

Here we have added `/home/user/bin` to `PATH`.

Some of the well-known environment variables are `HOME`, `PWD`, `USER`, `UID`, `SHELL`, and so on.



When using single quotes, variables will not be expanded and will be displayed as is. This means:

`$ echo '$var' will print $var`

Whereas, `$ echo "$var"` will print the value of the `$var` variable if defined or nothing at all if it is not defined.

### There's more...

Let us see more tips associated with standard and environment variables.

### Finding the length of a string

Get the length of a variable value using the following command:

```
length=${#var}
```

For example:

```
$ var=12345678901234567890$  
echo ${#var}  
20
```

The `length` parameter will bear the number of characters in the string.

### Identifying the current shell

To identify the shell which is currently being used, we can use the `SHELL` variable, like so:

```
echo $SHELL
```

Or:

```
echo $0
```

For example:

```
$ echo $SHELL  
/bin/bash
```

```
$ echo $0  
/bin/bash
```

## Checking for super user

UID is an important environment variable that can be used to check whether the current script has been run as a root user or regular user. For example:

```
If [ $UID -ne 0 ]; then
    echo Non root user. Please run as root.
else
    echo Root user
fi
```

The UID value for the root user is 0.

## Modifying the Bash prompt string (username@hostname:~\$)

When we open a terminal or run a shell, we see a prompt string such as `user@hostname:/home/$`. Different GNU/Linux distributions have slightly different prompts and different colors. We can customize the prompt text using the `PS1` environment variable. The default prompt text for the shell is set using a line in the `~/ .bashrc` file.

- ▶ We can list the line used to set the `PS1` variable as follows:
 

```
$ cat ~/.bashrc | grep PS1
PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
```
- ▶ To set a custom prompt string, enter the following command:
 

```
slynux@localhost: ~$ PS1="PROMPT>"
PROMPT> Type commands here # Prompt string changed.
```
- ▶ We can use colored text using the special escape sequences such as `\e[1;31` (refer to the *Printing in the terminal* recipe of this chapter).

There are also certain special characters that expand to system parameters. For example, `\u` expands to username, `\h` expands to hostname, and `\w` expands to the current working directory.

## Function to prepend to environment variables

Environment variables are often used to store a list of paths of where to search for executables, libraries, and so on. Examples are `$PATH`, `$LD_LIBRARY_PATH`, which will typically look like this:

```
PATH=/usr/bin:/bin
LD_LIBRARY_PATH=/usr/lib:/lib
```