
Subject: Re: extracting strings between ' '

Posted by [Benjamin Hornberger](#) on Sun, 20 Nov 2005 05:22:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

kl_tah@hotmail.com wrote:

> I was wondering if there's a way to extract a string between inverted
> commas (or any other symbol for that matter) from a longer string in
> idl?
> e.g. extracting just aaaa from the longer string xxxx = 'aaaa'
>

I can offer the function below, which might help. If you know that there is exactly one matching pair of quotes in the string, you might be better off locating the position of the quotes (strpos) and extracting everything in between (strmid). You can also use regular expressions, but that's a more complicated topic. If you're interested, check out <http://rsinc.com/codebank/search.asp?search=category&product=IDL&catid=48>

Benjamin

```
;+
; NAME:
;   QUOTESPLIT
;
;
; PURPOSE:
;
;   This function splits a scalar string into an array of substrings,
;   similar to STRSPLIT(). However, substrings enclosed in single or
;   double quotes are not split. Delimiters can be specified.
;
;
; AUTHOR:
;
;   Benjamin Hornberger
;   benjamin.hornberger@stonybrook.edu
;
;   Inspired by an example in the PHP documentation
;
;
; CATEGORY:
;
;   Utilities, string processing
;
;
; CALLING SEQUENCE:
```

```

;
; Result = QUOTESPLIT(String [, Delimiters])
;
;
;
; RETURN VALUE:
;
; A string array holding all the substrings of the String
; argument. Each substring is trimmed of leading or trailing blanks
; unless the blanks are within quotes.
;
;
; INPUT PARAMETERS:
;
; String: String to be split. Must be a scalar string or one
; element string array.
;
;
; OPTIONAL INPUT PARAMETERS:
;
; Delimiters: A string holding all characters which serve as
; delimiters. Only single characters can be delimiters. The
; String argument will be split on each occurrence of one or more
; delimiters except within quotes (single or double). If a
; string array is given, it will be joined by STRJOIN()
; internally. Example: if Delimiters is " ,;", the string will
; be split on each occurrence of a space, comma or semicolon, or
; any combination thereof (except within quotes). Default:
; Comma, Semicolon, Space, Tab, Carriage Return and Line Feed.
;
;
; OUTPUT KEYWORDS:
;
; COUNT: Set this keyword to a named variable which, upon return,
; will hold the number of elements in the string array. NOTE: If
; the original string consisted only of delimiters, the return
; value will be an empty string, but COUNT will be zero!
;
;
; SIDE EFFECTS:
;
; If the Delimiters argument is passed as string array, it will be
; joined by QUOTESPLIT.
;
; If an opened quote is not closed till the end of the string, no
; error will be raised. The return value is as if the quote was
; closed at the end.
;
;
;

```

```
; EXAMPLES:
;
; (quotes in the output show space characters)
;
; IDL> petstring = 'cat dog "ground hog" "bird"'
; IDL> pets = QUOTESPLIT(petstring)
; IDL> FOR i = 0, n_elements(pets)-1 DO print, pets[i]
;   'cat'
;   'dog'
;   'ground hog'
;   'bird'
;
; IDL> petstring = " cat , dog ; 'ground hog' : ' bird ' ,"
; IDL> pets = QUOTESPLIT(petstring, " ;,:")
; IDL> FOR i = 0, n_elements(pets)-1 DO print, pets[i]
;   'cat'
;   'dog'
;   'ground hog'
;   ' bird '
;
;
; MODIFICATION HISTORY:
;
; 2005-02-04 BH: Written, translated to IDL with some modifications
;   from http://php.net/split (User Contributed Note from
;   "moritz").
; 2005-03-21 BH: Added keyword COUNT
;-
```

FUNCTION quotesplit, string, delimiters, count=count

```
    on_error, 2

    IF n_params() EQ 0 THEN $
        message, 'STRING argument required in function QUOTESPLIT'
    IF n_elements(string) NE 1 THEN $
        message, 'STRING argument must be scalar in function QUOTESPLIT'

    count = 0 ;; walk through all characters in string
    length = strlen(string)
    ;; check delimiters
    IF n_elements(delimiters) EQ 0 THEN $
        delimiters = ' ; '+string(9B)+string(10B)+string(13B)
    IF n_elements(delimiters) GT 1 THEN delimiters = strjoin(delimiters)
    WHILE count LT length DO BEGIN
        ;; pass over all delimiters
        WHILE (count LT length && strpos(delimiters, strmid(string,
count, 1)) NE -1) DO count++
```

```

;; double quotes
IF strmid(string, count, 1) EQ "" THEN BEGIN
    count++
    start = count
    WHILE (count LT length && strmid(string, count, 1) NE "") DO
count++
        IF n_elements(array) EQ 0 THEN $
            array = [strmid(string, start, count-start)] ELSE $
            array = [array, strmid(string, start, count-start)]
            count += 2 ;; jump over 2nd quote
        ENDIF ELSE IF strmid(string, count, 1) EQ "" THEN BEGIN ;;
single quotes
    count++
    start = count
    WHILE (count LT length && strmid(string, count, 1) NE "") DO
count++
        IF n_elements(array) EQ 0 THEN $
            array = [strmid(string, start, count-start)] ELSE $
            array = [array, strmid(string, start, count-start)]
            count += 2 ;; jump over 2nd quote
        ENDIF ELSE BEGIN ;; all other characters
            start = count
            WHILE (count LT length && strpos(delimiters, strmid(string,
count, 1)) EQ -1) DO count++
                IF count GT start THEN $
                    IF n_elements(array) EQ 0 THEN $
                        array = [strmid(string, start, count-start)] ELSE $
                        array = [array, strmid(string, start, count-start)]
                    count++
                ENDELSE
            ENDWHILE

;; array could still be undefined here if string had only delimiters
;; in it
IF size(array, /type) EQ 0 THEN BEGIN
    array = ""
    count = 0
ENDIF ELSE BEGIN
    count = n_elements(array)
ENDELSE

return, array

END

```
