One issue you may have thought about during our little exploration of regular expressions is how to search for certain punctuation marks in text considering that those same symbols are used as metacharacters! For example, how would you find a plus sign (+) in a line of text since the plus sign is **also** a metacharacter? The answer is simply using a backslash (\) before the plus sign in a regex, in order to "escape" the metacharacter functionality. Here are a few examples:

```
egrep "\+" small.txt

## tragedy + time = humor

egrep "\." small.txt

## http://www.jhsph.edu/
```

There are three more metacharacters that we should discuss, and two of them come as a pair: the caret (^), which represents the start of a line, and the dollar sign ($) which represents the end of line. These "anchor characters" only match the beginning and ends of lines when coupled with other regular expressions. For example, going back to looking at states.txt, I could search for all of the state names that begin with "M" with the following command:

```
egrep "^M" states.txt

## Maine
## Maryland
## Massachusetts
## Michigan
## Minnesota
## Mississippi
## Missouri
## Montana
```

There's a mnemonic that I love for remembering which metacharacter to use for each anchor: "First you get the power, then you get the money." The caret character is used for exponentiation in many programming languages, so "power" (^) is used for the beginning of a line and "money" ($) is used for the end of a line.

Finally, let's talk about the "or" metacharacter (|), which is also called the "pipe" character. This metacharacter allows you to match either the regex on the right or on the left side of the pipe. Let's take a look at a small example:

```
egrep "North|South" states.txt

## North Carolina
## North Dakota
## South Carolina
## South Dakota
```

In the example above we're searching for lines of text that contain the words "North" or "South". You can also use multiple pipe characters to, for example, search for lines that contain the words for all of the cardinal directions:

```
egrep "North|South|East|West" states.txt

## North Carolina
## North Dakota
## South Carolina
## South Dakota
## West Virginia
```

Just two more notes on grep: you can display the line number that a match occurs on using the -n flag:

```
egrep -n "t$" states.txt

## 7:Connecticut
## 45:Vermont
```

And you can also grep multiple files at once by providing multiple file arguments:

```
egrep "New" states.txt canada.txt

## states.txt:New Hampshire
## states.txt:New Jersey
## states.txt:New Mexico
## states.txt:New York
## canada.txt:Newfoundland and Labrador
## canada.txt:New Brunswick
```

You now have the power to do some pretty complicated string searching using regular expressions! Imagine you wanted to search for all of the state names that both begin and end with a vowel. Now you can:

```
egrep "^[AEIOU]{1}.+[aeiou]{1}$" states.txt

## Alabama
## Alaska
## Arizona
## Idaho
## Indiana
## Iowa
## Ohio
## Oklahoma
```

I know there a many metacharacters to keep track of here so below I've included a table with several of the metacharacters we've discussed in this chapter:

Metacharacter	**Meaning**

.	Any Character
\w	A Word
\W	Not a Word
\d	A Digit
\D	Not a Digit
\s	Whitespace
\S	Not Whitespace
[def]	A Set of Characters
[^def]	Negation of Set
[e-q]	A Range of Characters
^	Beginning of String
$	End of String
\n	Newline
+	One or More of Previous
*	Zero or More of Previous
?	Zero or One of Previous
|	Either the Previous or the Following
{6}	Exactly 6 of Previous
{4, 6}	Between 4 and 6 or Previous
{4, }	More than 4 of Previous

If you want to experiment with writing regular expressions before you use them I highly recommend playing around with http://regexr.com/.

Mark as completed