

---

Subject: Re: splitting strings

Posted by [David Fanning](#) on Thu, 03 Feb 2005 20:48:46 GMT

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

---

Benjamin Hornberger writes:

> I would like to split a string by whitespace characters, while anything  
> between quotes should be recognized as one elements (even if it contains  
> whitespace). Let's say I have the string  
>  
> 'cat dog "ground hog" bird'  
>  
> I want to split it into ['cat', 'dog', 'ground hog', 'bird']. Does  
> anybody have a general algorithm or a function to do that? Or do I have  
> to work it out myself?

```
str = 'cat dog "ground hog" bird'  
print, StrSplit(str, "", /Extract)
```

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---

---

Subject: Re: splitting strings

Posted by [Benjamin Hornberger](#) on Thu, 03 Feb 2005 20:53:42 GMT

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

---

David Fanning wrote:

> Benjamin Hornberger writes:  
>  
>  
>> I would like to split a string by whitespace characters, while anything  
>> between quotes should be recognized as one elements (even if it contains  
>> whitespace). Let's say I have the string  
>>  
>> 'cat dog "ground hog" bird'  
>>  
>> I want to split it into ['cat', 'dog', 'ground hog', 'bird']. Does  
>> anybody have a general algorithm or a function to do that? Or do I have  
>> to work it out myself?  
>  
>  
> str = 'cat dog "ground hog" bird'

> print, StrSplit(str, "", /Extract)  
>  
> Cheers,  
>  
> David

Doesn't work, the result is an empty string.

If you use " " as pattern for strsplit (the default), I get ['cat',  
'dog', '"ground', 'hog', 'bird'], which is not what I want either.

Benjamin

---

---

Subject: Re: splitting strings

Posted by [David Fanning](#) on Thu, 03 Feb 2005 21:04:50 GMT

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

---

Benjamin Hornberger writes:

> Doesn't work, the result is an empty string.  
>  
> If you use " " as pattern for strsplit (the default), I get ['cat',  
> 'dog', '"ground', 'hog', 'bird'], which is not what I want either.

Oh, right. I'm still faint from an exhilarating win at tennis today. :-)

Probably have to do a two-pass thing where you remove internal quotes.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

---

---

Subject: Re: splitting strings

Posted by [Benjamin Hornberger](#) on Fri, 04 Feb 2005 20:06:35 GMT

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

---

Benjamin Hornberger wrote:

> Hi all,  
>  
> I would like to split a string by whitespace characters, while anything  
> between quotes should be recognized as one elements (even if it contains

> whitespace). Let's say I have the string  
>  
> 'cat dog "ground hog" bird'  
>  
> I want to split it into ['cat', 'dog', 'ground hog', 'bird'].

Ok, here we go. Found on <http://php.net/split> and translated to IDL with some modifications. Any comments are welcome.

```
;+
; 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
;
;
;
; 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.
;
;
;
; INPUTS:
;   String: String to be split. Must be a scalar string or one
;   element string array.
;
;
;
; OPTIONAL INPUTS:
;   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.
```

#### ; SIDE EFFECTS:

```
; If the Delimiters argument is passed as string array, it will be
; joined by QUOTESPLIT.
```

#### ; 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:

```
; Written: BH 2005-02-04, translated to IDL with some modifications
; from http://php.net/split (User Contributed Note from "moritz").
```

```
; -
```

FUNCTION quotesplit, string, delimiters

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 array = "

return, array

```

END

---

---

Subject: Re: splitting strings  
Posted by [KM](#) on Sat, 05 Feb 2005 15:21:49 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 4 Feb 2005, Benjamin Hornberger wrote:

> Benjamin Hornberger wrote:  
>> Hi all,  
>>  
>> I would like to split a string by whitespace characters, while  
>> anything between quotes should be recognized as one elements  
>> (even if it contains whitespace). Let's say I have the string  
>>  
>> 'cat dog "ground hog" bird'  
>>  
>> I want to split it into ['cat', 'dog', 'ground hog', 'bird'].  
>  
> Ok, here we go. Found on <http://php.net/split> and translated to  
> IDL with some modifications. Any comments are welcome.

Looks like it works. It is loopy, but I doubt you need much speed, unless you have a lot of strings. That being said, I got this working before you posted that with your 2nd example. The 2nd example breaks the code, but works with some minor modifications. The following is not very robust.

```
str = 'cat dog "ground hog" "bird"'
str2 = strsplit(str,/extract) ; split on whitespace
split = STREGEX(str2,"") ; find all ""s
split_loc = where( split gt 0, n )-1 ; use only the 2st half
str3 = str2
str3[ split_loc ] = str3[split_loc]+" "+str3[split_loc+1] ; patch
str3[ split_loc+1 ] = "rem" ; remove the 2nd half of the bad split
str3 = str3[ where( str3 NE 'rem' ) ]
```

Also, unrelated to the above:

```
IDL> print, stregex(str,".*( ).*"/subexpr)
prints both the location of the " and the space.
```

So I'm pretty sure there is a way to do this without loops...

-k.

---