profile
```

OR

```
$ vi $HOME/.bashrc
```

Append the following line:

```
export PS1="\e[0;31m[\u@\h \W]\$ \e[m"
```

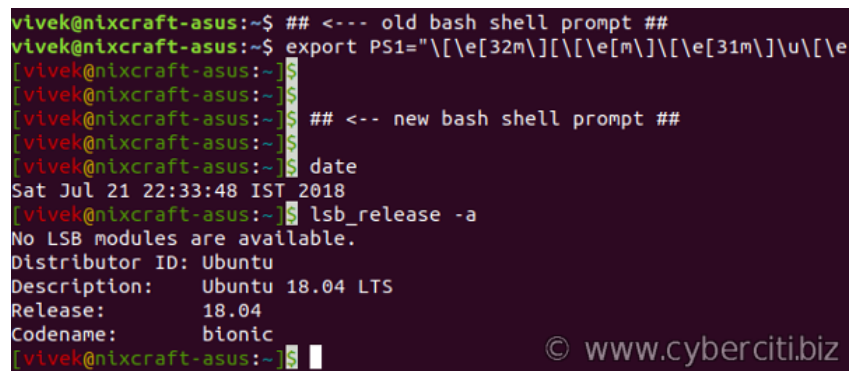
[Save and close the file in vim/vi.](#)

Bash Change The Color of Shell Prompt Example

Add the following command in ~/.bashrc:

```
# let us setup prompt
export PS1="\[\e[32m\][\[\e[m\]\[\e[31m\]\u\[\e[m\]\[\e[33m\]@\[\e[m\]\[\e[32m\]\h\[\e[m\]:\[\e[36m\]\w\[\e[m\]\[\e[32m\]\]\[\e[m\]\[\e[32;4'
```

You will get prompt as follows:



```
vivek@nixcraft-asus:~$ ## <--- old bash shell prompt ##
vivek@nixcraft-asus:~$ export PS1="\[\e[32m\][\[\e[m\]\[\e[31m\]\u\[\e[m\]\[\e[33m\]@\[\e[m\]\[\e[32m\]\h\[\e[m\]:\[\e[36m\]\w\[\e[m\]\[\e[32m\]\]\[\e[m\]\[\e[32;4'
[vivek@nixcraft-asus:~]$
[vivek@nixcraft-asus:~]$
[vivek@nixcraft-asus:~]$ ## <-- new bash shell prompt ##
[vivek@nixcraft-asus:~]$
[vivek@nixcraft-asus:~]$ date
Sat Jul 21 22:33:48 IST 2018
[vivek@nixcraft-asus:~]$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04 LTS
Release:        18.04
Codename:       bionic
[vivek@nixcraft-asus:~]$
```

© www.cyberciti.biz

Here is another example that include exit status of command as well:

```
#
#\u - user name
#\h - short hostname
#\W - current working dir
#\? - exit status of the command
export PS1="{\[\e[32m\]\u\[\e[m\]@\[\e[36m\]\h\[\e[m\]:\W_\$?\}$ " "
```

Say hello to tput command

You can also use tput command to set terminal and modify the prompt settings. For example, to display RED color prompt using a tput:

```
export PS1="\[$(tput setaf 1)\]\u@\h:\w $ \[$(tput sgr0)\]"
```

However, we do not hard-code ANSI color escape sequences. Hence we use the tput command as follows:

```
_GREEN=$(tput setaf 2)
_BLUE=$(tput setaf 4)
_RED=$(tput setaf 1)
```

```
_RESET=$(tput sgr0)
_BOLD=$(tput bold)
export PS1="${_GREEN}\h${_BLUE}@${_RED}\u${_RESET} ${_BOLD}\$ ${_RESET}"
```

Here is a list of handy tput command line options

- `tput bold` – Bold effect
- `tput rev` – Display inverse colors
- `tput sgr0` – Reset everything
- `tput setaf {CODE}` – Set foreground color, see color {CODE} table below for more information.
- `tput setab {CODE}` – Set background color, see color {CODE} table below for more information.

Various color codes for the tput command

Color {code}	Color
0	Black
1	Red
2	Green
3	Yellow
4	Blue
5	Magenta
6	Cyan
7	White

Customize bash colors prompt content in Linux or Unix terminal

My current PS1 settings from the ~/.bash_aliases file on Ubuntu Linux desktop displayed using the [cat command](#):

```
$ cat ~/.bash_aliases
```

Config that changes the color of shell prompt as per my needs along with the Tux ASCII logo.

