

BY BOBBY ILIEV

Introduction to Bash Scripting

FOR DEVELOPERS



Table of Contents

About the book	7
About the author	8
Sponsors	9
Ebook PDF Generation Tool	11
Book Cover	12
License	13
Introduction to Bash scripting	14
Bash Structure	15
Bash Hello World	16
Bash Variables	18
Bash User Input	21
Bash Comments	23
Bash Arguments	24
Bash Arrays	27
Bash Conditional Expressions	29
File expressions	30
String expressions	32

Arithmetic operators	34
Bash Conditionals	36
If statement	37
If Else statement	38
Switch case statements	40
Conclusion	42
Bash Loops	43
For loops	44
While loops	46
Until Loops	48
Continue and Break	49
Bash Functions	50
Debugging, testing and shortcuts	52
Creating custom bash commands	55
Example	56
Making the change persistent	58
Listing all of the available aliases	59
Conclusion	60
Write your first Bash script	61
Planning the script	62
Writing the script	63
Adding comments	64
Adding your first variable	65
Adding your first function	66

Adding more functions challenge	68
The sample script	69
Conclusion	71
Creating an interactive menu in Bash	72
Planning the functionality	73
Adding some colors	75
Adding the menu	76
Testing the script	78
Conclusion	82
Executing BASH scripts on Multiple Remote Servers	83
Prerequisites	84
The BASH Script	87
Running the Script on all Servers	89
Conclusion	91
Work with JSON in BASH using jq	92
Planning the script	93
Installing jq	94
Parsing JSON with jq	96
Getting the first element with jq	98
Getting a value only for specific key	99
Using jq in a BASH script	100
Conclusion	103
Working with Cloudflare API with Bash	104
Prerequisites	105
Challenge - Script requirements	106

Example script	107
Conclusion	109

BASH Script parser to Summarize Your NGINX and Apache Access Logs 110

Script requirements	111
Example script	112
Running the script	113
Understanding the output	114
Conclusion	116

Sending emails with Bash and SSMTP	117
Prerequisites	118
Installing SSMTP	119
Configuring SSMTP	120
Sending emails with SSMTP	121
Sending A File with SSMTP (optional)	122
Conclusion	123

Password Generator Bash Script	124
:warning: Security	125
Script summary	126
Prerequisites	127
Generate a random password	128
The script	130
The full script:	131
Conclusion	132
Contributed by	133

Wrap Up	134
----------------------	------------

About the book

- **This version was published on Feb 01 2021**

This is an open-source introduction to Bash scripting guide that will help you learn the basics of Bash scripting and start writing awesome Bash scripts that will help you automate your daily SysOps, DevOps, and Dev tasks. No matter if you are a DevOps/SysOps engineer, developer, or just a Linux enthusiast, you can use Bash scripts to combine different Linux commands and automate tedious and repetitive daily tasks so that you can focus on more productive and fun things.

The guide is suitable for anyone working as a developer, system administrator, or a DevOps engineer and wants to learn the basics of Bash scripting.

The first 13 chapters would be purely focused on getting some solid Bash scripting foundations, then the rest of the chapters would give you some real-life examples and scripts.

About the author

My name is Bobby Iliev, and I have been working as a Linux DevOps Engineer since 2014. I am an avid Linux lover and supporter of the open-source movement philosophy. I am always doing that which I cannot do in order that I may learn how to do it, and I believe in sharing knowledge.

I think it's essential always to keep professional and surround yourself with good people, work hard, and be nice to everyone. You have to perform at a consistently higher level than others. That's the mark of a true professional.

For more information, please visit my blog at <https://bobbyiliev.com>, follow me on Twitter [@bobbyiliev_](#) and [YouTube](#).

Sponsors

This book is made possible thanks to these fantastic companies!

DigitalOcean

DigitalOcean is a cloud services platform delivering the simplicity developers love and businesses trust to run production applications at scale.

It provides highly available, secure, and scalable compute, storage, and networking solutions that help developers build great software faster.

Founded in 2012 with offices in New York and Cambridge, MA, DigitalOcean offers transparent and affordable pricing, an elegant user interface, and one of the largest libraries of open source resources available.

For more information, please visit <https://www.digitalocean.com> or follow [@digitalocean](https://twitter.com/digitalocean) on Twitter.

If you are new to DigitalOcean, you can get a free \$100 credit and spin up your own servers via this referral link here:

[Free \\$100 Credit For DigitalOcean](#)

DevDojo

The DevDojo is a resource to learn all things web development and web design. Learn on your lunch break or wake up and enjoy a cup of coffee with us to learn something new.

Join this developer community, and we can all learn together, build together, and grow together.

[Join DevDojo](#)

For more information, please visit <https://www.devdojo.com> or follow [@thedeveloper](#) on Twitter.

Ebook PDF Generation Tool

This ebook was generated by [Ibis](#) developed by [Mohamed Said](#).

Ibis is a PHP tool that helps you write eBooks in markdown.

Book Cover

The cover for this ebook was created with [Canva.com](https://www.canva.com).

If you ever need to create a graphic, poster, invitation, logo, presentation – or anything that looks good — give Canva a go.

License

MIT License

Copyright (c) 2020 Bobby Iliev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Introduction to Bash scripting

Welcome to this Bash basics training guide! In this **bash crash course**, you will learn the **Bash basics** so you could start writing your own Bash scripts and automate your daily tasks.

Bash is a Unix shell and command language. It is widely available on various operating systems, and it is also the default command interpreter on most Linux systems.

Bash stands for Bourne-Again SHell. As with other shells, you can use Bash interactively directly in your terminal, and also, you can use Bash like any other programming language to write scripts. This book will help you learn the basics of Bash scripting including Bash Variables, User Input, Comments, Arguments, Arrays, Conditional Expressions, Conditionals, Loops, Functions, Debugging, and testing.

In order to write Bash scripts, you just need a UNIX terminal and a text editor like Sublime Text, VS Code, or a terminal-based editor like vim or nano.

Bash Structure

Let's start by creating a new file with a `.sh` extension. As an example, we could create a file called `devdojo.sh`.

To create that file, you can use the `touch` command:

```
touch devdojo.sh
```

Or you can use your text editor instead:

```
nano devdojo.sh
```

In order to execute/run a bash script file with the bash shell interpreter, the first line of a script file must indicate the absolute path to the bash executable:

```
#!/bin/bash
```

This is also called a Shebang.

All that the shebang does is to instruct the operating system to run the script with the `/bin/bash` executable.

Bash Hello World

Once we have our `devdojo.sh` file created and we've specified the bash shebang on the very first line, we are ready to create our first **Hello World** bash script.

To do that, open the `devdojo.sh` file again and add the following after the `#!/bin/bash` line:

```
#!/bin/bash  
  
echo "Hello World!"
```

Save the file and exit.

After that make the script executable by running:

```
chmod +x devdojo.sh
```

After that execute the file:

```
./devdojo.sh
```

You will see a "Hello World" message on the screen.

Another way to run the script would be:

```
bash devdojo.sh
```


As bash can be used interactively, you could run the following command directly in your terminal and you would get the same result:

```
echo "Hello DevDojo!"
```

Putting a script together is useful once you have to combine multiple commands together.