



Weeks 9:

Regular Expressions



ExampleofRegExp.vbs

- Regular Expression refs
- Lomax p.379 ff
- MSDN VB scripting reference.



ExampleofRegExp.vbs

```
Dim MySearch, SearchResults
Set MySearch = new RegExp
' now my search is an Regular Expression object
```

RegExp objects have four properties:

```
MySearch.IgnoreCase = False
' I care about case
MySearch.Multiline = False
' only one line
MySearch.Global = True
' search for all occurrence
MySearch.Pattern = searchPattern of symbols (p.381-2 Lomax)
```



ExampleofRegExp.vbs

```
Dim MySearch, SearchResults
Set MySearch = new RegExp
RegExp objects have three methods:
```

```
MySearch Execute (SearchString)
' Searches SearchString for occurrences of pattern
' Returns a collection object of matches

MySearch Replace (SearchString, Replacement String)
' Searches SearchString for occurrences of pattern
' Replaces matches with replacementString

MySearch Test (SearchString)
' Searches SearchString for occurrences of pattern
' Returns True if found, False otherwise
```



ExampleofRegExp.vbs

```
Dim MySearch, SearchResults
Set MySearch = new RegExp
.Execute method returns collection of matches:
Set SearchResults = MySearch.Execute(SourceString)
```

The collection has *count* and *item()*

```
msg = msg & "There are " & searchResult.Count & _
" entries in the matches array" & vbCrLf & vbCrLf
```

The collection can be run through with For-each loop:

```
For Each match in searchResult
...
...
Next
```



ExampleofRegExp.vbs

```
Dim MySearch, SearchResults
Set MySearch = new RegExp
.Set SearchResults = MySearch.Execute(SourceString)
```

Each item() in the collection has three properties:

FirstIndex, *Length*, *Value*:

```
For Each match in searchResult
msg = msg & "Index At= " & match.FirstIndex & _
":Length= " & match.Length & _
" Value=[" & match.Value & "]" & vbCrLf
Next
```



ExampleofRegExp.vbs

```
SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5.  
What is so interesting about sTrings Anyway?"
```

```
MySearch.Pattern = "est"  
Call DoSearch (MySearch, SourceString, pattern)
```

```
Sub DoSearch (ByRef MySearch, ByVal SourceString, ByVal  
searchPattern)
```

```
MySearch.Pattern = searchPattern
```

```
Set SearchResults = MySearch.Execute(SourceString)
```

```
PrintSearchResult SearchResults, searchPattern, _  
"STRING: " & sourceString
```



ExampleofRegExp.vbs

```
SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5.  
What is so interesting about sTrings Anyway?"
```

```
Sub PrintSearchResult(ByRef searchResult, ByVal pattern, ByVal label)
```

```
msg = "With Pattern: " & pattern & vbCrLf
```

```
msg = msg & "There are " & searchResult.Count & _  
,
```

```
For Each match in searchResult
```

```
msg = msg & "Index At= " & match.FirstIndex & ": Length= " &  
match.Length & _
```

```
" Value=[" & match.Value & "]" & vbCrLf
```

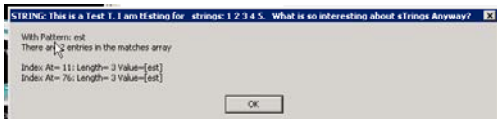
```
Next
```

```
MsgBox msg, , label
```



ExampleofRegExp.vbs

```
SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5.  
What is so interesting about sTrings Anyway?"
```





ExampleofRegExp.vbs

SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5.
What is so interesting about sTrings Anyway?"

- ' [a-z] - means everything between a to z
- ' [a-d] - means everything between a to d
- ' [a-z]* - means zero or more occurrence of characters between a-z
- ' [a-z]+ - means one or more occurrence of characters between a-z
- ' [0-9]+ - means one or more occurrence of numbers between 0 to 9
- ' [0-4]+ - means one or more occurrence of numbers between 0 to 4
- ' T[a-z]* - looks for "T" followed by zero or more characters between a-z
- ' T[a-z]+ - looks for "T" followed by one or more character between a-z

