

(/)



Curriculum

SE Foundations ^

Average: 108.76% v

0x04. Loops, conditions and parsing

DevOps

Shell

Bash

Scripting

By: Sylvain Kalache

Weight: 1

 Project over - took place from Jun 29, 2023 6:00 AM to Jul 1, 2023 6:00 AM☒ An auto review will be launched at the deadline

In a nutshell...

- **Auto QA review:** 47.7/83 mandatory & 0.0/30 optional
- **Altogether: 57.47%**
 - Mandatory: 57.47%
 - Optional: 0.0%
 - Calculation: $57.47\% + (57.47\% * 0.0\%) == 57.47\%$

About Bash projects

Unless stated, all your projects will be auto-corrected with Ubuntu 20.04 LTS.

Background Context

(<https://youtu.be/BC2neyc5Gcl>)

Resources

Read or watch:

- Loops sample (/rltoken/wT98UJfv_E2tk4yP9PcLLw)
- Variable assignment and arithmetic (</rltoken/olvOKX699pq50rkHRE5cSA>)
- Comparison operators (/rltoken/HxohzllkOWh0t4dy_HptlQ)
- File test operators (</rltoken/g8of2ABPEJfCNtPrDQaqVw>)



Help

- Make your scripts portable (/rltoken/O0Ay21p7tDhfLMsYbtAKug)

(/)

man or help:

- env
- cut
- for
- while
- until
- if

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/UnkzDNdH09TFJ0-Y56azyg), **without the help of Google**:

General

- How to create SSH keys
- What is the advantage of using `#!/usr/bin/env bash` over `#!/bin/bash`
- How to use `while`, `until` and `for` loops
- How to use `if`, `else`, `elif` and `case` condition statements
- How to use the `cut` command
- What are files and other comparison operators, and how to use them

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted on Ubuntu 20.04 LTS
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project, is mandatory
- All your Bash script files must be executable
- You are not allowed to use `awk`
- Your Bash script must pass `Shellcheck` (version `0.7.0`) without any error
- The first line of all your Bash scripts should be exactly `#!/usr/bin/env bash`
- The second line of all your Bash scripts should be a comment explaining what is the script doing

Copyright - Plagiarism

- You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
- You will not be able to meet the objectives of this or any following project by copying and pasting someone else's work.
- You are not allowed to publish any content of this project.
- Any form of plagiarism is strictly forbidden and will result in removal from the program.



More Info

Shellcheck

Shellcheck (/rltoken/joK6l_yEZ9N7T0GQ1RDjLA) is a tool that will help you write proper Bash scripts. It will make recommendations on your syntax and semantics and provide advice on edge cases that you might not have thought about. Shellcheck is your friend! **All your Bash scripts must pass Shellcheck without any error or you will not get any points on the task.**

Shellcheck is available on the school's computers. If you want to use it on your own computer, here is how to install it (/rltoken/jbz0_-i3TV3WpKgxyhrtpA).

Examples:

Not passing Shellcheck :

```
[sylvain@ubuntu$ cat bad_script
#!/usr/bin/env bash
var='some text'
unused_variable='some unused variable'

echo $var
[sylvain@ubuntu$ shellcheck bad_script

In bad_script line 3:
unused_variable='some unused variable'
^-- SC2034: unused_variable appears unused. Verify it or export it.


In bad_script line 5:
echo $var
    ^-- SC2086: Double quote to prevent globbing and word splitting.

sylvain@ubuntu$
```

Passing Shellcheck :

```
[sylvain@ubuntu$ cat good_script
#!/usr/bin/env bash
var='some text'
unused_variable='some unused variable'

echo "$var"
echo "$unused_variable"
[sylvain@ubuntu$ shellcheck good_script
sylvain@ubuntu$
```

For every feedback, Shellcheck will provide a code that you can use to get more information about the issue, for example for code SC2034 , you can browse <https://github.com/koalaman/shellcheck/wiki/SC2034> (/rltoken/dxp49_rfO4_9Yjtcg59exg). 

Tasks

0. Create a SSH RSA key pair

mandatory

Score: 100.0% (Checks completed: 100.0%)

Read for this task:

- Linux and Mac OS users (/rltoken/Cy1pIV2eR3VphjPqliXB8A)
- Windows users (/rltoken/074M_gTsD34x3Q6MX55PDw)

man: ssh-keygen

You will soon have to manage your own **servers** concept page hosted on remote data centers (/rltoken/nDPzEm5SYxcdGxP_OpVYXQ). We need to set them up with your RSA public key so that you can access them via SSH.

Create a RSA key pair.

