Reading: A Brief Introduction to Shell Variables

Learning Objectives

After completing this reading, you will be able to:

- Describe shell variables
- Create shell variables

What is a shell variable?

Shell variables offer a powerful way to store and later access or modify information such as numbers, character strings, and other data structures by name. Let's look at some basic examples to get the idea.

Consider the following example.

```
1. 1
2. 2
3. 3
```

```
1. $ firstname=Jeff
2. $ echo $firstname
3. Jeff
```

Copied!

The first line assigns the value `Jeff` to a new variable called `firstname`. The next line accesses and displays the value of the variable, using the `echo` command along with the special character `$` in front of the variable name to extract its value, which is the string `Jeff`.

Thus, we have created a new shell variable called `firstname` for which the value is `Jeff`.

This is the most basic way to create a shell variable and assign it to a value all in one step.

Reading user input into a shell variable at the command line

Here's another way to create a shell variable, using the `read` command.
After entering

```
1. 1
```

```
1. $ read lastname
```

Copied!

on the command line, the shell waits for you to enter some text:

```
1. 1
2. 2
3. 3

1. $ read lastname
2. Grossman
3. $
```

Copied!

Now we can see that the value `Grossman` has just been stored in the variable `lastname` by the `read` command:

```
1. 1
2. 2
3. 3
4. 4

1. $ read lastname
2. Grossman
3. $ echo $lastname
4. Grossman
```

Copied!

By the way, notice that you can echo the values of multiple variables at once.

```
1. 1
2. 2

1. $ echo $firstname $lastname
2. Jeff Grossman
```

Copied!

As you will soon see, the `read` command is particularly useful in shell scripting. You can use it within a shell script to prompt users to input information, which is then stored in a shell variable and available for use by the shell script while it is running. You will also learn about **command line arguments**, which are values that can be passed to a script and automatically assigned to shell variables.

Summary

In this reading, you learned that:

- Shell variables store values and allow users to later access them by name
- You can create shell variables by declaring a shell variable and value or by using the `read` command

Authors

Jeff Grossman

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-05-04	1.4	Benny Li	Minor formatting changes
2023-04-27	1.3	D Teel-Friedman	Copy edit pass
2023-04-14	1.2	Nick Yi	ID Review
2023-03-22	1.1	Jeff Grossman	Added brief intro and minor edits
2022-12-23	1.0	Jeff Grossman	Created initial version of the reading