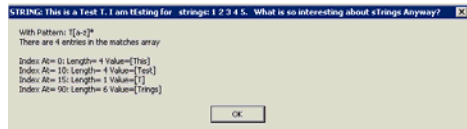
See Tables on pp 381-83 of Lomax



ExampleofRegExp.vbs

SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5.
What is so interesting about sTrings Anyway?"

```
MySearch.Pattern = "T[a-z]*"
Set SearchResults = MySearch.Execute(SourceString)
PrintSearchResult SearchResults, "Result from looking for: T[a-z]*"
```



See Tables on pp 381-83 of Lomax



ExampleofRegExp.vbs

SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5.
What is so interesting about sTrings Anyway?"

```
MySearch.Pattern = "s[a-z]? ."
Set SearchResults = MySearch.Execute(SourceString)
PrintSearchResult SearchResults, "Result from looking for: s[a-z]? ."
```

- ' . - means any character
- ' ? - means zero or one times
- ' T.? - Looks for "T" followed by zero or one character
- ' Th.?s - Look for "Ths" or "Th"-any-one-character-including-blank"s
- ' [0-9]? - Looks for
- ' "s[a-z]? ." - looks for s-followed-by-zero-or-one-character-followed-by-a-space-then-exactly one character

Example of RegExp.vbs

```
SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5. What is so interesting about sTrings Anyway?"

MySearch.Pattern = "[a-z]?"
Set SearchResults = MySearch.Execute(SourceString)
PrintSearchResult SearchResults, "Result from looking for: s[a-z]?"
```

```
STRING: This is a Test T. I am tESting for strings: 1 2 3 4 5. What is so interesting about sTrings Anyway?
With Pattern: s[a-z]?
There are 5 entries in the matches array
Index At= 3; Length= 3 Value=[s ]
Index At= 6; Length= 3 Value=[s a]
Index At= 12; Length= 4 Value=[at ]
Index At= 66; Length= 3 Value=[s s]
Index At= 95; Length= 3 Value=[s A]
```

See Tables on pp 381-83 of Lomax

Example of RegExp.vbs

```
SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5. What is so interesting about sTrings Anyway?"
```

```
MySearch.Pattern = "\t"
Set SearchResults = MySearch.Execute(SourceString)
PrintSearchResult SearchResults, "Result from looking for:\t"
```

```
' \t - means tab
' \n - means newline
' " " - is a space
' \r - means carriage return
' \v - means vertical bar "|"
```

Example of RegExp.vbs

```
SourceString = "This is a Test T. I am tESting for strings: 1 2 3 4 5. What is so interesting about sTrings Anyway?"
```

```
MySearch.Pattern = "\t"
Set SearchResults = MySearch.Execute(SourceString)
PrintSearchResult SearchResults, "Result from looking for:\t"
```

```
STRING: This is a Test T. I am tESting for strings: 1 2 3 4 5. What is so interesting about sTrings Anyway?
With Pattern: \t
There are 1 entries in the matches array
Index At= 35; Length= 1 Value=[ ]
```

WorkingWithRegExp.vbs

200064789 Ulcvpohu Bothell
 200053171 Qukvelkm Kenmore

MySearch.Pattern = "[\t]*[a-zA-Z0-9]+[\t]*"

WorkingWithRegExp.vbs

200064789 Ulcvpohu Bothell
 200053171 Qukvelkm Kenmore

```

' now let's get rid of all those spaces
MySearch.Pattern = "[\t ]*"
' here we are saying [space<tab>] zero or more of these
Index = 0
For Each match in SearchResults
  ArrayOfFields(Index) = MySearch.Replace(match.value, "")
  ' for each string from the matches, replace the tabs and spaces
  ' with empty string
  Index = Index + 1
Next

' Here we have our results!
MsgBox "Id = [" & ArrayOfFields(0) & "] Name=[" & ArrayOfFields(1) & "]
and City=[" & ArrayOfFields(2) & "]"
  
```

WorkingWithRegExp.vbs

200064789 Ulcvpohu Bothell
 200053171 Qukvelkm Kenmore

MySearch.Pattern = "[\t]*[a-zA-Z0-9]+[\t]*"

MsgBox "Id = [" & ArrayOfFields(0) & "] Name=[" & ArrayOfFields(1) & "] and City=[" & ArrayOfFields(2) & "]"

Repeated for line 2.



Activity 15
