6. EXAMPLES 2

Examples with the Curly Braces

Introduction

As we have already seen the curly braces ***are a repetition qualification that specifies the number of instances of the preceding character (or grouping) in a RegEx.***

And we will remember that we can use it in four ways, to indicate that the preceding character (or grouping) appears the following times:

`{num}`	Appears exactly `num` times.
• e.g. `a{5}` matches "aaaaa".	
`{min,}`	Appears at least `min` times.
• e.g. `a{4,}` matches "aaaa", "aaaaa", "aaaaaa", "aaaaaaa", etc.	
`{,max}`	Appears up to `max` times.
• e.g. `a{,4}` matches "a", "aa", "aaa", and "aaaa".	
`{min,max}`	Appears between `min` and `max` times.
• e.g. `a{2,5}` matches "aa", "aaa", "aaaa" and "aaaaa".	

Some Common Examples

If we were searching my emails for "`Yippee`", "`Yippeee`", "`Yippeeee`", and all the way up to "`Yippeeeeeeee`", we could do it simply as:

```
RegEx_Pattern = "Yippe{2,8}"
```

So this will match with "`Yipp`" and between two (2) to eight (8) "`e`"s.

If we were looking for "`CamelCaseCamelCaseCamelCaseCamelCase`", we can avoid a lot of typing (and avoid potential typos) by simply do the following:

```
RegEx_Pattern = "(CamelCase){4}"
```

And this will look for the phrase "`CamelCase`" repeated four (4) times.

Password Length

Let imagine we were writing a computer program to check if a user creates a password that adheres to the following rules:

- *The password has to have between 9 and 15 characters.*
- *The password cannot contain special characters (e.g. "%", "@", etc.).*
- *The password can be any combination of lowercase letters, uppercase letters, and numbers.*

We can express that as a Regular Expression as follows:

```
RegEx_Pattern = "[a-zA-Z0-9]{9,15}"
```

So, this matches any combination of lowercase letters, uppercase letters and numbers, in a range of nine (9) to fifteen (15) characters. Additional rules for passwords that might be added could be things like "*the password should be hard to guess*", and "*the password should not be one you have used before*", but these rules wouldn't necessarily be implemented using Regular Expressions.