Quest00

Subject 1 Solution Additional Resources (7)

Quest00

Remember to git add && git commit && git push each exercise!

We will execute your function with our test(s), please DO NOT PROVIDE ANY TEST(S) in your file

For each exercise, you will have to create a folder and in this folder, you will have additional files that contain your work. Folder names are provided at the beginning of each exercise under `submit directory` and specific file names for each exercise are also provided at the beginning of each exercise under `submit file(s)`.

Introduction

Welcome to the `Shell environment` quest.

Control Center

roup formation

In Progress

Go To DoCode

Access:

READ WRITE

Go To Gitea

mirsalik_i

Submit

Cancel Your Solution

We will dive into the environment of the famous command box. It's the Command center of the computer, very powerful but might be a little confusing at first sight.

In this quest, you will learn how to create directories and files with the right permissions, to read the listing directory command and to git push all your exercises.
You will also use the archive command `tar` and you will finish by writing your first script.

If we compare learning how to stand up before being able to walk then run in order to enjoy playing a sport with this quest you will learn to stand up! :-)

Quest00	My First File
Submit directory	ex00
Submit file	my_first_file

Description

For this very first assignment, you will be asked to create a file called `my_first_file` in your directory.

Which directory?
It's specified above this sentence (Submit directory: ex00 ;-))

Each assignment follow the same structure, you have to create a directory, go to the directory (cd) and work on your assignment.

Proceed by creating a file with the command: `touch` as you've seen in the video.

To verify if everything is in order: `ls`

You should see the file `my_first_file`.

Tip
To test if your exercise(s) is/are correct(s), you can execute the command `gandalf` in your terminal.

---

ubmitted

est review

inished

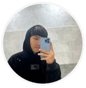
Looking for a group

No body is looking for a partner at the moment

Also working on the project

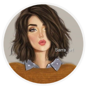
alvarado_j

regine-a_t


g-ulomjo_m

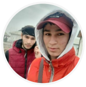
xudayqul_a

dehqonov_l


abdulla_su

xolxojay_f

bababeko_j


xolmamat_a

Type
Project

Group Size
1 Participant

Review system
Test Review (Gandalf)

Quest00	My First File With Content
Submit directory	ex01
Submit file	my_first_file_with_content

Description

Create a file called my_first_file_with_content in your directory.
Update its size.

> Size needs to be: `40`.

How can a file be size of 40?
Each character (or letter) is 1, so you simply need to add 40 characters inside the file. ;-)

In order to print this information, use the command `ls -l` (command is `ls` and `-l` is one option).

Example00

```
$>ls -l
xxxxxxxxx  x xxxxx  xxxxx  40 May  25
14:30 my_first_file_with_content
$>
```

We've replaced some values with `x`, it means they are not important at the moment. You can disregard them :)

Tips

`$>` is a very common way to represent a `prompt`.

How to read the above `Example00`:

> You are in a shell (the `$>`) and you've executed the command `ls -l`.
> It has printed: `xxxxxxxxx  x xxxxx  xxxxx  40 May  25 14:30 my_first_file_with_content`
> And the last line is reprinting the `prompt` (`$>`).

Tips
The terminal is the little box where you execute command at the bottom part of `Docode`.
To test if your exercise(s) is/are correct(s), you can execute the command `gandalf` in your terminal.

Difficulty
Initiation

Average duration
1 Week

Project's Metadata

Project

id: 29

name: quest00

visible: True

Quest00	My First File With Content And Perms
Submit directory	ex02
Submit file	my_first_file_with_content_and_perms

Description

Part 00

Create a file called my_first_file_with_content_and_perms in your directory. Update its size THEN its permissions. (in this order: because you will be asked to remove the permission to WRITE to a file which will be problematic to add a content inside it ;-))

> Size needs to be: `40` (be creative, here, how a file can have a size of 40...? :))
> Permissions need to be: `r--r-xr-x` (Google chmod? :-))

In order to print this information, use the command `ls -l` (command is `ls` and `-l` is one option).

Example00

```
$>ls -l
-r--r-xr-x  1 login  group  40 May  25
14:30
my_first_file_with_content_and_perms
$>
```

Tips

`$>` is a very common way to represent a `prompt`.

How to read the above `Example00`:

> You are in a shell (the `$>`) and you've executed the command `ls -l`.
> It has printed: `-r--r-xr-x  1 login  group  40 May  25`

`14:30 my_first_file_with_content_and_perms`
And the last line is reprinting the `prompt` (`$>`).

Tip

The terminal is the little box where you execute command at the bottom part of `Docode`.

Part 01

Once you've finished the previous steps, execute the following command to create the file that will be submitted:

```
tar -cf
my_first_file_with_content_and_perms.tar
my_first_file_with_content_and_perms
```

Tip
To test if your exercise(s) is/are correct(s), you can execute the command `gandalf` in your terminal.

Quest00	My Gitaddcommitpush
Submit directory	ex03
Submit file	file

Description

Time to Git!

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows. @Wikipedia

0. Create a file with the word "content" as the actual content of the file (disturbing?!)

```
$>echo "content" > file
```

1. Add that file to the commit (feel free to Google what a commit is!)

```
$>git add file
```

2. Prepare the commit with a message (please find below an example of a commit message; you should provide an accurate message for each commit you will do during your software engineer's life)

```
$>git commit –m "I swear I will ALWAYS
add a great message for my commits"
```

3. Push the commit

```
$>git push
```

4. Now we know you know how to use git. No excuses and ALWAYS PUSH YOUR WORK. :-)

Extra:
0. git log

1. git status
2. git clone

Quest00	My Z
Submit directory	ex04
Submit file	my_z

Description

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

```
$>cat my_z
Z
$>
```

"Z" Stands For Zorro. :-)