WSL (Web Scraping Language)

Nandan Naik (nan2118)

Abstract

WSL is a Web Scraping Language that allows retrieval of HTML pages on the web, extraction of data from the retrieved pages and persistence of extracted data to files.

Motivation

A large amount of data is available on the web in a loosely structured form in HTML pages. While this data is largely meant for human consumption, some websites make it available in a structured format via “Web Services” using mechanisms such as REST and SOAP. These web services allow for programmatic interaction with the data.

However, a significant number of websites do not make such web services available, but the data in them are interesting nevertheless. In such cases, Web Scrapers are written to extract information in them and to load the data into more structured stores (ex. Databases) so something useful can be learned from it.

The variations in the structure of HTML pages on the web disallow a generic one-size-fits-all algorithm from being used to extract the data; data must be extracted on a case-by-case basis. WSL allows the retrieval of the contents of an HTML page, processing of the retrieved HTML for data extraction and persisting of extracted data to a flat-file (which can then be used to load the data into a more structured store such as an RDBMS).

Examples of WSL Syntax and Semantics

Scraping, String Processing and File Output Primitives

getPage URL
: Returns a string containing the contents of an HTML page pointed to by URL.

strReplace Content, MatchString, ReplacementString
: Replaces any occurrences of MatchString in Content with ReplacementString. Returns Content after the replacement.

regexMatch Content, MatchRegEx
: Returns a string which matches the MatchRegEx.

regexReplace Content, MatchRegEx, ReplacementString

Replaces any text that matches MatchRegEx in Content with ReplacementString.
Returns Conent after the replacement.

appendFile FilePath, String

Appends the given string to the end of the file specified by the path.
Returns 0 on success and non-zero on failure.

Types

```
string SomeString
int SomeInt

string AStringArray[5]
```

A string array having size 5.

```
int AnIntArray[5]
```

An int array having size 5.

Control Flow

Example of an If then-else-endIf statement

```
if SomeInt == 0
    # WSL statement(s)
else
    # WSL statement(s)
endIf
```

Example of a While statement

```
while Counter < 0
    # WSL statement(s)
endWhile
```

Example of a function

```
function SomeFunction
    # WSL statement(s)
endFunction
```

Example that calls a function

```
callFunction SomeFunction
```

Operators

+	Adds two integers
-	Subtracts one integer from another
&	Concatenates two strings
==	Checks if two integers are equal
!=	Checks if two integers are not equal
eq	Checks if two strings are equal
neq	Checks if two strings are not equal
<	Checks if one integer is less than another
>	Checks if one integer is greater than another
<=	Checks if one integer is less than or equal to another
>=	Checks if one integer is greater than or equal to another
and	Used to indicate that two expressions must both be true
or	Used to indicate that either of two expressions can be true

Example WSL Program

```
# Many Html Pages have a meta tag in the head section to communicate to search engines what keywords are most relevant to the contents of the page. These keywords can be useful to categorize an HTML page, or categorize a web-site in general (based on the most frequent keywords in all its HTML pages). This program written in WSL fetches an HTML page, reads the keywords in the meta tag for the page, and writes it to a file on disk.

# All variables are global and must be declared outside functions.
string URL
string HtmlPage
string MetaKeywords
string File
string Temp

# This is a user defined function to get the keywords from a meta tag in the head section.
# All functions must be defined before their use. Functions do not have any input or output parameters since all variables have global scope.
function getMetaKeywords
	Temp = regexMatch HtmlPage, "<meta name=""keywords"" content=""[.]*"" />"
	Temp = regexMatch Temp, "content=""[a-zA-Z,0-9]*"""
	Temp = strReplace Temp, "content=""", ""
	MetaKeywords = strReplace Temp, """", ""
endFunction
```

```
# The main function must always be present. It is the entry point for a WSL program.
function main
	URL = “http://finance.yahoo.com/q?s=msft”
	File = “c:\keywords.txt”

	HtmlPage = getPage URL

	if HtmlPage neq “”
		callFunction getMetaKeywords
		appendFile File, MetaKeywords
	endIf
endFunction
```