# Working with text

In the previous lesson, you learned about grep and how we can use it to extract information from text files.

Review the following videos for the basics of **grep**, **awk**, and **sed** from LinkedIn Learning:

<https://www.linkedin.com/learning/learning-linux-command-line-14447912/search-for-text-in-files-and-streams-with-grep>

<https://www.linkedin.com/learning/learning-linux-command-line-14447912/manipulate-text-with-awk-sed-and-sort>

The great thing about grep is that you can use it with pipe to search from any input. For example, when you issue the following command in bash:

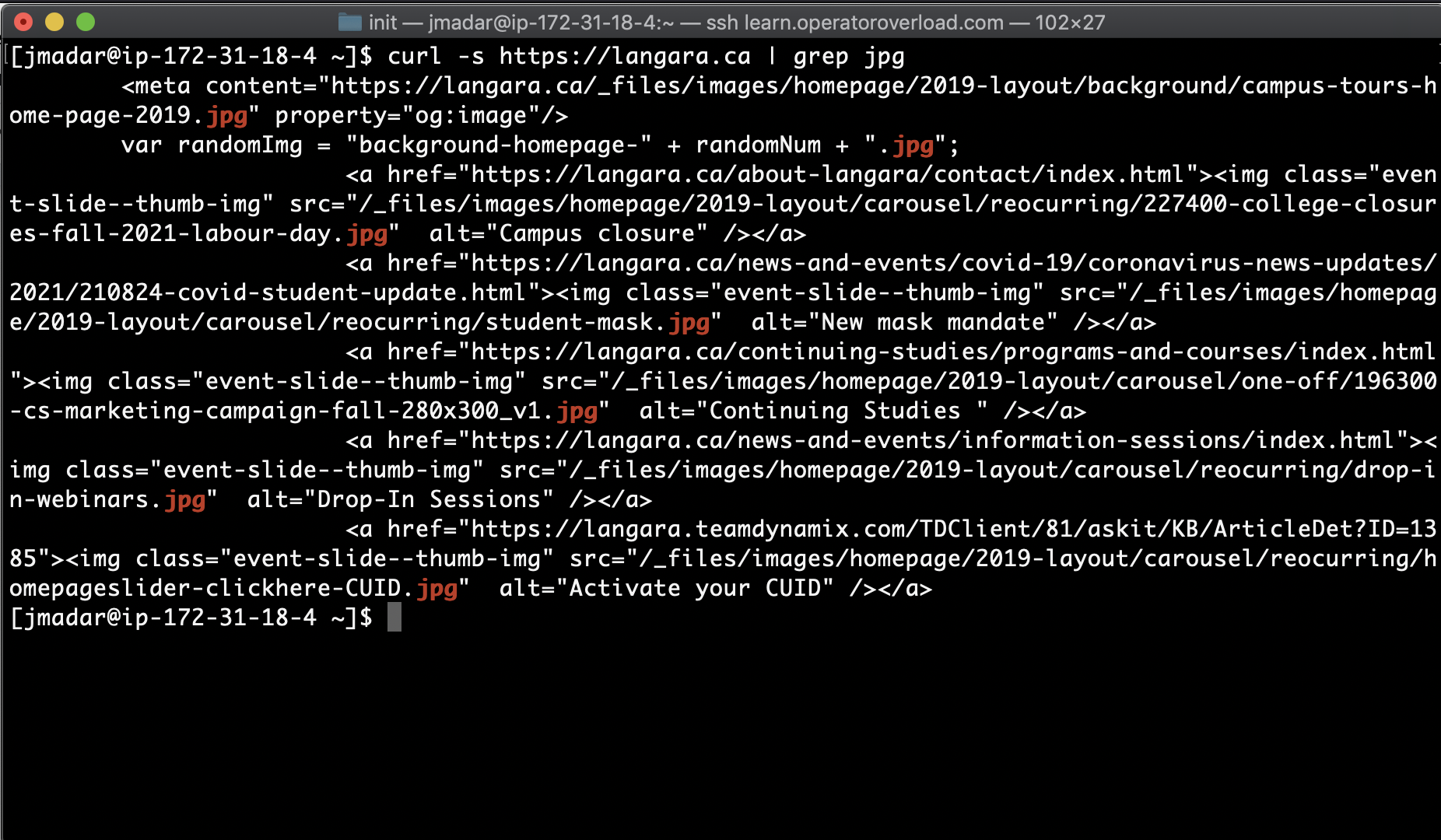
**curl -s https://langara.ca | grep -i covid**

it returns all occurrences of the word covid in the langara main webpage.

