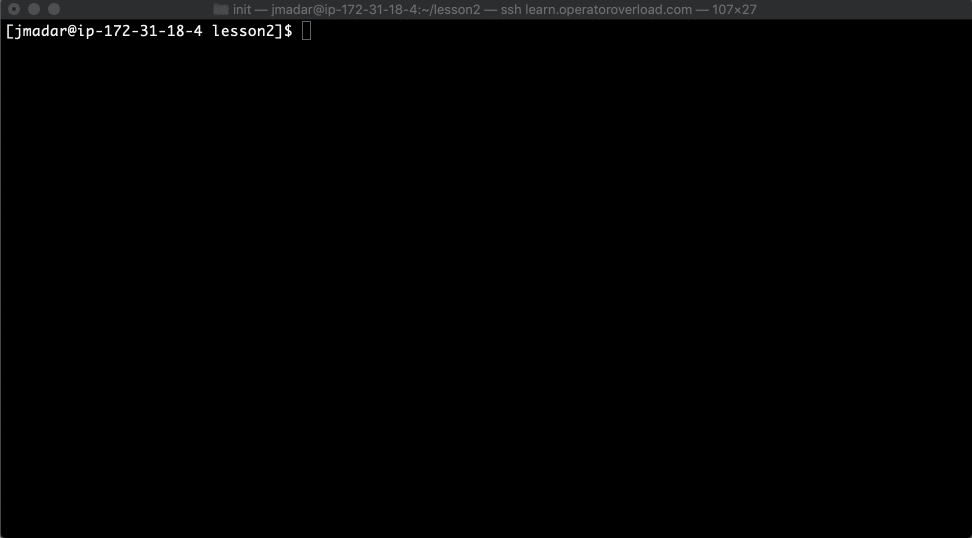
# Bash Scripting

This brings us to the last part of this section. If you haven’t already, please watch the following basic introduction to bash expansion and scripting syntax:

<https://www.linkedin.com/learning/learning-bash-scripting-17063287/bash-expansions-and-substitutions?resume=false&u=57075641>

<https://www.linkedin.com/learning/learning-bash-scripting-17063287/understanding-bash-script-syntax?resume=false&u=57075641>

Let’s say if I want to download all the images from the langara website that we retrieved from the previous section, we can run the **wget** command on every file above, like this:



I got tired typing after 2 downloads, lol.

There is a better way, basically we want to run the command

curl -s https://langara.ca | grep -o -E '/\_files.\*jpg'

We store the output into an array variable, then we want to run wget on every item of the array. This is how we can do it via a shell script, a program that uses shell commands. Here is the code:

# Stores the output into the array call FILES

# NOTE: we use command substitution $( ) to capture the

# output of a command into a variable

FILES=$(curl -s https://langara.ca | grep -o -E '/\_files.\*jpg')

# Loop over the FILES array. In bash, assigning we need

# to prefix a variable with the $ sign to access its content

for F in $FILES

do

# we are inside the loop, and we can now run wget on

# the $F variable. Noticed the use of variable expansion ${}

wget https://langara.ca${F}

done

Save this in a file called download.sh and we can run it using the command **bash download.sh**. As follows:

To make the file easier to run, we can make the file executable. To do this, we need to first add a line in the beginning of the file to indicate to Linux that this file requires the bash program. This is generally called the shebang line:

#!/bin/bash

# Stores the output into the array call FILES

# NOTE: we use command substitution $( ) to capture the

# output of a command into a variable

FILES=$(curl -s https://langara.ca | grep -o -E '/\_files.\*jpg')

# Loop over the FILES array. In bash, assigning we need

# to prefix a variable with the $ sign to access its content

for F in $FILES

do

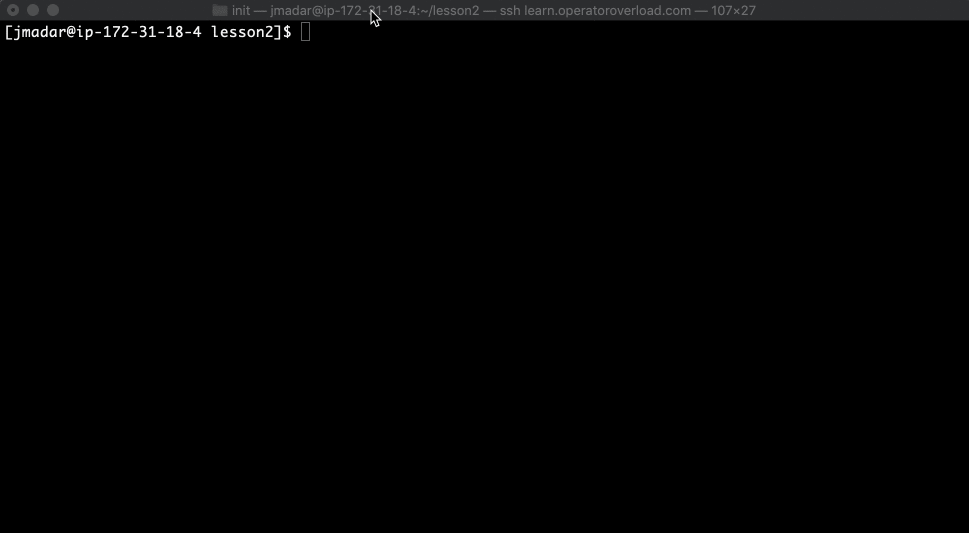
# we are inside the loop, and we can now run wget on

# the $F variable. Noticed the use of variable expansion ${}

wget https://langara.ca${F}

done

We then change the permission of the file so it is executable by Linux, finally we can run it on the command line:



Notice that we have to provide the path to the current directory when calling the script file.