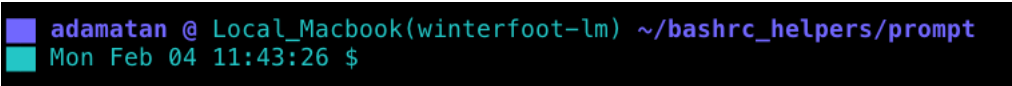


Long lines overlap in Bash PS1 customized prompt

I have configured a fancy two-line `PS1` bash prompt, with server name, time, user name and other useful variables.

```
PS1="\n\[\033[1;34m\]\342\226\210\342\226\210 \u @ $SERVER_NAME""$BBlue"" \w""$Color_Off\n\[\033[0;36m\]\342\226\210\342\226\210 \d \t $ \[\033[0;39m\]"
```



Full Github source here.

The problem is that long lines are often wrapped around themselves, meaning that the new line overlaps the previous one.

The problem is inconsistent: some lines are wrapped correctly, some are not.

Any idea how to fix this?

```
bash      prompt      ps1
```

3 Answers

Here is my fancy two-line `PS1` for bash prompt.

`PS1`

```
\[\e[1;32m\]\H\[\e[0m\] :: \[\e[1;30m\]\d\[\e[0m\] \@ :: \[\e[1;33m\]\w\n\[\e[1;34m\]\u $ \[\033[0m\]
```

So here is what I did. If this was my color:

```
\e[1;32m
```

I changed it to:

```
\[\e[1;32m\]
```

Essentially you need to enclose it in:

```
\[$mycolor\]
```

Checkout this answer.

edited Apr 13 '17 at 12:23

Community ♦

1

answered Feb 25 '13 at 14:12

varunyellina

194 2 3 11

- +1 Thanks. Oddly enough, it doesn't seem to work on mac. – Adam Matan Feb 25 '13 at 14:21
- You're a life saver! As for mac, Adam.. It's most likely due to the column count. Try setting the prompt to just a string of characters, like 'this is my prompt: ' and see if the problem persists. If so, your column setting on your terminal is either too wide or too narrow. – Alex Summers Oct 18 '17 at 5:16

Found this solution online, which suggested a closing brackets to color variables:

Change this:

```
Black='\e[0;30m'      # Black
Red='\e[0;31m'        # Red
Green='\e[0;32m'      # Green
```

to:

```
Black='\e[0;30m\]'    # Black
Red='\e[0;31m\]'      # Red
Green='\e[0;32m\]'    # Green
```

I've committed the change - hope it works as expected.

answered Feb 4 '13 at 11:47

Adam Matan

4,264 18 53 83

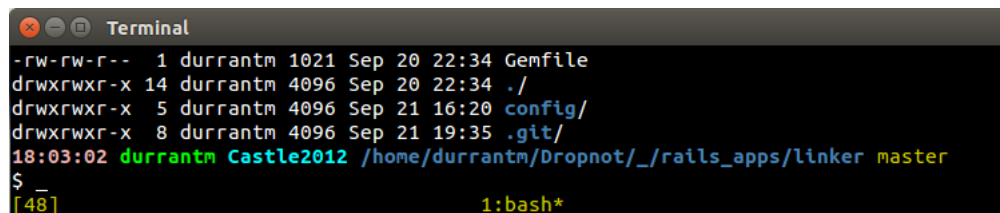
1 not sure what this is supposed to achieve. You need to wrap anything that doesn't take up space (ansi-escapes) in `\[\]`, so you can't just end all the ansi-escape colors with `\]` because what if there isn't a `\]`. When you are putting in non-printable chars, you should just wrap them in `\[\]` (e.g. `PS1='\[\e[32;1m\] green prompt \[\e[0m\] $' \)`. – dylmnc

Sep 26 '16 at 21:15

Indeed! Without the opening `\[`, this boshes the prompt for us linux users. Maybe it's different for OSX? How strange. `\[` and `\e` are interpreted as two different things for linux. The first indicates to bash that the sequence following is unprintable and not to count it as part of the displayed characters of the prompt. The `\e` is the escape bracket to start an ANSI color scheme. – Alex Summers Oct 18 '17 at 5:18

My preferred solution to the issue of long lines overlap in bash PS1 is to have a `\n` at the end of PS1 prompt line(s).

e.g.



```
Terminal
-rw-rw-r-- 1 durrantm 1021 Sep 20 22:34 Gemfile
drwxrwxr-x 14 durrantm 4096 Sep 20 22:34 ./
drwxrwxr-x  5 durrantm 4096 Sep 21 16:20 config/
drwxrwxr-x  8 durrantm 4096 Sep 21 19:35 .git/
18:03:02 durrantm Castle2012 /home/durrantm/Dropnot/_/rails_apps/linker master
$ _
[48]                               1: bash*
```

If you want all my other stuff - time, user, host, directory and git branch(!) you can use this (mac or linux):

<https://unix.stackexchange.com/a/124012/10043>

edited Apr 13 '17 at 12:37

Community ♦
1

answered Sep 23 '14 at 22:04

Michael Durrant
4,820 19 46 67

for git branch you should use `if [[$(git rev-parse --is-inside-git-dir 2> /dev/null)]]; then branch="$(git symbolic-ref --short -q HEAD 2> /dev/null)"; fi`. This is a standard when putting git into a script and evaluating the branch name from what I have read previously. – dylmnc Sep 26 '16 at 21:11