Requirements:

- Share your **public key** in your answer file 0-RSA_public_key.pub
- Fill the SSH public key field of your intranet profile (/rltoken/qsEQ3ZWrgs-zoueDpXpPA) with the public key you just generated
- **Keep the private key to yourself in a secure location**, you will use it later to connect to your servers using SSH. Some storing ideas are Dropbox, Google Drive, password manager, USB key. Failing to do so will prevent you to access your servers, which will prevent you from doing your projects
- If you decide to add a passphrase to your key, make sure to save this passphrase in a secure location, you will not be able to use your keys without the passphrase

SSH and RSA keys will be covered in depth in a later project.

Repo:

- GitHub repository: alx-system_engineering-devops
- Directory: 0x04-loops_conditions_and_parsing
- File: 0-RSA_public_key.pub

☒ Done!

Help

Check your code

> Get a sandbox

QA Review

1. For Best School loop

mandatory



Score: 100.0% (Checks completed: 100.0%)

Write a Bash script that displays Best School 10 times.

Requirement:

(1)

- You must use the `for` loop (`while` and `until` are forbidden)

```
sylvain@ubuntu$ head -n 2 1-for_best_school
#!/usr/bin/env bash
# This script is displaying "Best School" 10 times
sylvain@ubuntu$ ./1-for_best_school
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
sylvain@ubuntu$
```

Note that:

- The first line of my Bash script starts with `#!/usr/bin/env bash`
- The second line of my Bash scripts is a comment explaining what it is doing


Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `1-for_best_school`

☒ Done!

Help

Check your code

 Get a sandbox

QA Review

2. While Best School loop

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Write a Bash script that displays `Best School` 10 times.

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)



```
sylvain@ubuntu$ ./2-while_best_school
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
sylvain@ubuntu$
```


Repo:

- GitHub repository: alx-system_engineering-devops
- Directory: 0x04-loops_conditions_and_parsing
- File: 2-while_best_school

☒ Done!

Help

Check your code

 Get a sandbox

QA Review

3. Until Best School loop

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a Bash script that displays `Best School` 10 times.

Requirements:

- You must use the `until` loop (`for` and `while` are forbidden)

```
sylvain@ubuntu$ ./3-until_best_school
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
sylvain@ubuntu$
```

**Repo:**

- GitHub repository: alx-system_engineering-devops

- Directory: 0x04-loops_conditions_and_parsing
- (/). File: 3-until_best_school

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

4. If 9, say Hi!

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a Bash script that displays `Best School` 10 times, but for the 9th iteration, displays `Best School` *and then* `Hi` on a new line.

Requirements:

- You must use the `while` loop (`for` and `until` are forbidden)
- You must use the `if` statement

```
sylvain@ubuntu$ ./4-if_9_say_hi
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Best School
Hi
Best School
sylvain@ubuntu$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x04-loops_conditions_and_parsing`
- File: `4-if_9_say_hi`

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

5. 4 bad luck, 8 is your chance

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a Bash script that loops from 1 to 10 and:

(/)

- displays bad luck for the 4th loop iteration
- displays good luck for the 8th loop iteration
- displays Best School for the other iterations

Requirements:

- You must use the while loop (for and until are forbidden)
- You must use the if, elif and else statements

```
sylvain@ubuntu$ ./5-4_bad_luck_8_is_your_chance
Best School
Best School
Best School
bad luck
Best School
Best School
Best School
good luck
Best School
Best School
sylvain@ubuntu$
```

For the most curious:

- 8 in the Chinese culture (/rltoken/3DKWikCCFAmt7JJt7Bv12w)
- 4 in the Chinese culture (/rltoken/WwpjD57ABmwWSfdUVcBhNg)


Repo:

- GitHub repository: alx-system_engineering-devops
- Directory: 0x04-loops_conditions_and_parsing
- File: 5-4_bad_luck_8_is_your_chance

☒ Done!

Help

Check your code

 Get a sandbox

QA Review

6. Superstitious numbers

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a Bash script that displays numbers from 1 to 20 and:

- displays 4 *and then* bad luck from China for the 4th loop iteration
- displays 9 *and then* bad luck from Japan for the 9th loop iteration
- displays 17 *and then* bad luck from Italy for the 17th loop iteration

Requirements:

- You must use the while loop (for and until are forbidden)



- You must use the case statement

(//)