```
export PS1="\[\e[31m\][\[\e[m\]\[\e[38;5;172m\]\u\[\e[m\]@\[\e[38;5;153m\]\h\[\e[m\] \[\e[38;5;214m\]\W\[\e[m\]\[\e[31m\]\]\[\e[m\]\[\e[31m\]\$ "
cat<<'EOF'
```

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 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
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

EOF

Which will produce:

My customized and colorized bash prompt


Conclusion

You learned how change the color of shell prompt under Linux and Unix when using bash. For more information see the following resources:

- [HowTo: Change or setup your bash custom prompt \(PS1\)](#) – This tutorial explains how-to customizing a bash shell to get a good looking prompt, configure the appearance of the terminal and apply themes using bashish package.
- man page – bash

This entry is **5 of 5** in the **Coloring Outputs in the CLI Apps Tutorial** series. Keep reading the rest of the series:

1. [How to enable colorized output for ls command in MacOS X Terminal](#)
2. [How To Turn On/Off Colors For ls Command In Bash On a Linux/Unix](#)
3. [Turn off color in Linux terminal/bash session](#)
4. [Turn On or Off Color Syntax Highlighting In vi or vim Editor](#)
5. BASH Shell: Change The Color of My Shell Prompt Under Linux or UNIX

 If you liked this page, please **support my work** on [Patreon](#) or with a [donation](#).

 Get the latest tutorials on SysAdmin, Linux/Unix, Open Source/DevOps topics:

- [RSS feed](#) or [Weekly email newsletter](#)
- Share on [Twitter](#) • [Facebook](#) • 67 comments... [add one](#) ↓

UP NEXT


[Ubuntu Linux: Install Doom3 Game](#)


Essentials Back In Stock Walgreens Pharmacy at P


[Linux Network IP Accounting](#)

[Python Command Line Arguments Examples](#)

[Linux/UNIX: Bash Read a File Line By Line](#)

[RHEL / Fedora / CentOS Linux: Set RPM Build Environment](#)

[FreeBSD csh / tcsh - Export Shell Variable](#)

[CentOS / Redhat Iptables Firewall Configuration Tutorial](#)

Category	List of Unix and Linux commands
File Management	cat
Firewall	Alpine Awall • CentOS 8 • OpenSUSE • RHEL 8 • Ubuntu 16.04 • Ubuntu 18.04 • Ubuntu 20.04
Network Utilities	dig • host • ip • nmap
OpenVPN	CentOS 7 • CentOS 8 • Debian 10 • Debian 8/9 • Ubuntu 18.04 • Ubuntu 20.04
Package Manager	apk • apt
Processes Management	bg • chroot • cron • disown • fg • jobs • killall • kill • pidof • pstree • pwdx • time
Searching	grep • whereis • which
User Information	groups • id • lastcomm • last • lid/libuser-lid • logname • members • users • whoami • who • w
WireGuard VPN	Alpine • CentOS 8 • Debian 10 • Firewall • Ubuntu 20.04

ADVERTISEMENTS

67 comments... [add one](#) ↓

Martin Norbäck

Jan 12, 2007 @ 16:35

you also need to put `\[` and `\]` around any color codes so that bash does not take them into account when calculating line wraps. Also you can make use of the `tput` command to have this work in any terminal as long as the `TERM` is set correctly. For instance `$(tput setaf 1)` and `$(tput sgr0)`

[reply](#) [link](#)

Charly

Aug 13, 2012 @ 9:00

Thx Martin

The `\[` and `\]` around color codes are really useful.

When going through the history, the size of the prompt was incorrectly calculated and I had characters from a previous command that would add to the prompt. Thanks to you, this error is now corrected.

As an example here is my `PS1`:

