
shellshock Documentation

Release 0.0.1

Matt Dodge

May 14, 2019

Contents

1	Getting Started	3
1.1	Install	3
1.2	Hello World	3
2	Python feature compatability	5
3	Testing	7
4	Converters	9
5	Converter Reference	11
6	CLI Reference	13
7	Installation	15
8	Usage	17

Contents:

CHAPTER 1

Getting Started

A walkthrough highlighting the main features of Shellshock

1.1 Install

If you haven't yet, install Shellshock

```
pip install shellshock
```

1.2 Hello World

To start, we'll create the Hello World of Python/Shell scripts. Let's create a Python file called **script.py** that has the following contents:

```
print('Hello world!')
```

Now, you can convert this to a shell script using the Shellshock CLI. Without any arguments, Shellshock will output the resulting shell script to stdout.

```
$ shellshock script.py  
echo 'Hello world!'
```

If you want, you can even pipe the output to your favorite shell to run it.

```
$ shellshock script.py | bash  
Hello world!
```


CHAPTER 2

Python feature compatability

Python Feature	Shellshock Support	Example
Strings	YES	<code>"here we go"</code>
Integers/Floats	YES	<code>my_val = 3.14</code>
Function definitions	YES	<pre>def doit(var1, var2='default'): print(var1) print(var2) doit('once', 'twice')</pre>

CHAPTER 3

Testing

Finally, you can unit test your shell scripts!

CHAPTER 4

Converters

Converter Reference

Shellshock comes with many useful converter functions built-in. Here is the reference of these functions.

Examples are provided for most functions. For examples assume *script.py* is the input Shellshock script and *script.sh* is the output after Shellshock transpilation.

`ss.shell(raw_shell_cmd)`

Run a raw shell command. The argument will be output in its entirety.

Use this method when you want to write regular shell commands in your script.

Example: Basic Usage

Listing 1: script.py

```
import shellshock as ss
ss.shell("chown 755 myfile")
```

Listing 2: script.sh

```
#!/bin/sh
chown 755 myfile
```

Example: Convert Existing Script

Wrap an entire existing shell script (or a part of a script) in *ss.shell* to partially migrate an existing script to Shellshock

Listing 3: script.py

```
import shellshock as ss
ss.shell("""
chown 755 myfile
# other existing shell script
```

(continues on next page)

(continued from previous page)

```
echo "output"  
""")
```

Listing 4: script.sh

```
#!/bin/sh  
chown 755 myfile  
# other existing shell script  
echo "output"
```


CHAPTER 6

CLI Reference

CLI command reference

CHAPTER 7

Installation

```
pip install shellshock
```


CHAPTER 8

Usage

```
shellshock script.py -o script.sh
```


S

`shell()` (*in module ss*), [11](#)