Variables in bash

Understand the concept of a variable and how to do every process to create your own.

What is a variable?

A variable as its name implies is to store varied content in a memory location / space, instead of using numbers to make your call, we use tags / names! This makes it easier to make the call, isn't it? Shell variables can receive any content! from a simple string / text, numbers, output status of a command and return output.

When should we use a variable?

At any time! variables are cool to work with, let's suppose as an example i needed to call a specific directory 4 times! Would it be feasible for me to always type this manual directory? Wouldn't it be flawed if in the future I decided to change the directory itself? And if I forget to change any path is for sure fail! For this I use a variable to help me!

```
# Declaration
DIRECTORY="/home/slackjeff/local_directory"

# If directory does not exist, create.
if [! -d "$DIRECTORY"]; then
   mkdir "$DIRECTORY"
fi
cd "$DIRECTORY"
echo "I'm in the directory '$DIRECTORY'"
```

Note that I made the same directory call 4 times! It was more feasible to use a variable for this, so if I need to, I just need to change the contents of the variable without having to touch anything else in the code. The same goes for a message for example where it is repetitive or even long! For this example I will use the dialog.

```
# HEREDOCUMENT
text=$(
cat << END
    Hi, welcome to my site, man!
    Here you will find everything but money.
END
)
# PRINT
dialog --msgbox "$text" 0 0
```

Variable Types

In shell there are no variables of type (char, int, float, double). Here is all merciless string! Don't worry … Just schedule it.

Declaring a variable

In shell we don't need to declare variables! but get used to the way you call: D The order to declare a variable is:. Note that the variable name cannot contain any special characters like: '@ - | & * # $ \ /' among others, they cannot also start with numbers and have spaces … If your variable name contains spaces replace with _ 'underline', yes! underline can. Do not use - 'dash' that will not roll, it is a special character. After declaring a GOOD variable name you must enter the '=' sign and the value / content. Let's go to some examples:

CORRECT STATEMENTS

```
size_calca_42="Lie or use 42"
name="Jefferson"
_my_street="Pedro dos Santos"
MyHomeNumber="1076552"
CPF_AND_RG = "CPF: xxxxxxxx-xxxxxx.xx RG: y.yyy.yyyy"
```

WRONG statements

```
echo="Jeffa" **Works more is booked.**
ls="Uoool" **Works more is booked**
*Nom&="Paula"
1name="Fernanda"
goat-street="Pedrino Nicolau"
language=favorite="shell"
MY YELLOW="this won't work"
```

Take care if you are beginner and making the wrong statement will not worka. The shell will complain, yes! he is a very angry guy with bad statements. Then it binds, without special characters, start numbers and spaces.

Sources

- Originally written by Slackjeff

howtos, misc, variables bash, shell, author slackjeff

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:misc:variables_in_bash

Last update: **2020/01/02 18:20 (UTC)**