```
PS1='\[\e[0;32m\]\u@\h:\w\$ \[\e[0m\]'
```

[reply](#) [link](#)

Tony

Feb 9, 2015 @ 16:55

Thank You! Been trying to figure out how to do that for years.

[reply](#) [link](#)

Xavier59

Jan 23, 2017 @ 19:24

Thanks, still usefull in 2017 ! :D

+ 1

[reply](#) [link](#)

Josh

Jan 17, 2017 @ 2:55

Thank you Martin. I've spent ages trying to fix an annoying behavior in my terminal when lines wrapped over to the next line, and/or I hit backspace in some situations.

I'd not wrapped my color codes in ``\['`` or ``\['``, and now they're fixed! Thank you!

[reply](#) [link](#)

sandeep Kumar

Jul 7, 2007 @ 5:33

The way to costumise shell prompt is very impressive. I WANT TO CHANGE BACKGROUND COLOR ALSO GIVE SOME IDEA.

[reply](#) [link](#)

T. King

Jul 11, 2007 @ 20:08

Using the first example did cause the first line wrap to simple start back over the first line again, creating issues when entering a two-line command. I'm sure using #1's suggestion about using `\['`` and `\['`` around color codes would work, but I settled on using the `tput` instead, which fixed the line-wrap issue.

[reply](#) [link](#)

sanjith

Aug 7, 2007 @ 12:57

This topic was winderful. it is very easy to learn and simple

[reply](#) [link](#)

Mark

Oct 22, 2007 @ 3:08

Can anyone tell me why after adding the color codes, once text has gone to the next line, i cant backspace back to the 1st line, it just stops at the far left of second line?

[reply](#) [link](#)

Johu

May 9, 2013 @ 10:13

Read the first comment. I had the same issue.

[reply](#) [link](#)

Richard Burford

Dec 7, 2007 @ 14:21

I've looked at three other articles on colourising my bash prompt and this is the most clear and understandable of them all.

Thanks!

[reply](#) [link](#)

ramigs

Dec 24, 2007 @ 1:35

@Martin Norbäck

thank you. that's the first time i see an explanation for that.

[reply](#) [link](#)

Ruben

Dec 30, 2007 @ 18:14

After read some articles here is my way:

```
# ==== alias for colors =====  
BLACK="tput setf 0"  
BLUE="tput setf 1"  
GREEN="tput setf 2"  
CYAN="tput setf 3"  
RED="tput setf 4"  
MAGENTA="tput setf 5"  
YELLOW="tput setf 6"  
  
WHITE="tput setf 7"
```

```
RETURN="tput sgr0"  
BOLD="tput bold"  
REV="tput rev"
```

Some examples of use

```
PS1="`$REV``$RED`[$netip ` $BLUE`/\w]#`$RETURN` "
```

```
PS1="`$REV``$BLACK`[`$RED`$netip ` $BLUE`\w`$BLACK`]#`$RETURN` "
```

```
PS1="`$REV`[`$RED`$netip ` $BLUE`\w]#`$RETURN` "
```

```
PS1="`$REV``$RED`[$netip ` $BLUE`\w]#`$RETURN` "
```

where netip is

```
netip=`/sbin/ifconfig eth0 | awk -F: '/inet addr/ {print $2}' | awk '{print $1}' | cut -b 11-`;
```

I hope someone find it useful.

Regards, Ruben.ie

[reply](#) [link](#)

ramana

Mar 4, 2008 @ 7:07

how to change promt(% to \$) in unix ?

[reply](#) [link](#)

buu700

Jun 13, 2008 @ 18:47

Just use this command:

```
export PS1="\e[0;31m$(echo PS1)\e[m"
```

That way, all you're doing is adding the color code around your current PS1 variable (completely non-confusing and non-non-working).

[reply](#) [link](#)

matelot

Jul 7, 2008 @ 3:15

> buu700 Says:

> Just use this command:

hey a***** – test the command before posting !

