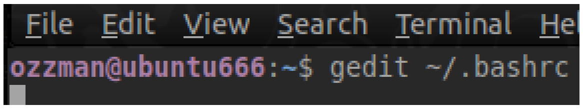
Set Raspbian/Debian Command-Line Prompt Colors

by: Ubuntu Genius July 11, 2011

The terminal for running commands in **Rasbian** and any other **Linux** distributions is dull to look at, but you can easily spice it up a little with some color.



```
ozzman@ubuntu666:~$ gedit ~/.bashrc
```

Simply run the following command in a terminal:

```
nano ~/.bashrc
```

When **.bashrc** opens, locate and *uncomment* “**force_color_prompt=yes**” (that is, remove the *hash*, so it no longer looks like: **#force_color_prompt=yes**).

Save the file, and open a new terminal window, and you should already see a change (the prompt should be ***Light Green***, which is defined by **1;32**). You can then change any color value you like; eg: **0;35 = *Purple***.

To edit the color values, locate the following section, and change the default values with some of the examples listed further down:

```
if [ "$color_prompt" = yes ]; then
PS1=’${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;31m\]\w\[\033[00m\]\$ ‘
else
PS1=’${debian_chroot:+($debian_chroot)}\u@\h:\w\$ ‘
fi
```

You can **check out this Bash color chart** for a full range of color values, but here are a few basic ones you can play around with (note that “Light” isn’t what you might think – it actually means “bold”):

Black **0;30** – ***Dark Gray*** **1;30** – ***Blue*** **0;34** – ***Light Blue*** **1;34** – ***Green*** **0;32** – ***Light Green*** **1;32** – ***Cyan*** **0;36** – ***Light Cyan*** **1;36** – ***Red*** **0;31** – ***Light Red*** **1;31** – ***Purple*** **0;35** – ***Light Purple*** **1;35** – ***Brown*** **0;33** – ***Yellow*** **1;33** – ***Light Gray*** **0;37** – ***White*** **1;37**

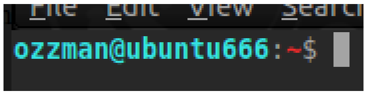
For those curious about the codes used in the example pic, here’s the line from that section:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;35m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
```

As you can see, **1;35** is the ***Light Purple*** user and machine name, while the **1;34** is the ***Light Blue*** *tilde* (~). If you want yours a bit brighter, try:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;36m\]\u@\h\[\033[00m\]:\[\033[01;31m\]\w\[\033[00m\]\$ '
```

… which will give you ***Light Cyan*** and ***Light Red***, and look like the following:

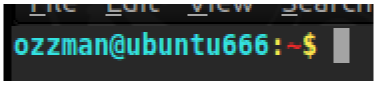


```
ozzman@ubuntu666:~$
```

In case you're wondering about the colon and dollar sign, you can change those as well, but you need to do more than just edit the color values. You'll need to insert code in the appropriate places, so the line looks like this:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;36m\]\u@\h\[\033[01;33m\]:\[\033[01;31m\]\w\[\033[01;33m\]\$ '
```

You'll notice the first highlighted code is just before the colon (**:**) while the second is before the dollar sign (**$**). In this example, both are yellow, with the result looking like:

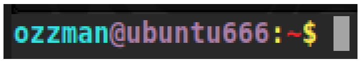


```
ozzman@ubuntu666:~$
```

Now, if you want to go even further, you can make the user name stand out by doing the following:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;36m\]\u\[\033[01;35m\]@\h\[\033[01;33m\]:\[\033[01;31m\]\w\[\033[01;33m\]\$ '
```

That extra bit of code is specifying ***Light Purple*** for the **@** and the desktop name, making it now look like:

