5. QUALIFICATIONS

Character Classes: Ranges

Introduction

We are already seen the idea of a Character Class (or Character Set), it's **a list of characters enclosed in square brackets, and any one of those characters will represent a match to the class**. So, for example, the following Regular Expression:

```
RegEx_Pattern = "Dami(a|e)n"
```

Can be more compactly represented as follows:

```
Character_Class = "Dami[ae]n"
```

Character Ranges

If we wanted to match only with Lowercase letters, we could do either this:

```
RegEx_Pattern =
"(a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z)"
```

or by using a Character Class:

```
RegEx_Pattern = "[abcdefghijklmnopqrstuvwxyz]"
```

However, we can also represent the same range of letters as follows:

```
RegEx_Pattern = "[a-z]"
```

Which means to match to any character in the range "a" to "z". So, in other words, the Regular Expression will match with any single Lowercase letter.

We can create a similar Regular Expression for Uppercase letter, as follows:

```
RegEx_Pattern = "[A-Z]"
```

Which means to match to any character in the range "A" to "Z".

And we can create a similar Regular Expression for numbers, as follows:

```
RegEx_Pattern = "[0-9]"
```

Which means to match to any number in the range "0" to "9".

Combining Ranges

So, if we wanted to search for a String of five characters long, with the first one being Uppercase and the rest being Lowercase, we could do that as follows:

```
RegEx_Pattern = "[A-Z][a-z][a-z][a-z][a-z]"
```

And that Pattern will match to Upper, Lower, Lower, Lower, Lower, for example:

Test_Message = "Hello"	**MATCH** ✓
Test_Message = "Aaaaa"	**MATCH** ✓
Test_Message = "Abcde"	**MATCH** ✓

And it wouldn't match with a String that doesn't match that format:

Test_Message = "AAaaa"	**NO MATCH** ✗
Test_Message = "aaaaa"	**NO MATCH** ✗

And we will see that Character Ranges are very common and very useful in RegExes.

#RegExThursday © Damian Gordon