```
export PS1="\e[0;31m$(echo $PS1)\e[m"
```

[reply](#) [link](#)

R.I.M. Rizme

Sep 4, 2008 @ 3:24

```
$ export PS1="\e[0;31m[\u@\h \w]\$ \e[m "
```

try this

[reply](#) [link](#)

MacUser

Dec 3, 2008 @ 1:41

On a mac, the tget bold command doesn't use the bold color from your terminal preferences. But you can get around this annoyance by using

```
\[\e[1m\] \[e[m\]
```

Which will send the ASCII bold character and terminal respects that.

[reply](#) [link](#)

strnik

Jan 13, 2009 @ 13:07

all examples above did not work properly in bash on Solaris 10 for me, so this is what I ended up with :

```
if [ `hostname|cut -c -11` == lonlnddebt ]; then
    PS1='\[\e[1;31m\][\u@\h:\w]\$'\[\e[0m\] ' # PROD, red color
elif [ `hostname|cut -c -11,13-` == lonlnddebt-d-z1 ]; then
    PS1='\[\e[2;32m\][\u@\h:\w]\$'\[\e[0m\] ' # UAT, green color
else
    PS1='[\u@\h:\w]\$ '
fi
```

[reply](#) [link](#)

Andres

Jan 31, 2009 @ 1:56

I use this to set my prompt

```
set prompt = "%B%{33[31m%}%m %{33[37m%}%B%/ / \n"
```

It placed the machine name in orange color, and the path in white. It is separated by a space, so I can easily select the path by double clicking it...and I also included a / to allow me to append anything else afterwards

[reply](#) [link](#)

Amira

Feb 18, 2009 @ 6:47

hi,..

sry but what if i want to change the colours of the command it self

like (ls -l /etc/passwd)i want for example(ls) in red and (-l) with blue and (the name of file) with green

[reply](#) [link](#)

penduleum

Feb 21, 2009 @ 19:35

hey all.

What line do i need to use to get the text in green, whit a orange hostname?

thx in advanced,

penduleum

[reply](#) [link](#)

Lionel.rpm

May 8, 2009 @ 0:46

```
export PS1="\[$(tput setaf 1)\]\u@\h:\w $ \[$(tput sgr0)\]"
```

```
\u@\h:\w $ \[ Why would i use $ for root?
```

means? \u@\h:\w\\\$ \[—> this is the best way.

Nice work Vivek

;)

[reply](#) [link](#)

Purav

May 15, 2009 @ 14:04

this helped me. many thanks.

[reply](#) [link](#)

Black_Ps`

Jun 11, 2009 @ 3:35

```
export PS1="\[[33[01;32m\]\u@\h\[[33[01;34m\] \w]\]"
```

how can i execute this command from a .sh script or .c and make it work,can somebody help.

Thanks.

[reply](#) [link](#)

Saurabh

Aug 13, 2009 @ 7:33

Hi,

I am using cygwin. I could not find the files .bash_profile file or .bashrc file anywhere in installation directory. I want to change color parmanent.

[reply](#) [link](#)

Sassy Grrrl

Jun 9, 2011 @ 14:52

Files that start with a . are invisible if you just run a regular ls command you can't see them so run ls -la instead. Also, even when you don't see them you can still edit them using your favorite editor by just making sure that when you load the file you use the dot. For example: emacs .bash_rc

[reply](#) [link](#)

oinkboing

Sep 16, 2009 @ 12:01

Replace for example Color 34 with 44.

(4)4 = Background

```
green_bg_color="\E[0;37;42m"
```

[reply](#) [link](#)

Jigsaw

Sep 27, 2009 @ 19:55

What do 0 and 31 in 0;31m indicate? Is there a way of specifying color as RGB values?

[reply](#) [link](#)

SS

Feb 8, 2010 @ 22:38

For KSH ...

1 example is :

```
FGWHITE=`echo "33[1;37m"``  
echo "${FGWHITE}"
```

to back to normal :

```
NORMAL=`echo "33[m"``  
echo "${NORMAL}"
```

[reply](#) [link](#)

Tynged

Apr 14, 2010 @ 19:57

I figured out how to color parts separately. Here's what I used:

