

2. SIMPLE MATCHING

Using the Brackets to Group

Introduction

Our next special characters or “metacharacters” are the brackets - () or []. We’ll be mainly focusing on Round Brackets (also known as *Parentheses*), but we’ll also look at the square brackets [] a little bit as well. The brackets allow us to group together operations, and is thus also called the “Grouping” operator. **The Brackets metacharacters are used to group together parts of a Regular Expression.**

Why Use the Round Brackets?

We will remember that if we are searching for “Damian” or “Damien”, we can use the period metacharacter to give us an *approximate solution*:

```
Regex_Pattern = "Dami.n"
```

And for a *precise solution*, we can use the vertical bar:

```
Regex_Pattern = "Damian|Damien"
```

Using the brackets we can restate the *precise solution* more compactly, as follows:

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

Which can be read as creating a Regular Expression to match to any String with the letters: “D”, “a”, “m”, “i”, (“a” or “e”), and “n”. So, the brackets allow specific parts of the Regular Expression that can be grouped together.

In the example above we are creating a choice between two single characters (“a” and “e”), however, we will see in later examples that the brackets can also be used to create choices between multicharacter Strings as well.

Character Class

A Character Class (or Character Set) is a list of characters enclosed in square brackets, and any one of those characters will represent a match to the class. So for the moment we can say that the Regular Expression with the single characters above, and Character Class below are equivalent in terms of their functionality:

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

This can be read as creating a Character Class to match to any String with the letters “D”, “a”, “m”, “i”, (“a” or “e”), and “n”.

Matching the Brackets Character

If we are searching for the Round Brackets characters (not the metacharacter), then mathematically we represent the Open Bracket as follows: \ (

and for the Close Bracket we can mathematically represent it as follows: \)

However, most programming languages prefer we state the Open Bracket as:

```
Regex_Pattern = "\\("
```

and most programming languages prefer we state the Close Bracket as:

```
Regex_Pattern = "\\)"
```

And it is the same for the Square Brackets: “\[” and “\]”.

#RegExThursday © Damian Gordon