
Subject: Re: format integers with place separators

Posted by [Benjamin Hornberger](#) on Tue, 04 Oct 2005 15:42:00 GMT

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

Ben Tupper wrote:

>
> Is it possible to format an integer using the format specifiers such
> place separators are used? For example...
>
> 1234 becomes 1,234
>
> 1234567 becomes 1,234,567
>
> 123456789 becomes 123,456,789
>

I think it's not possible. I wrote my own function, see below. I used it only a couple times, hopefully it's too buggy.

Benjamin

```
;+
; NAME:
;   COMMAFORMAT
;
;
; PURPOSE:
;
;   This function takes a number and returns it as a string with a
;   comma every 10^3.
;
;
; AUTHOR:
;
;   Benjamin Hornberger
;   benjamin.hornberger@stonybrook.edu
;
;
; CATEGORY:
;
;   Utilities
;
;
; CALLING SEQUENCE:
;
;   Result = COMMAFORMAT(number, dec)
;
```

```

;
; INPUT PARAMETERS:
;
;   number: The number to be formatted
;
;   dec: number of decimals
;
;
; INPUT KEYWORDS:
;
;   INT: Set this keyword to return an integer without decimal
;        point.
;
;
; RESTRICTIONS:
;
;   Currently, the function just cuts off the number at the specified
;   decimal position, without performing any rounding.
;
;
; EXAMPLE:
;
;   IDL> print, commaformat(123456789.12)
;   123,456,789.12
;
;
; MODIFICATION HISTORY:
;
;   2004-05-27 BH: Written
;   2005-06-30 BH: Changed error handling. Added standard
;                   documentation header.
;-

```

FUNCTION commaformat, number, dec, int=int

on_error, 2

;; don't allow complex, strings, structures, pointers and object

;; references

IF size(number, /type) GT 5 AND size(number, /type) LT 12 THEN \$
 message, 'commaformat supports only integer and real numbers'

IF n_elements(dec) EQ 0 THEN dec = 0

IF keyword_set(int) THEN BEGIN

string = string(abs(number), format='(i0)')

dotpos = strlen(string)

ENDIF ELSE BEGIN

```

string = string(abs(number), format='(f0)')
dotpos = strpos(string, '.')
;; Cut off decimals. Unfortunately, the strmid function can't
;; take a separate 'First_Character' parameter for each array
;; entry.
FOR i=0, n_elements(string)-1 DO $
    string[i] = strmid(string[i], 0, dotpos[i]+1+dec[0])
ENDELSE

FOR i=0, n_elements(string)-1 DO BEGIN
    WHILE dotpos[i] GT 3 DO BEGIN
        string[i] = strmid(string[i], 0, dotpos[i]-3) + ',' + $
            strmid(string[i], dotpos[i]-3)
        dotpos[i] = dotpos[i]-3
    ENDWHILE
ENDFOR

negatives = where(number LT 0, count)
IF count GT 0 THEN $
    string[negatives] = '-' + string[negatives]

return, string

END

```