```
export PS1="\[\e[0;33m\][\[\e[0;32m\]\u\[\e[0;33m\]@\h:\[\e[0;39m\]\w\[\e[0;33m\]]\$\[\e[0m\] "
```

To get the neon colors in the video linked above I assume you'd have to change the precise display colors used by your ssh client. So, using PuTTY for example, you'd have to modify the RGB values of each ANSI color in the Window->Colours menu.

[reply](#) [link](#)

Matt

Jun 17, 2011 @ 19:20

Thanks for this. Don't forget to change the curly quotes to regular straight quotes "". The shell only know knows how to use regular quotes. So it needs to look like this.

```
export PS1="\[\e[0;33m\][\[\e[0;32m\]\u\[\e[0;33m\]@\h:\[\e[0;39m\]\w\[\e[0;33m\]]\$\[\e[0m\] "
```

[reply](#) [link](#)

Cq

Nov 11, 2011 @ 17:20

I do not see the answer to an earlier questio: Is it possible to simply reverse everything on the screen, including the normal outputs of utilities and scripts. By default the background is black with foreground white. I wish to reverse this everywhere to reduce reflections on the screen. Any way?

[reply](#) [link](#)

Richie

Feb 8, 2012 @ 17:10

You will want to modify the terminal parameters. If you are using xterm or putty, etc, change it there. If you are using true terminal no gui then this i am afraid may not be fully possible.

[reply](#) [link](#)

Nat Harward

Feb 8, 2012 @ 23:17

Just type: tput rev

However, if you run any program that manipulates the terminal settings (like running "ls" with the "--color" option, for example) it will probably get reset at that point. To make it permanent you'd need to adjust the settings of whatever your terminal program is (as Richie says below).

[reply](#) [link](#)

maseny

May 3, 2010 @ 15:44

i want to modify the way dirctory a creaded in bash

[reply](#) [link](#)

aciD

May 16, 2010 @ 13:26

The best PS1 :P :

```
PS1='\[33[1;33m\]\u\[33[1;37m\]@\[33[1;32m\]\h\[33[1;37m\]: \[33[1;31m\]\w\n\[33[1;36m\]\$ \[33[0m\]'
```

(dont forget to change it in .bashrc in all home folders include /root ;))

[reply](#) [link](#)

Stefan Lasiewski

Jun 8, 2010 @ 4:51

If you use `export PS1="\[${tput setaf 1}\]\u@h:\w $ \[${tput sgr0}\]"`, then won't this be forking 'tput' every time PS1 is set? The BashFAQ at <http://mywiki.woledge.org/BashFAQ/037?highlight=%28ps1%29> suggests doing this instead:

```
# Bash
red=$(tput setaf 1)
green=$(tput setaf 2)
blue=$(tput setaf 4)

reset=$(tput sgr0)
PS1='\[${red}\]\u\[${reset}\]@\[${green}\]\h\[${reset}\]:\[${blue}\]\w\[${reset}\]\$ '
```

[reply](#) [link](#)

Fabiano PS

May 25, 2011 @ 20:14

Thanks! Finally set with one scheme that does not break my lines using CTRL+R ou checking history!

```
green=$(tput setaf 2)
blue=$(tput setaf 4)
reset=$(tput sgr0)
PS1='\[$green\]\u\[$reset\] \[$blue\]\w\[$reset\] \$ '
```

[reply](#) [link](#)

Stefan Lasiewski

Jun 8, 2010 @ 17:55

Vivek, this is great stuff as always. However, if PS1 will run the \$(tput foo) command, won't it fork the tput commands every time PS1 is displayed?

The suggestion at <http://mywiki.woledge.org/BashFAQ/037?highlight=%28ps1%29> is to store the tput output in environment variables, and use those variables:

