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

Author: Vivek Gite • Last updated: August 6, 2024 • 68 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. 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.

Tutorial details	
Difficulty level	Easy
Root privileges	No
Requirements	Linux or Unix terminal

Tutorial details	
Category	Linux shell scripting
OS compatibility	BSD • Linux • macOS • Unix
Est. reading time	4 minutes

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.
- `\$`: Display # (indicates root user) if the effective UID is 0, otherwise display a $.

Task: Modifying the 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:

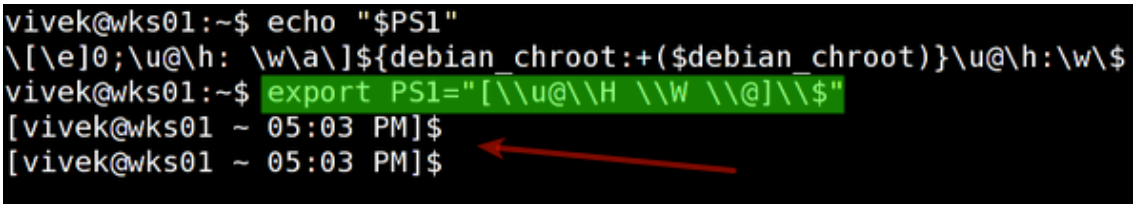

Fig.01: New prompt in action

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 "
```

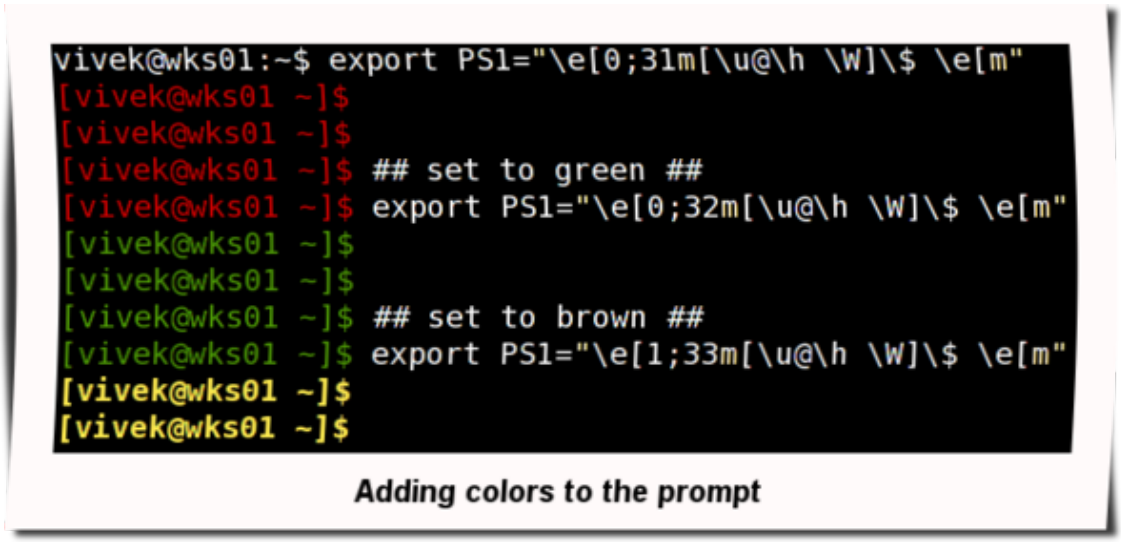

Fig.02: Adding the colors to the prompt

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. Change directory using the cd command:

```
$ 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 by pressing the `Esc+:+x!`.

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
```

You will get prompt as follows:

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 the 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\[\e[m\]@\[\e[38;5;153m\]
cat <<'EOF'
           _-···-_
         .'       '.
        /  _     _  \
        | (o)_(o) |
         \(       ) /
         //'._.'\ \
        //    .    \ \
       ||     .     \ \
       |\     :     / |
       \ `) '    (`  /_
     _)``".____,.'"` (_
     )      )'--'(     (
```