**How to Change Bash Prompt**

 Programming & Scripting

**Contents**

* + [1. Software Requirements and Conventions Used](https://linuxconfig.org/bash-prompt-basics/#h1-software-requirements-and-conventions-used)
  + [2. Bash prompt variables](https://linuxconfig.org/bash-prompt-basics/#h2-bash-prompt-variables)
  + [3. Bash PS1 prompt variable](https://linuxconfig.org/bash-prompt-basics/#h3-bash-ps1-prompt-variable)
  + [4. Bash PS2 prompt variable](https://linuxconfig.org/bash-prompt-basics/#h4-bash-ps2-prompt-variable)
  + [5. Bash prompt special characters](https://linuxconfig.org/bash-prompt-basics/#h5-bash-prompt-special-characters)
  + [6. Bash prompt customization](https://linuxconfig.org/bash-prompt-basics/#h6-bash-prompt-customization)
  + [7. Permanent bash variable definition](https://linuxconfig.org/bash-prompt-basics/#h7-permanent-bash-variable-definition)
  + [8. Temporary bash variable definition](https://linuxconfig.org/bash-prompt-basics/#h8-temporary-bash-variable-definition)
  + [9. Changing foreground and background bash prompt colors](https://linuxconfig.org/bash-prompt-basics/#h9-changing-foreground-and-background-bash-prompt-colors)
  + [10. Bash Prompt Examples](https://linuxconfig.org/bash-prompt-basics/#h10-bash-prompt-examples)
    - * [10.1.1. Display current Time](https://linuxconfig.org/bash-prompt-basics/#h10-1-1-display-current-time)
      * [10.1.2. Counting Files in the Current Directory](https://linuxconfig.org/bash-prompt-basics/#h10-1-2-counting-files-in-the-current-directory)

Default bash command line prompt on many Linux systems is quite minimal. As we will see in this article, it can be easily changed by modifying bash PS{n} variables, so to include information such as display time, load, number of users using the system, uptime and more.

**In this tutorial you will learn:**

* What are PS1 and PS2 shell variables
* How to create custom shell prompts
* What are the characters we can use to customize a shell prompt

[](https://linuxconfig.org/images/linux-bash-prompt.jpg)

Linux bash prompt

**Software Requirements and Conventions Used**

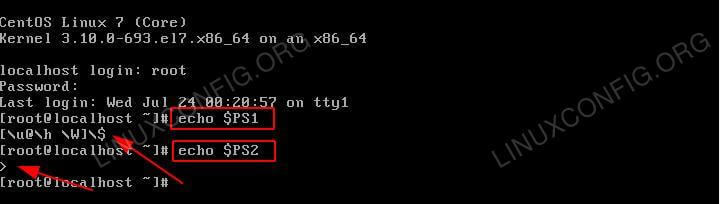
| *Software Requirements and Linux Command Line Conventions* | |
| --- | --- |
| **Category** | **Requirements, Conventions or Software Version Used** |
| **System** | Distribution-independent |
| **Software** | No special software is needed to follow this tutorial |
| **Other** | Minimal knowledge of the Bash shell |
| **Conventions** | **#** - requires given [linux commands](https://linuxconfig.org/linux-commands) to be executed with root privileges either directly as a root user or by use of sudo command **$** - requires given [linux commands](https://linuxconfig.org/linux-commands) to be executed as a regular non-privileged user |

**Bash prompt variables**

As anything else in the Linux system also bash prompt can be customized. We can accomplish the task by changing the values of bash PS1, PS2, PS3, PS4 variables. To keep the things simple, this article will be concerned just with the first two. Use echo command to see their values:

$ echo "Bash PS1 variable:" $PS1

$ echo "Bash PS2 variable:" $PS2

[](https://linuxconfig.org/images/Ps1_ps2_bash_prompt.jpg)

PS1 and PS2 bash prompt

***SUBSCRIBE TO NEWSLETTER*** *Subscribe to Linux Career*[*NEWSLETTER*](https://bit.ly/2X5D30q)*and receive latest Linux news, jobs, career advice and tutorials.*

**Bash PS1 prompt variable**

PS1 is a primary prompt variable. Currently it holds \u@\h:\w\$ special bash characters. This is the default structure of the bash prompt on many Linux systems and is displayed every time you log in using a terminal. Please see the following section "Bash prompt special characters" for explanation of \u, \h, \w and \$ symbols. Here is a classical bash prompt with default settings:

[ps1-bash-prompt](https://linuxconfig.org/images/Ps1_bash_prompt.gif)

PS1 bash prompt

**Bash PS2 prompt variable**

PS2 bash shell variable is a secondary prompt. This prompt is displayed if the shell waits for a user input, for example you forget to insert second quotation.

[ps2-bash-prompt](https://linuxconfig.org/images/Ps2_bash_prompt.gif)

PS2 bash prompt

**Bash prompt special characters**

Bash prompt can be customized by using special characters. Here is a quick overview of the most used characters and their meaning:

| *Bash prompt special characters* | | | |
| --- | --- | --- | --- |
| **Bash special character** | **Bash special character explanation** | **Bash special character** | **Bash special character explanation** |
| \\a | an ASCII bell character (07) | \\d | the date in "Weekday Month Date" format (e.g., "Tue May 26") |
| \\] | end a sequence of non-printing characters | \\e | an ASCII escape character (033) |
| \\h | the hostname up to the first `.' | \\H | the hostname |
| \\j | the number of jobs currently managed by the shell | \\l | the basename of the shell's terminal device name |
| \\n | newline | \\r | carriage return |
| \\s | the name of the shell, the basename of $0 (the portion following the final slash) | \\t | the current time in 24-hour HH:MM:SS format |
| \\T | the current time in 12-hour HH:MM:SS format | \\@ | the current time in 12-hour am/pm format |
| \\A | the current time in 24-hour HH:MM format | \\u | the username of the current user |
| \\v | the version of bash (e.g., 2.00) | \\V | the release of bash, version + patchelvel (e.g., 2.00.0) |
| \\w | the current working directory | \\W | the basename of the current working directory |
| \\! | the history number of this command | \\# | the command number of this command |
| \\$ | if the effective UID is 0, a #, otherwise a $ | \\nnn | the character corresponding to the octal number nnn |
| \\\\ | a backslash | \\[ | begin a sequence of non-printing characters, which could be used to embed a terminal control sequence into the prompt |
| \\D{format} | the format is passed to strftime(3) and the result is inserted into the prompt string; an empty format results in a locale-specific time representation. The braces are required | | |

