LINUX FOUNDATION

TRAINING

Basic Bash Scripting

A Linux Foundation Training Publication
www.training.linuxfoundation.org

Overview

Bash offers many programming and interactive capabilities that can be used to write useful scripts to automate tasks, save time, and increase your productivity. One of these is the use of variables, which this article will mostly focus on.

Variables are common to all programming languages. A variable is a character string that can be declared and stored with a particular value. The variable that then be used within the script as a substitute for that value, and can also be changed elsewhere in the script.

In the following example, the variable ‘STR’ is declared with “Hello world!”. The variable can then be called:

```
STR=“Hello world!”
echo $STR
```

Key Ideas

Bash shebang: The line ‘#!/bin/bash’ at the start of the script that tells the kernel to use the bash interpreter for the script.**Software repository:** trusted central source of compiled, tested, integrated software packages provided by a distribution.

Variable: A character string to which a particular value is assigned and stored for use in the script. A variable name is typically declared using uppercase letters, and invoked wrapped in curly braces preceded by a $ sign. For example, ${VARIABLE_NAME}.

```
NAME=John
echo “Your name is ${NAME}.”
Returns: Your name is John.
```

Read -p: The ‘read’ command takes user input and assigns it to a variable. The -p (for prompt) option is a neat way of prompting the user for input and assigning it to a variable on the one line.

```
read -p “What is your name?” NAME
Returns: What is your name?
```

Whatever the user types will be assigned as the value to the variable ‘NAME’

Example Scenario

Write a bash script that prints ‘Hello world!’ from an assigned variable, then ask for the user’s name so the user is greeted by name.

Now Do It

1. Create a file that ends in .sh and open it
2. Include the bash shebang at the top of the file
3. Write a comment that explains what the script does
4. Write a two-line script that
    a. declares the string ‘Hello world!’ as a variable
    b. prints the ‘Hello world!’ string
    c. prompts the user for their name to be used as a variable
    d. greets the user by their name
5. Give the script execute permissions
6. Run the script

If you remember nothing else...

Your distribution provides you with a safe, trusted source of packages. Packages created by and for one distribution probably won’t work on another, even if they use the same package format.

Answer Key

1 - 4.

script.sh:

```
#!/bin/bash

# This script asks for the user’s name so that it can greet them.
# A spin on the ‘Hello world’ example.

STR=“Hello world!”
echo $STR

read -p “Please type your name and hit [ENTER] “ NAME
echo “Hello ${NAME}!”
```

5. chmod +x script.sh

6. ./script.sh

LINUX FOUNDATION

Get Certified!

Get more information at http://training.linuxfoundation.org/certification/lfcs