2. SIMPLE MATCHING

Using the Period to Match

Introduction

This will be our first use of special characters (“metacharacters”) to match a String with a regular expression pattern, and the special character we are going to look at is the period (“.”) character, or the *Full Stop* character. ***<u>The period metacharacter represents any other single character, which can be a letter, number, or symbol</u>***.

A Simple Example

Sometimes I get emails from people who spell my name wrong, and they call me “Damien” instead of “Damian”, which isn’t a big deal, but if I had a big document full of emails to me, and I wanted to check how often my name is used in emails, I’d have to search for the string “Damian” and the string “Damien”. Regular Expressions give us a nice, simple way of doing this using the period character. We could create a Regular Expression using the period metacharacter as follows:

```
RegEx_Pattern = "Dami.n"
```

So, we can read this Regular Expression as to try to match Strings with the following pattern: “D”, “a”, “m”, “I”, any character, and “n”.

So we would get the following matches:

`Test_Message  = "Damian"`	MATCH ✓
`Test_Message  = "Damien"`	MATCH ✓

And it won’t match with names that don’t fit the pattern:

`Test_Message  = "Damani"`	NO MATCH ✗
`Test_Message  = "Dameon"`	NO MATCH ✗

But, it is worth noting, that the pattern will match many other Strings, including:

`Test_Message  = "Damixn"`	MATCH ✓
`Test_Message  = "Dami5n"`	MATCH ✓
`Test_Message  = "Dami=n"`	MATCH ✓
`Test_Message  = "Dami n"`	MATCH ✓

Since the period can represent any character, including the space character (“ “).

We should also remember, our pattern is six characters long (“D”, “a”, “m”, “I”, any character, and “n”), so the following Strings will not be considered a match:

`Test_Message  = "Damin"`	NO MATCH ✗
`Test_Message  = "Damiaan"`	NO MATCH ✗

As they don’t follow the pattern.