The nice thing about grep is that if you use the -E option, you can enter a regex pattern. To illustrate, let’s say you wanted to extract all the jpg file names from the langara home page, you may want use the command

**curl -s https://langara.ca | grep -i jpg**

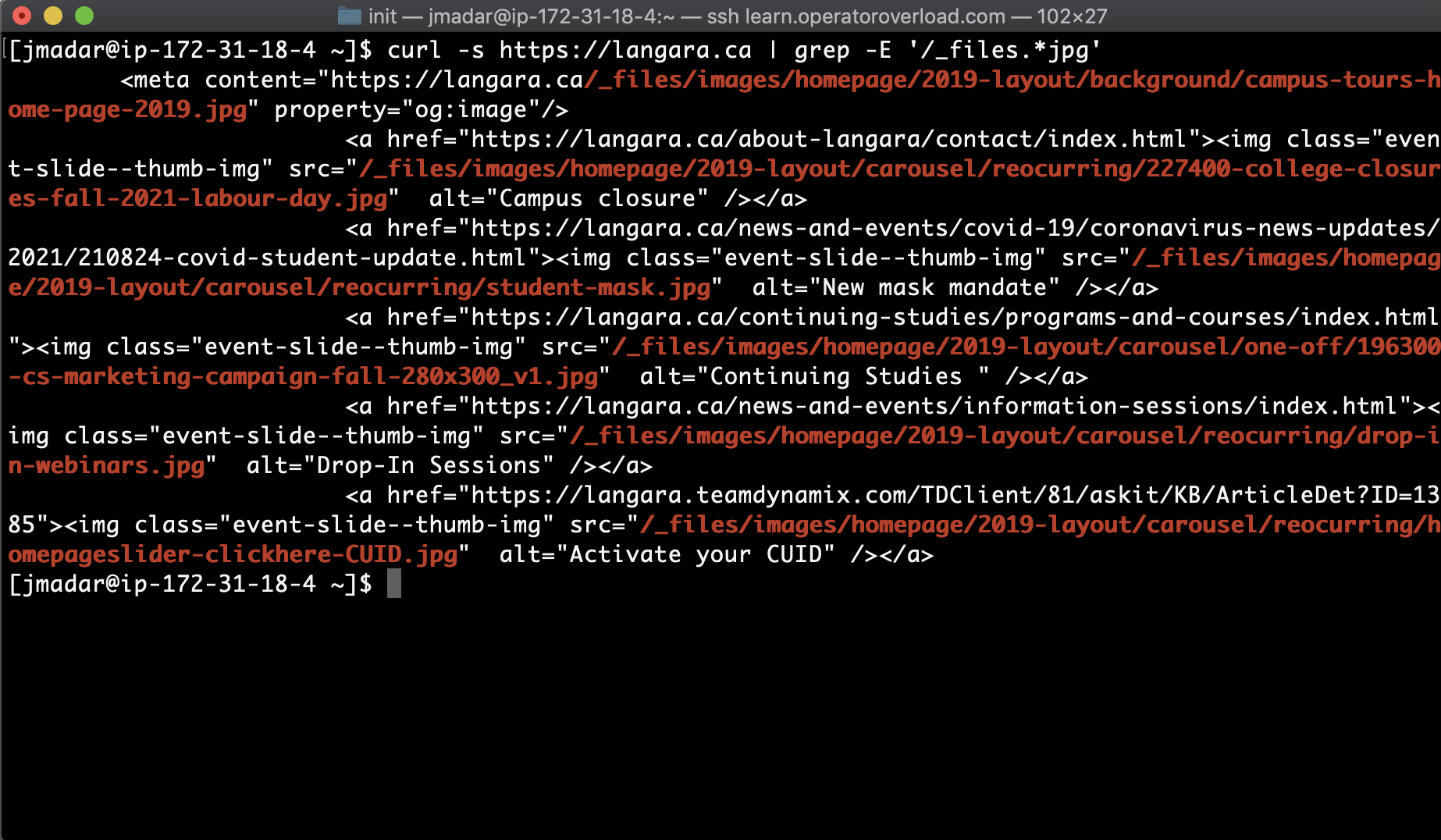
Which will return the following:



