# 8. ADVANCED FEATURES

## Character Classes: Advanced

### 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"
```

For a String of five characters long, with the first Uppercase and the rest Lowercase:

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

And that Pattern will match to Upper, Lower, Lower, Lower, Lower character,

### Chemical Symbols

The chemical elements are represented by a one-letter or two-letter symbol, so for example, Oxygen is O, Hydrogen is H, Sodium is Na, and Lead is Pb. As a RegEx:

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

So the pattern is one uppercase letter, and one optional lowercase letter. When people are trying to develop patterns like the pattern above `[A-Z][a-z]?`, sometimes they get confused and write it as `[A-Za-z]?`, which has a very different meaning, and it's saying match with or of two things:

- Match with a single character either uppercase or lowercase, or
- Match with a string with no characters in it (a "*Null String*").

### Negation in Character Classes

We have already seen the caret symbol (**^**) being used to indicate the beginning of a String boundary, but if this symbol is used inside a Character Class (i.e. inside the square brackets) it has another meaning; it means that a String will match with anything except the pattern in the Character Class. So, for example:

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

Means match any single uppercase character, and:

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

Means match anything single character except uppercase characters, so it will match to any of the lowercase characters, any numbers, any symbols, and any whitespace. So, if we wanted to remove all the vowels from an input String, we could do:

```
RegEx_Pattern = "[^aeiou]"
```

And that would match everything except vowels.

### Non-Literal Characters ^ - ] \

Some symbols inside Character Classes are treated literally, so for example, if we were looking for a period character outside of a Character Class, we would do `\\.` but inside a Character Class we can just do `[.]` All symbols are treated literally in Character Classes except for the following symbols: the caret (`^`), the hyphen (`-`), the close square bracket (`]`), and the backslash (`\`), which all have special meanings.