TCL - REGULAR EXPRESSIONS

The "regexp" command is used to match a regular expression in Tcl. A regular expression is a sequence of characters that contains a search pattern. It consists of multiple rules and the following table explains these rules and corresponding use.

SN	Rule	Description
1	**x**	Exact match.
2	**[a-z]**	Any lowercase letter from a-z.
3	**.**	Any character.
4	**^**	Beginning string should match.
5	**$**	Ending string should match
6	**\^**	Backlash sequence to match special character ^.Similarly you can use for other characters.
7		Add the above sequences inside parenthesis to make a regular expression.
8	**x***	Should match 0 or more occurrences of the preceding x.
9	**x+**	Should match 1 or more occurrences of the preceding x.
10	**[a-z]?**	Should match 0 or 1 occurrence of the preceding x.
11	**{digit}**	Matches exactly digit occurrences of previous regex expression. Digit that contain 0-9.
12	**{digit,}**	Matches 3 or more digit occurrences of previous regex expression. Digit that contain 0-9.
13	**{digit1,digit2}**	Occurrences matches the range between digit1 and digit2 occurrences of previous regex expression.

Syntax

The syntax for regex is given below.

```
regexp optionalSwitches patterns searchString fullMatch subMatch1 ... subMatchn
```

Here, regex is the command. We will see about optional switches later. Patterns are the rules as mentioned earlier. Search string is the actual string on which the regex is performed. Full match is any variable to hold the result of matched regex result. Submatch1 to SubMatchn are optional subMatch variable that holds the result of sub match patterns.

Let's look at some simple examples before diving into complex ones. A simple example for a string with any alphabets. When any other character is encountered the regex search will be stopped and returned.

```
#!/usr/bin/tclsh

regexp {([A-Z,a-z]*)} "Tcl Tutorial" a b
puts "Full Match: $a"
puts "Sub Match1: $b"
```

When above code is executed, it produces following result.

```
Full Match: Tcl
Sub Match1: Tcl
```

Multiple patterns

The following example shows how to search for multiple patterns. This is example pattern for any alphabets followed by any character followed by any alphabets.

```
#!/usr/bin/tclsh

regexp {([A-Z,a-z]*).([A-Z,a-z]*)} "Tcl Tutorial" a b c
puts "Full Match: $a"
puts "Sub Match1: $b"
puts "Sub Match2: $c"
```

When above code is executed, it produces following result.

```
Full Match: Tcl Tutorial
Sub Match1: Tcl
Sub Match2: Tutorial
```

A modified version of the above code to show that a sub pattern can contain multiple patterns is shown below.

```
#!/usr/bin/tclsh

regexp {([A-Z,a-z]*.([A-Z,a-z]*))} "Tcl Tutorial" a b c
puts "Full Match: $a"
puts "Sub Match1: $b"
puts "Sub Match2: $c"
```

When above code is executed, it produces following result.

```
Full Match: Tcl Tutorial
Sub Match1: Tcl Tutorial
Sub Match2: Tutorial
```

Switches for Regex command

The list of switches available in Tcl are,

- **-nocase :** Used to ignore case.
- **-indices :** Stores location of matched sub patterns instead of matched characters.
- **-line :** New line sensitive matching. Ignores the characters after newline.
- **-start index :** Set the offset of start of search pattern
- **-- :** Marks the end of switches

In the above examples, I have deliberately used [A-Z,a-z] for all alphabets, you can easily use -nocase instead as shown below.

```
#!/usr/bin/tclsh

regexp -nocase {([A-Z]*.([A-Z]*))} "Tcl Tutorial" a b c
puts "Full Match: $a"
puts "Sub Match1: $b"
puts "Sub Match2: $c"
```

When above code is executed, it produces following result.

```
Full Match: Tcl Tutorial
Sub Match1: Tcl Tutorial
Sub Match2: Tutorial
```

Another example using switches is shown below.

```
#!/usr/bin/tclsh

regexp -nocase -line -- {([A-Z]*.([A-Z]*))} "Tcl \nTutorial" a b
puts "Full Match: $a"
puts "Sub Match1: $b"
regexp -nocase -start 4 -line -- {([A-Z]*.([A-Z]*))} "Tcl \nTutorial" a b
puts "Full Match: $a"
puts "Sub Match1: $b"
```

When above code is executed, it produces following result.

```
Full Match: Tcl
Sub Match1: Tcl
Full Match: Tutorial
Sub Match1: Tutorial
```