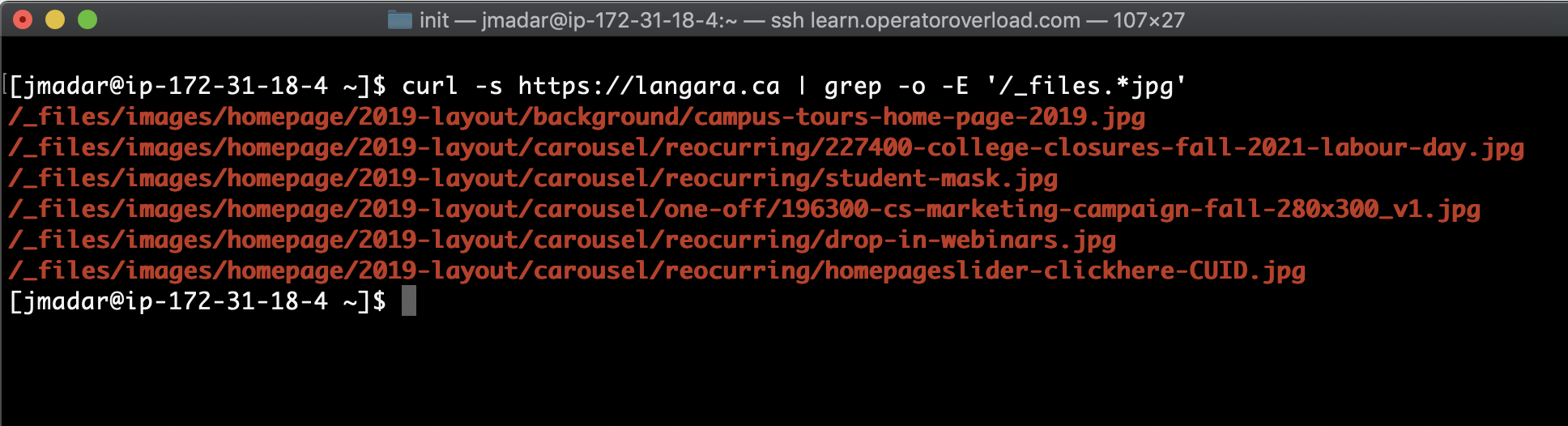
This is clearly not what we want, since the output includes a javascript line as well (line 2).

However, we noticed how all jpg files come from the **/\_files/** directory, we basically want to extract all patterns that have **/\_files/** and end with **jpg**. So we issue the following command (note: in regex, .\* means any characters of any length - see next optional section on regular expression):  
  
 **curl -s https://langara.ca | grep -E '/\_files.\*jpg'**

This gives us the following output



Now we only get lines with valid jpg files. If we add the -o option to grep, it will only output the data that matched the pattern and throw away the rest, like this:



We basically now have a list of image files from langara.ca!

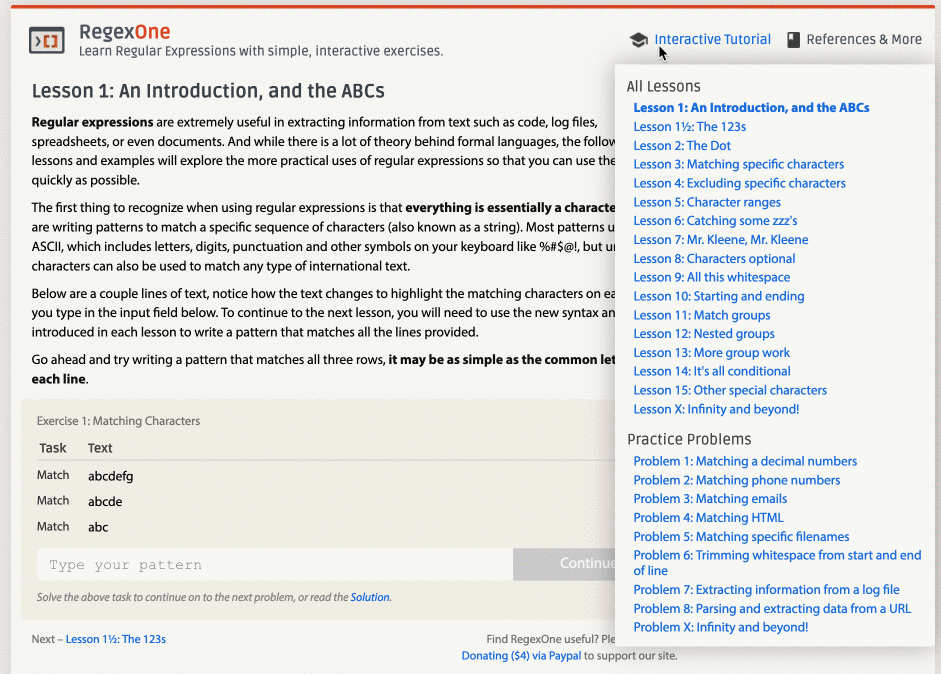
# Regular Expressions (Regex or RE)

The following section goes deeper into regex, which was mentioned in the previous section when we tried to extract images from urls. REs are extremely useful for developers, and will become part of your toolbox for the rest of your career (since we deal with text files all the time).

There are lots of resources on the Internet when it comes to REs, but I found the tutorials at <https://regexone.com/> to be one of the best.

There are a total of 15 lessons and 8 practice problems on the<https://regexone.com/> website. I’d recommend completing all 15 lessons before attempting the practice problems.

To complete this lesson, I would like you to come up with **alternative** patterns that solve each of the 8 practice problems. The following gif should illustrate what I mean (and I have completed Question 1 for you!):



There are a total of 8 questions in this quiz, please put in your solution for each of the questions. If you couldn’t figure it out, please enter into the explanation slot explaining what you have tried.

The goal of this lesson is to illustrate to you how powerful, flexible and potentially confusing REs are, and to give you an appreciation of how hard it is for us to give you an official “answer key” to these problems, since your solution could be completely different than ours but still completely valid.