Parsing Text (HTML Parsing)

by Jerry Muelver

-
  jmuelver

Table of Contents

Parsing Overview

This article presumes to give the rationale and procedure for parsing HTML to extract references to images.

Suppose you've got a nice text string, and you know there's something (or several somethings) in the string that you want to extract and process. If you do it by visual inspection of printed text, you would scan or read through the text to find the pattern (symbols or letters or words) that marks or identifies the target. When the text is in a computer file or string, the process of scanning the text is called parsing, which means "to take apart and analyze according to structure".

For instance, if you want to find a URL buried in HTML source, you'd look for ""
truncate to the left of the first blank space, if one exists
copy string tag$ from html$ using tagpos+9 and tagend
cut off enclosing quotes, if needed
check for "HTTP:"
add urlPath to front of tag$
add itag$ to a list of found links

DEMO

Suppose we've got an HTML file downloaded and saved as "mypage.htm". Suppose further that we know the file came from "http://www.msn.com", and that we'd like to get a list of graphics referenced on that page. This is one way to do it:

```
' HTML parsing demo
' setup
DIM filelist$(300)
count = 1
'replace with the URL for the webpage you are using for the demo
myURL$ = "http://www.msn.com/"

' read the source into a string html$
open "file.html" for input as #f
html$ = input$(#f, LOF(#f))
close #f

' copy the file into string par$ for parsing
' convert parstring into uppercase
par$ = UPPER$(html$)

' use INSTR to find position tagpos of "<IMG SRC="
' use INSTR to find position endpos of next ">"
' copy string tag$ from html$ using tagpos+9 and tagend
' check for blank space and truncate with LEFT$, if needed
' cut off enclosing quotes, if needed
tagpos = INSTR(par$,"<IMG SRC=")
while tagpos > 0
   endpos = INSTR(par$,">",tagpos)
   tag$ = MID$(html$,tagpos+9,endpos-tagpos-9)
   blank = INSTR(tag$," ")
   if blank > 0 then tag$ = left$(tag$,blank-1)
   filelist$(count) = tag$

   ' cut off enclosing quotes, if needed
   if LEFT$(tag$,1) = chr$(34) then
      tag$ = MID$(tag$,2)
   end if
   if RIGHT$(tag$,1) = chr$(34) then
      tag$ = LEFT$(tag$,LEN(tag$)-1)
   end if

   ' check for "HTTP:"
   ' add urlPath to front of tagstring to make imgUrl
   if LEFT$(UPPER$(tag$),5) <> "HTTP:" then
      tag$ = myURL$ + tag$
   end if

   ' add tag$ to a list of found links
   filelist$(count) = tag$
   count = count + 1
   tagpos = INSTR(par$,"<IMG SRC=",endpos)
wend

for x = 1 to count -1
   print filelist$(x)
next
```

To test it out, save a web page from your browser with File... SaveAs > Web Page, rename the saved file to file.html, and run the demo.

You can also download a file in your program, using the method in Downloading A File.

Where to Go from Here

The core notion in parsing text is to use the syntactical patterns in the text by nailing a starting pattern, marking up to an ending pattern, and grabbing everything between the two. INSTR is your friend, and MID$ is your workhorse. To expand on the idea, you could:

- parse strings individually, as they are read in from a file, instead of parsing the whole file at once
- rig the parser to grab text-only from the page (hint: start$ = ">", end$ = "