Regular Expressions (Regex)

This lesson gives a conceptual summary of regular expressions.

> Regular expressions are a powerful way to filter out specific pieces of information by using various arithematic patterns to describe certain set of strings.

You are probably now familiar with the *wildcards* e.g. ***.txt** to find all text files in some directory. Its regex would be something like this: `^.*\.txt$.` .

Some very important bash commands like `grep` and `egrep` use regex to search for different patterns of texts in inputs and files. Given below is a summary to make you aware of what various regex notations mean.

Regex Pattern Notations:

Regex Operator	Description
?	The preceding item is optional and matched at most once.
*	The preceding item will be matched zero or more times.
+	The preceding item will be matched one or more times.
	The preceding item is matched

{n}	The preceding item is matched exactly n times.
{n,}	The preceding item is matched n or more times.
{n,m}	The preceding item is matched at least n times, but not more than m times.
$	Matches the end of the line.
^	Matches the beginning of the line.
()	Allows us to group several characters to behave as one.
|	Its the logical OR operation.
.	A single character.
[agd]	The character is one of those included within the square brackets.
[^agd]	The character is not one of those included within the square brackets.
[a-d]	The dash within the square brackets operates as a range. Here, it means all characters between **a** and **d**, including “a” and “d”.