CHAPTER 1

re – Regular Expressions

1.1 RE – REGULAR EXPRESSION OPERATIONS

This module provides regular expression matching operations similar to those found in Perl.

Both patterns and strings to be searched can be Unicode strings as well as 8-bit strings. However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match an Unicode string with a byte pattern or vice-versa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

1.1.1 REGULAR EXPRESSION SYNTAX

A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).

Some characters, like `'|'` or `'('`, are special. Special characters either stand for classes of ordinary characters, or affect how the regular expressions around them are interpreted. Regular expression pattern strings may not contain null bytes, but can specify the null byte using a `\number` notation such as `'\x00'`.

The special characters are:

`'.'` (Dot.) In the default mode, this matches any character except a newline. If the DOTALL flag has been specified, this matches any character including a newline.

`'$'` Matches the end of the string or just before the newline at the end of the string, and in MULTILINE mode also matches before a newline. `foo` matches both `'foo'` and `'foobar'`, while the regular expression `foo$` matches only `foo'`. More interestingly, searching for `foo.$` in `'foo1\nfoo2\n'` matches `'foo2'` normally, but `'foo1'` in MULTILINE mode; searching for a single `$` in `'foo\n'` will find two (empty) matches: one just before the newline, and one at the end of the string.

`[ ]` Used to indicate a set of characters. In a set:

- Characters can be listed individually, e.g. [amk] will match `'a'`, `'m'`, or `'k'`.
- Ranges of characters can be indicated by giving two characters and separating them by a `'-'`, for example `[a-z]` will match any lowercase ASCII letter, `[0-5][0-9]` will match all the two-digits numbers from 00 to 59, and [0-9A-Fa-f] will match any hexadecimal digit. If - is escaped (e.g. `[a\-z]`) or if it is placed as the first or last character (e.g. `[a-]`), it will match a literal `'-'`.