



# CodeInActionLabs

## Shell 00

*Summary: This document is the subject for the Shell 00 module of the CodeInActionLabs @ aba2020.*

*Version: 1.0*

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>3</b>
<b>III</b>	<b>Exercise 00 : Z</b>	<b>5</b>
<b>IV</b>	<b>Exercise 01 : testShell00</b>	<b>6</b>
<b>V</b>	<b>Exercise 02 : SSH me!</b>	<b>7</b>
<b>VI</b>	<b>Exercise 03 : midLS</b>	<b>8</b>
<b>VII</b>	<b>Exercise 04 : GiT commit</b>	<b>9</b>
<b>VIII</b>	<b>Exercise 05 : gitignore</b>	<b>10</b>
<b>IX</b>	<b>Exercise 06 : clean</b>	<b>11</b>
<b>X</b>	<b>Submission and peer-evaluation</b>	<b>12</b>

# Chapter I

## Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change up before submission.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for all your exercises.
- Your exercises will be checked and graded by your fellow classmates.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. So if you want to avoid bad surprises, be as thorough as possible.
- Moulinette is not very open-minded.
- These exercises are carefully laid out by order of difficulty - from the easiest to the hardest. We suggest you to solve them in the given order.
- Using a forbidden function is considered cheating. Cheaters get -42, and this grade is non-negotiable.
- If your program contains a syntax error, you'll get 0.
- You cannot leave any additional file in your directory other than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called Google / man / the Web / ...
- Examine the examples thoroughly. Your code must exactly reproduce the examples, for the specified test cases. Beware that the examples could very well call for details that are not explicitly mentioned in the problem statement.
- By Odin, by Thor ! Use your brain !!!

# Chapter II

## Foreword

Here are the lyrics for City Hunter's theme song "Moonlight Shadow":

The last time ever she saw him  
Carried away by a moonlight shadow  
He passed on worried and warning  
Carried away by a moonlight shadow.  
Lost in a riddle that Saturday night  
Far away on the other side.  
He was caught in the middle of a desperate fight  
And she couldn't find how to push through

The trees that whisper in the evening  
Carried away by a moonlight shadow  
Sing a song of sorrow and grieving  
Carried away by a moonlight shadow  
All she saw was a silhouette of a gun  
Far away on the other side.  
He was shot six times by a man on the run  
And she couldn't find how to push through

[Chorus]  
I stay, I pray  
See you in Heaven far away...  
I stay, I pray  
See you in Heaven one day.

Four A.M. in the morning  
Carried away by a moonlight shadow  
I watched your vision forming  
Carried away by a moonlight shadow  
A star was glowing in the silvery night  
Far away on the other side  
Will you come to talk to me this night  
But she couldn't find how to push through


[Chorus]

Far away on the other side.  
Caught in the middle of a hundred and five  
The night was heavy and the air was alive  
But she couldn't find how to push through  
Carried away by a moonlight shadow  
Carried away by a moonlight shadow  
Far away on the other side.

Unfortunately, this topic has nothing to do with City Hunter.

# Chapter III

## Exercise 00 : Z


	Exercise 00
Only the best know how to display Z	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <b>z</b>	
Allowed functions : <b>None</b>	

- Create a file called **z** that returns "Z", followed by a new line, whenever the command **cat** is used on it.

```
?>cat z
Z
?>
```

# Chapter IV

## Exercise 01 : testShell00

	Exercise 01
What are attributes anyway ?	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <b>testShell00.tar</b>	
Allowed functions : None	

- Create a file called **testShell00** in your submission directory.
- Figure out a way for the output to look like this (except for the “total 1” line):

```
%> ls -l
total 1
-r--r-xr-x 1 XX XX 40 Jun 1 23:42 testShell00
%>
```

- Once you’ve achieved the previous steps, execute the following command to create the file to be submitted: **tar -cf testShell00.tar testShell00**.




Don’t worry about what you’ve got instead of “XX”.



A year will be accepted instead of the time, on the timestamp of the file.

# Chapter V

## Exercise 02 : SSH me!

	Exercise 02
SSH Key	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <code>id_rsa_pub</code>	
Allowed functions : None	

- Create your own SSH key. Once it is done:
  - Add your public key to your repository, in a file name `id_rsa_pub`
  - Update your ssh key on the intranet. This will allow you to push the repository to our git server.



The file's name was not chosen randomly.




Make sure you understand the difference between the public key and the private key.



# Chapter VI

## Exercise 03 : midLS

	Exercise 03
midLS	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <b>midLS</b>	
Allowed functions : <b>None</b>	

- In a `midLS` file, place the command line that will list all files and directories in your current directory (except for hidden files or any file that starts by a dot - yes, that includes double-dots), separated by a comma, by order of modification date. Make sure the directory's names are followed by a slash character.




What has not been asked for should not be done!



Read The Fucking Manual! You have to use only the `ls` command

# Chapter VII

## Exercise 04 : GiT commit

	Exercise 04
GiT commit?	
Turn-in directory : <i>ex04/</i>	
Files to turn in : <i>git_commit.sh</i>	
Allowed functions : None	

- Create a shell script that displays the ids of the last 5 commits of your git repository.

```
%> bash git_commit.sh | cat -e
baa23b54f0adb7bf42623d6d0a6ed4587e11412a$
2f52d74b1387fa80eea844969e8dc5483b531ac1$
905f53d98656771334f53f59bb984fc29774701f$
5ddc8474f4f15b3fcb72d08fcb333e19c3a27078$
e94d0b448c03ec633f16d84d63beaef9ae7e7be8$
%>
```

To test your script, we will use our own environment.




Create a folder with a new git repo. Then create 5 test commit



RTFM!

# Chapter VIII

## Exercise 05 : gitignore

	Exercise 05
GiT	
Turn-in directory : <i>ex05/</i>	
Files to turn in : <b>git_ignore.sh</b>	
Allowed functions : None	

- In this exercise, you will write a short shell script that lists all the existing files ignored by your GiT repository. Example:

```
%> bash git_ignore.sh | cat -e
.DS_Store$
mywork.c~$
%>
```


To test your script, we will use our own environment.



RTFM!

# Chapter IX

## Exercise 06 : clean

	Exercise 06
Turn-in directory : <i>ex06/</i>	
Files to turn in : <b>clean</b>	
Allowed functions : <b>None</b>	

- In a file called **clean** place the command line that will search for all files - in the current directory as well as in its sub-directories - with a name ending by ~, or a name that start and end by #
- The command line will show and erase all files found.
- Only one command is allowed: no ';' or '&&' or other shenanigans.



man find

# Chapter X

## Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your files to ensure they are correct.



You need to return only the files requested by the subject of this project.