```
sylvain@ubuntu$ ./6 superstitious_numbers
1
2
3
4
bad luck from China
5
6
7
8
9
bad luck from Japan
10
11
12
13
14
15
16
17
bad luck from Italy
18
19
20
sylvain@ubuntu$
```


Repo:

- GitHub repository: alx-system_engineering-devops
- Directory: 0x04-loops_conditions_and_parsing
- File: 6-superstitious_numbers

☒ Done!

Help

Check your code

 Get a sandbox

QA Review

7. Clock

mandatory

Score: 0.0% (Checks completed: 0.0%)

Write a Bash script that displays the time for 12 hours and 59 minutes:

- display hours from 0 to 12
- display minutes from 1 to 59

Requirements:

- You must use the while loop (for and until are forbidden)



Note that in this example, we only display the first 70 lines using the `head` command.

(/)



```
sylvain@ubuntu$ ./7-clock | head -n 70
```

```
Hour: 0
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
```



```
51
52
53
54
55
56
57
58
59
Hour: 1
1
2
3
4
5
6
7
8
9
sylvain@ubuntu$
```

Repo:

- GitHub repository: alx-system_engineering-devops
- Directory: 0x04-loops_conditions_and_parsing
- File: 7-clock

☐ Done?

Help

Check your code

Ask for a new correction

>_ Get a sandbox

QA Review

8. For ls

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a Bash script that displays:

- The content of the current directory
- In a list format
- Where only the part of the name after the first dash is displayed (refer to the example)

Requirements:

- You must use the `for` loop (`while` and `until` are forbidden)
- Do not display hidden files



```
sylvain@ubuntu$ ls
100-read_and_cut          1-for_best_school        6-superstitious_numbers
101-tell_the_story_of_passwd 2-while_best_school      7-clock
102-lets_parse_apache_logs  3-until_best_school      8-for_ls
103-dig_the-data          4-if_9_say_hi            9-to_file_or_not_to_file
e
10-fizzbuzz               5-4_bad_luck_8_is_your_chance
sylvain@ubuntu$ ./8-for_ls
read_and_cut
tell_the_story_of_passwd
lets_parse_apache_logs
dig_the-data
fizzbuzz
for_best_school
while_best_school
until_best_school
if_9_say_hi
4_bad_luck_8_is_your_chance
superstitious_numbers
clock
for_ls
to_file_or_not_to_file
sylvain@ubuntu$
```

Repo:

- GitHub repository: alx-system_engineering-devops
- Directory: 0x04-loops_conditions_and_parsing
- File: 8-for_ls

☒ Done!

Help

Check your code

> Get a sandbox

QA Review

9. To file, or not to file

mandatory

Score: 0.0% (Checks completed: 0.0%)

Write a Bash script that gives you information about the `school` file.

Requirements:

- You must use `if` and, `else` (`case` is forbidden)
- Your Bash script should check if the file exists and print:
 - if the file exists: `school file exists`
 - if the file does not exist: `school file does not exist`
- If the file exists, print:
 - if the file is empty: `school file is empty`
 - if the file is not empty: `school file is not empty`



- (/)
- if the file is a regular file: school is a regular file
 - if the file is not a regular file: (nothing)

```
sylvain@ubuntu$ file school
school: cannot open `school' (No such file or directory)
sylvain@ubuntu$ ./9-to_file_or_not_to_file
school file does not exist
sylvain@ubuntu$ touch school
sylvain@ubuntu$ ./9-to_file_or_not_to_file
school file exists
school file is empty
school is a regular file
sylvain@ubuntu$ echo 'betty' > school
sylvain@ubuntu$ ./9-to_file_or_not_to_file
school file exists
school file is not empty
school is a regular file
sylvain@ubuntu$ rm school
sylvain@ubuntu$ mkdir school
sylvain@ubuntu$ ./9-to_file_or_not_to_file
school file exists
school file is not empty
sylvain@ubuntu$
```

Repo:

- GitHub repository: alx-system_engineering-devops
- Directory: 0x04-loops_conditions_and_parsing
- File: 9-to_file_or_not_to_file

☐ Done?

Help

Check your code

Ask for a new correction

>_ Get a sandbox

QA Review

10. FizzBuzz

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a Bash script that displays numbers from 1 to 100.

Requirements:

- Displays `FizzBuzz` when the number is a multiple of 3 and 5
- Displays `Fizz` when the number is multiple of 3
- Displays `Buzz` when the number is a multiple of 5
- Otherwise, displays the number
- In a list format



```
sylvain@ubuntu$ ./10-fizzbuzz | head -20
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
sylvain@ubuntu$
```


Repo:

- GitHub repository: alx-system_engineering-devops
- Directory: 0x04-loops_conditions_and_parsing
- File: 10-fizzbuzz

☒ Done!

Help

Check your code

 Get a sandbox

QA Review

Done with the mandatory tasks? Unlock 4 advanced tasks now!

