3. EXAMPLES 1

Examples with Letter Cases

Introduction

When we are working with Strings, it's important to note that Capital Letters (Uppercase Letters) are treated as being different from Lowercase Letters. So, for example, if the Regular Expression is the Uppercase Letter "D", as follows:

```
RegEx_Pattern = "D"
```

This will match with the String "D", but not to the String "d". We say that Regular Expressions are *Case Sensitive* to describe this property of Regular Expressions.

Uppercase and Lowercase Letters

So, the following three Strings should be considered as three different Strings:

- hello
- Hello
- HELLO

And to detect these three Strings, we could use the following Regular Expression:

```
RegEx_Pattern = "hello|Hello|HELLO"
```

And we can reduce the length of that Regular Expression by doing the following:

```
RegEx_Pattern = "(h|H)ello|HELLO"
```

Lowercase Letters

If we had a Notepad file full of text and we wanted to read all of that text into a computer program, and then only print out the letters that are in Lowercase, there is a really easy way to do that using Regular Expressions. We just read in the file one character at a time, and then check if that is a Lowercase character (or not) by comparing it to the following Regular Expression:

```
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)"
```

So, if the character read in is a Lowercase letter, it will match the RegEx pattern above, and we can print it out, and if it is anything other than a Lowercase letter, it will not match and therefore we should not print it out. So, for example:

Test_Message = "a"	**MATCH** ✓
Test_Message = "z"	**MATCH** ✓

And any Uppercase letters, Numbers and Symbols will not match, for example:

Test_Message = "A"	**NO MATCH** ✗
Test_Message = "4"	**NO MATCH** ✗
Test_Message = "@"	**NO MATCH** ✗

We also note that the same Pattern can be created by using a Character Class:

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

And this will work exactly the same way, but the Pattern is more compact.

#RegExThursday © Damian Gordon