**Bash prompt customization**

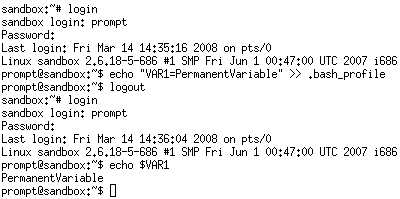
After user logins into the system, user environment variables are initialized from various files:

* /etc/profile or /etc/bashrc (system wide)
* ~/.bash\_profile , ~/.bash\_login , ~/.profile , ~/.bashrc or ~/.bash\_logout (user)

It is important to know that all users environment variable have a life time equal to the terminal session. When the terminal session is closed the user's variables including [bash shell variables](https://linuxconfig.org/bash-scripting-tutorial-for-beginners#h11-variables) defined during a terminal session are emptied and a again redefined when new terminal session is created either via logo in shell or interactive shell. Lets define two variables to prove this statement.

**Permanent bash variable definition**

First, we will define a permanent variable in one of the bash initialization files, ~/.bash\_profile, then we will define a temporary variable in the shell prompt. Let's define permanent user variable:

[](https://linuxconfig.org/images/Bash_permanent_variable.gif)

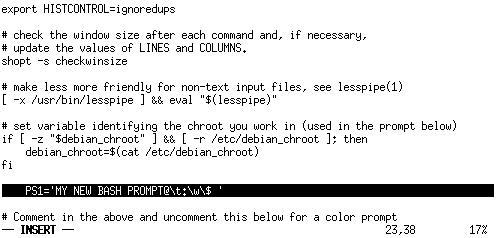
Bash permanent variable

What happened here, is that user "prompt" modified its own .bash\_profile initialization file located in his/her home directory by appending a VAR1 variable definition. When user "prompt" logged out and logged in again the $var1 variable is initialized and available for the new terminal session.

On the same principles we can define our bash prompt. The best place to do it is that bash initialization file .~/bashrc. Open up your ~/.bashrc file and add/edit the line defining a PS1 variable to something like:

PS1='MY NEW BASH PROMPT@\t:\w\$ '

*NOTE: Your ~/.barshrc file may differ from the example below !*

[](https://linuxconfig.org/images/Bash_new_prompt_define.gif)

New prompt

$ source .bashrc

or similarly:

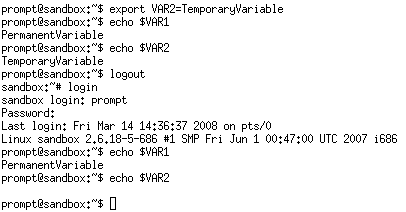
$ . .bashrc

[another-prompt-define](https://linuxconfig.org/images/Bash_another_prompt_define.gif)

Another prompt

**Temporary bash variable definition**

A temporary bash variable lasts only as long as the current terminal session. This is tome by an export command.

[](https://linuxconfig.org/images/Bash_temporary_variable.gif)

Bash temporary variable

As you can see the variable $VAR2 is not defined when user closes his/her terminal session. The permanent variable $VAR1 is always defined from the bash initialization file: ~/.bash\_profile. As we can use an export command to define new bash variables we can also use it to modify a bash prompt $PS1 variable. To change a current bash prompt to display only time we could do:

export PS1="\t: "

[bash-temporary-prompt-define](https://linuxconfig.org/images/Bash_temporary_prompt_define.gif)

Temporary prompt definition

**Changing foreground and background bash prompt colors**

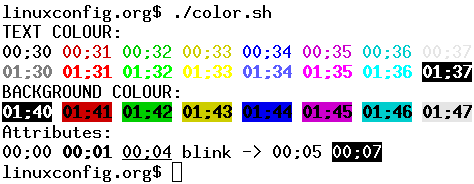
Syntax for changing colors in the bash is as follows:

\033[ - Indicates the beginning of color in the text

x;yzm - Indicates color code

\033[00m - Indicates the end of color in the text

Bash color codes:

[](https://linuxconfig.org/images/bash-color-codes.gif)

Bash color codes

export PS1="\033[01;31mBASH IN RED\033[00m: "

[Bash prompt - red color](https://linuxconfig.org/images/Bash_prompt_red_color.gif)

Bash prompt - red color

**Bash Prompt Examples**

To get you started with your new bash prompt here are couple examples:

**Display current Time**

export PS1="\u@\h \t:\$ "

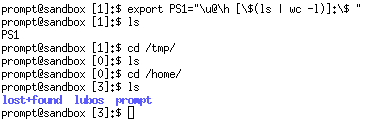
[bash-prompt-current-time](https://linuxconfig.org/images/Bash_prompt_current_time.gif)

Bash prompt with current time

**Counting Files in the Current Directory**

This bash prompt displays current number of files and directories in the current directory.

export PS1="\u@\h [\$(ls | wc -l)]:\$ "

[](https://linuxconfig.org/images/Bash_prompt_count_files_time.gif)

Bash prompt with files count