```
# Bash
red=$(tput setaf 1)
green=$(tput setaf 2)
blue=$(tput setaf 4)
reset=$(tput sgr0)
PS1='\[$red\]\u\[$reset\]@\[$green\]\h\[$reset\]:\[$blue\]\w\[$reset\] $ '
```

[reply](#) [link](#)

Christian Haarmann

Jun 18, 2010 @ 15:04

You can give three arguments, not only two!

"\e[x,y,zm" with x=brightness, y=foreground, z=background. For example:

"\e[0;33;40m" shows dark (=0) yellow (=33) characters on black (=40) background

The alternate way using "tput" doesn't work in my cygwin and prints the error message:

tput: unknown Terminal "xterm-color"

[reply](#) [link](#)

Stefan Lasiewski

Jun 24, 2010 @ 0:20

Sadly, tput on FreeBSD 7.x doesn't support these fancy options. The tput man page has no mention of options like 'bold' and 'color'. This works on Linux.

```
$ echo $TERM
```

xterm-color

```
$ echo "$(tput bold)BOLD$(tput sgr0)"
```

BOLD

[reply](#) [link](#)

Michael

Aug 2, 2010 @ 16:01

Can we change the colors of terminal that is appearing on the booting?

[reply](#) [link](#)

Albert

Oct 25, 2010 @ 13:44

Hi, i think there is a little mistake.

Tried to add the initial and ending colors mark and being unsuccessful, i've found that the complete color start mark is `\e[x;ym`
Note the final m, which in my opinion is not clearly explained above.

Great tutorial anyway, thx! ;)

[reply](#) [link](#)

Rohit

Feb 15, 2011 @ 20:22

I changed the color of my Linux prompt successfully. But now the commands that I run do not carry forward to the same line. If I have a `tar -zcvf` command with 5 or 6 log files, the command rolls over to the same line. Its so confusing that I cant see the first part of my own command. Is there a way out of this?

Thanks guys...

[reply](#) [link](#)

Jin

May 22, 2011 @ 18:19

Yes, I have exactly the same problem with my Mac OS X:

My PS1 is:

```
PS1='\e[0;31m[\u@h \w]\$ \e[m'
```

And then, if my command line is really long, it wraps around and overwrites the prompt; and if the command line is super long and it wraps around the second time, this time it would correctly push the terminal output one line up and start to print in the new line.

Help please~

[reply](#) [link](#)

Jin

May 22, 2011 @ 18:26

Duh...10 seconds after I post the question, I figured out. Basically any sequence like "\e[32m" needs to be enclosed by "[" and "]", just like Tynged mentioned below.

```
PS1='\[\e[0;31m\][\u@h \w]\$ \[\e[m\]'
```

[reply](#) [link](#)

Matt

Jun 17, 2011 @ 18:45

I had to use double quotes "" instead of back quotes ". When I tried back quotes I got this message.

```
$ export PS1='\[\e[0;31m\][\u@h \w]\$ \[\e[m\]'
```

bash: 31m][u@h: command not found...

Failed to search for file: Invalid input passed to daemon: char ']' in text!

```
'[e[0
```

When I used this it works perfectly.

```
export PS1="\[\e[0;31m\][\u@h \w]\$ \[\e[m\"]"
```

[reply](#) [link](#)

sangdol

Feb 25, 2012 @ 15:19

@Jin Thanks a lot! Author should've write about that.

[reply](#) [link](#)

Nat Harward

Apr 1, 2011 @ 22:42

This thread is old but if anyone is still reading it, you probably want to change "tput" with " `tput -T${TERM:-dumb}` " so that if by some chance you are *not* in a nice TTY you won't get garbage characters.

[reply](#) [link](#)

Nat Harward

May 28, 2011 @ 3:04

This inspired me to create Baven [<https://github.com/nharward/baven>] by not ever wanting to write "tput" ever again :)

It is a plugin loader for BASH allowing to dynamically load BASH functions, over the network if needed (similar to Maven plugins). One is there specifically to handle ANSI colors, see <https://github.com/nharward/baven/blob/master/bin/samples/ansi-colors.bash> for a working example.

If anyone has suggestions for making it better, I'm all ears...

[reply](#) [link](#)

Matt

Jun 17, 2011 @ 19:41

These 2 methods work best for me.