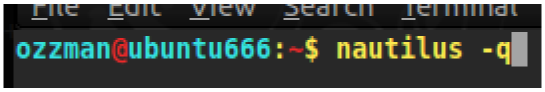


```
ozzman@ubuntu666:~$
```

And of course, one last bit of fiddling and you can have every element a different color. In this last example, we're going to make the computer name the same as the user name, and have them broken up by red, as with the yellow elements:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;36m\]\u\[\033[01;31m\]@\[\033
[01;36m\]\h\[\033[01;33m\]:\[\033[01;31m\]\w\[\033[01;33m\]\$ '
```

As you can see, the **@** is now ***Light Red***, while the code before **\h** is specifying ***Light Cyan***, like the user name:



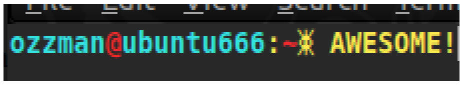
```
ozzman@ubuntu666:~$ nautilus -q
```

You'll also notice when you type commands that the color of the text will match that of the **$**, which can be preferable if using a "light" color, since the bold text is easier to see.

Lastly, in case you're wondering whether the prompt can end in anything other than a **$**, the answer is yes, and it's as easy as opening the **Character Map** (`sudo apt-get install gucharmap` if you don't have it), selecting a character, and pasting it over the **$** at the end of the line of code:

```
PS1='${debian_chroot:+($debian_chroot)}\[\033[01;36m\]\u\[\033[01;31m\]@\[\033
[01;36m\]\h\[\033[01;33m\]:\[\033[01;31m\]\w\[\033[01;33m\] '
```

In that example, I simply selected a cool looking character from the font *Runic*, and replaced the **$** with that. You'll also note one other thing you'll have to do, and that's remove the **** before it, or else that will appear too (obviously, that doesn't happen if using the **$**, but will with other characters).



```
ozzman@ubuntu666:~ᛯ AWESOME!
```

Have fun experimenting!

The actual web page can be found at the URL below:

http://ubuntugenius.wordpress.com/2011/07/11/how-to-change-the-command-line-prompt-colour-in-the-ubuntulinux-terminal/

Comments:

Arun

Yes, I've noticed when you type commands that the color of the text will match that of the $. But then, for commands like ls, the output continues in the color of the $. How to change back to default text color for the output of the command typed? Thanks.

----

on March 29, 2012 at 9:40 am | **Reply Ubuntu Genius**

I just tried it on my system, and ls output is how it should be, with none of the colors matching the prompt. My prompt looks exactly like the very last example, with bold yellow on a dark background, and ls output is in white and pale blue, so I'm not sure what is happening your way, as I can't recreate it. For other commands that spit out data, like du or running apt-get update, etc, the output will be the same as the prompt color, which it always has been. But ls should be an exception, so if you find the answer, please post it here.

----

on March 29, 2012 at 10:06 am | **Arun**

What I notice with ls or du is that the color of the output continues to be in the color of the prompt. However, for ls, the output takes the default colors set by **CLICOLOR** and **LSCOLORS** after the first instance of a directory name appearing in the output which is set to a different color.

So, in the following code for a very colorful prompt, I know that if I add **\[\e[0m\]** after the last **$**, it will terminate the overflow of the prompt color. However, that would also disable the color of the typed in command after the prompt. I was wondering if there is an option that can go between the **$** and **\[\e[0m\]** to set the color of the command typed in.

```
function prompt
 {
 local GREEN=”\[33[0;32m\]”
 local CYAN=”\[33[0;36m\]”
 local BLUE=”\[33[0;34m\]”
 local RED=”\[33[1;31m\]”
 export PS1=”
 \[\e]2;\u@\h\a\e[0;31m${GREEN}\u${CYAN}@${BLUE}\h
${CYAN}\w${RED}
 $ ”
 }
 prompt
```

----

on December 28, 2012 at 4:35 pm | **zef**

add the line below after the **export PS1=”** line

trap ‘echo -ne “\e[0m”‘ DEBUG

----

on June 4, 2012 at 5:52 pm | **Reply Gabriel**

Hi. When you change the color and then you type su, the change done before do not take effect like superuser, how can I fix that ?

----

on November 22, 2012 at 4:35 am | **Reply marwa**

Thank you soooo much

----

on February 4, 2013 at 9:31 pm | **Reply Stefan**

hmmm, it doesn’t work!