```
export PS1='\[33[0;32m\][\u@\h \w]\$\[33[0m\] '
export PS1="\[ \e[0;31m\][\u@\h \w]\$ \[ \e[m\]"
```

These are some good references.

<http://www.ibm.com/developerworks/linux/library/l-tip-prompt/>

<http://tldp.org/HOWTO/Bash-Prompt-HOWTO/x329.html>

[reply](#) [link](#)

Thibaut

Aug 5, 2011 @ 8:35

That was helpful, thanks.

[reply](#) [link](#)

JJ

Nov 27, 2011 @ 9:03

I just want to thank you for having a simple, concise explanation after each bash expression.

Too often guides simply say: "Paste X at the bottom of file Y. Got that? Ok, GTFO"

[reply](#) [link](#)

pradeep

Jul 3, 2012 @ 9:17

Thanks, working fine.

[reply](#) [link](#)

g

Dec 7, 2012 @ 7:08

On Ubuntu 12.04.1 LTS (precise, xubuntu) bash uses ~/.bashrc

This file has settings to modify the prompt and colors.

By default the colors are off and it explains why:

```
# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
# force_color_prompt=yes
```

So to turn the colors on: remove the octothorpe in front of force_color_prompt

[reply](#) [link](#)

splatter

Sep 17, 2013 @ 21:34

THANK YOU! That prompt over write was driving me crazy, and I had a hard time finding anything on it until today I finally plugged in the right terms in google. You guys rock!

[reply](#) [link](#)

Heith

Jul 23, 2014 @ 19:39

Simple with color:

```
export PS1="\[$(tput setaf 5)\]\h:\[$(tput setaf 6)\]\W\[$(tput sgr0)\]$ "
```

[reply](#) [link](#)

sidsingh

Nov 12, 2015 @ 14:00

Thanks for putting this together. This is exactly what I was looking for.

[reply](#) [link](#)

spike

Jan 7, 2016 @ 10:40

Thanks, really useful :)

Just a note, you have repeated some colors in the table (red, purple, brown).

[reply](#) [link](#)

Dan

Jan 7, 2016 @ 14:47

Instead of this use `setterm` e.g. `setterm -back (colour)` then type `clear` to change the entire background. to change text do `setterm -fore (colour)`

[reply](#) [link](#)

Doug Stevens

Feb 1, 2016 @ 16:04

Thanks for the nice docs.

In "A list of color codes", each foreground color (0;30 to 0;37) is listed twice. It would be nice if they were in numeric order, followed by the list of background colors (0;40 to 0;47).

[reply](#) [link](#)

yourprosecutor

Mar 6, 2016 @ 9:29

Man, edit this article to include the `\[` and `\]` enclosing braces when the color codes are! I was in the dumps trying to figure why the hell my terminal doesn't wrap long lines!

A big -1 !!

[reply](#) [link](#)

Mohsen Pahlevanzadeh

Mar 15, 2016 @ 22:06

What's blink code without `tput`? Would you please example me with `echo` ?

[reply](#) [link](#)

omkar

Apr 16, 2016 @ 12:32

Nice work

[reply](#) [link](#)

Kevin

Mar 6, 2017 @ 10:00

Please edit this command:

```
export PS1="\e[0;31m[\u@\h \W]\$ \e[m "
```

to contain '[' and ']' allowing for correct placement of the cursor for multiple line commands.

Following your example led me to have to having issues with two line commands for the last month until I finally figured out what the problem was and how to fix it.

[reply](#) [link](#)

name1

Oct 29, 2020 @ 11:44

Kaco was impressed, works 1/1 times, would change color again.

[reply](#) [link](#)

Leave a Reply

Your email address will not be published.

Comment

Name

Email

Website

Post Comment

Use HTML `<pre>...</pre>`, `<code>...</code>` and `<kbd>...</kbd>` for code samples.

Next FAQ: [Blank screen during a Linux graphical installation](#)

Previous FAQ: [Linux Set Date and Time From a Command Prompt](#)

©2020 nixCraft • [Privacy](#) • [ToS](#) • [Contact/Email](#) • [Search](#) • Sponsored by [Linode](#)

