Subject: Re: printing floats/integer
Posted by Klaus Scipal on Thu, 08 Mar 2001 13:57:00 GMT
View Forum Message <> Reply to Message

you can use the the string command in connection with format

for example
IDL> x=1.222221000000000000
IDL> print, string(format='(f8.6)',x)
 1.222221

klaus

Sean Heukels <sean77=cuthere=@dds.nl> wrote in message
news:9881p0$55o$1@newshost.accu.uu.nl...
> I wrote a small module for integers. The variable is formatted and
returned.
>
> "1 " returns as "1"
>
> Now this doesn't work with floats. for example if I want to print
"1.22221"
> I
> dont want to see it as "1.222221000000000000" or
> "1.222222 "
>
> Does anyone know how I can solve this ??
>
> Greets Sean
>
>

Subject: Re: printing floats/integer
Posted by Sean Heukels on Thu, 08 Mar 2001 14:21:08 GMT
View Forum Message <> Reply to Message

What if I dont know that the float will
be 8 in length and has 6 decimals ??

How can we put this in a subroutine, that handles
a variable (e.g. PRO FORMAT, var1) and returns
the right way to print ??

For the int I use

PRO FORMAT, var
which=SIZE(var, /TYPE)

if (which eq 2) then begin
le=FLOOR(ALOG10(var)) + 1
RETURN, STRING(var, FORMAT='(I'+STRTRIM(STRING(le))+')'
endif else if (which eq 4) then begin

How do I do this for a float >>>??
1.22222200000 must become 1.22222 without leading or trailing spaces.
And in the best scenario 1.00 doesn;t become 1., but 1.0 ....

Sean

Klaus Scipal <kscipal@ipf.tuwien.ac.at> schreef in berichtnieuws
9882vbsbi1@news.tuwien.ac.at...
> you can use the the string command in connection with format
>
> for example
> IDL> x=1.222221000000000000
> IDL> print, string(format='(f8.6)',x)
> 1.222221
>
> klaus
>
> Sean Heukels <sean77=cuthere=@dds.nl> wrote in message
> news:9881p0$55o$1@newshost.accu.uu.nl...
>> I wrote a small module for integers. The variable is formatted and
> returned.
>>
>> "1 " returns as "1"
>>
>> Now this doesn't work with floats. for example if I want to print
> "1.22221"
>> I
>> dont want to see it as "1.222221000000000000" or
>> "1.222222 "
>>
>> Does anyone know how I can solve this ??
>>
>> Greets Sean
>>
>>
>
>

Subject: Re: printing floats/integer
Posted by Paul van Delst on Thu, 08 Mar 2001 15:16:34 GMT
View Forum Message <> Reply to Message

Sean Heukels wrote:
>
> What if I dont know that the float will
> be 8 in length and has 6 decimals ??

You, as the user of the number, have to specify something about what you want for the output format. E.g. maximum length, how many decimal places, etc. IDL, and computers in general, have absolutely no intelligence - that's where the user comes in. :o) If you don't know what the size of the number will be, use exponential format.

> How can we put this in a subroutine, that handles
> a variable (e.g. PRO FORMAT, var1) and returns
> the right way to print ??

You haven't defined what "right" is in a way a program can understand it.

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
paul.vandelst@noaa.gov Alexander Pope.

Subject: Re: printing floats/integer
Posted by Klaus Scipal on Thu, 08 Mar 2001 16:00:39 GMT
View Forum Message <> Reply to Message

the only solution to your problem I would come up with is doing it in a loop recursively but i am afraid this is not very elegant

klaus

Sean Heukels <sean77=cuthere=@dds.nl> wrote in message news:9884hbcoo1@newshost.accu.uu.nl...
>
> What if I dont know that the float will
> be 8 in length and has 6 decimals ??
>
> How can we put this in a subroutine, that handles
> a variable (e.g. PRO FORMAT, var1) and returns

> the right way to print ??
>
> For the int I use
>
> PRO FORMAT, var
> which=SIZE(var, /TYPE)
> if (which eq 2) then begin
> le=FLOOR(ALOG10(var)) + 1
> RETURN, STRING(var, FORMAT='(I'+STRTRIM(STRING(le))+')'
> endif else if (which eq 4) then begin
>
> How do I do this for a float >>>??
> 1.22222200000 must become 1.22222 without leading or trailing spaces.
> And in the best scenario 1.00 doesn;t become 1., but 1.0 ....
>
>
>
> Sean
>
>
> Klaus Scipal <kscipal@ipf.tuwien.ac.at> schreef in berichtnieuws
> 9882vbsbi1@news.tuwien.ac.at...
>> you can use the the string command in connection with format
>>
>> for example
>> IDL> x=1.222210000000000000
>> IDL> print, string(format='(f8.6)',x)
>> 1.222221
>>
>> klaus
>>
>> Sean Heukels <sean77=cuthere=@dds.nl> wrote in message
>> news:9881p0$55o$1@newshost.accu.uu.nl...
>>> I wrote a small module for integers. The variable is formatted and
>> returned.
>>>
>>> "1 " returns as "1"
>>>
>>> Now this doesn't work with floats. for example if I want to print
>> "1.22221"
>>> I
>>> dont want to see it as "1.222210000000000000" or
>>> "1.222222 "
>>>
>>> Does anyone know how I can solve this ??
>>>
>>> Greets Sean
>>>

>>>
>>
>>
>
>

Subject: Re: printing floats/integer
Posted by Christopher W. O'Dell on Thu, 08 Mar 2001 17:01:20 GMT
View Forum Message <> Reply to Message

This problem has been around about as long as computers I think. I used to have an old pascal routine that would tell you what format code you needed for a given number, assuming you wanted to see the whole number, but not anymore more than is necessary (for instance, 8.6). Has anyone written such a function for IDL?

You could probably piece together what you need using this function I just found on the IDL Libraries Browser at Washington (see below).

Good luck,
Chris

```
 ;------------------------------------------------------------ -------
                function sigfig, range
;
;+
; NAME:
;       SIGFIG
;
; PURPOSE:
;       This function will return the number of significant figures in
;       the value "range"
;
; CATEGORY:
;       utilities
;
; CALLING SEQUENCE:
;       result = sigfig(range)

; INPUTS:
;       range = range for the significance, may be an array.
;
; OUTPUTS:
;       result = the number of significant figures expressed in base 10,
;                Example: sigfig(1000) = 3,
;                         sigfig(0.01) = -2
;
; COMMON BLOCKS:
```

;	none.
; SIDE EFFECTS:
;	none.
; MODIFICATION HISTORY:
;	Written by: Trevor Harris, Physics Dept., University of Adelaide,
;		July, 1990.

Subject: Re: printing floats/integer
Posted by R.G.S. on Thu, 08 Mar 2001 17:27:35 GMT
View Forum Message <> Reply to Message

Sean Heukels <sean77=cuthere=@dds.nl> wrote in message
news:9881p0$55o$1@newshost.accu.uu.nl...
> I wrote a small module for integers. The variable is formatted and
returned.
>
> "1 " returns as "1"
>
> Now this doesn't work with floats. for example if I want to print
"1.22221"
> I
> dont want to see it as "1.222221000000000000000" or
> "1.222222 "
>
> Does anyone know how I can solve this ??
>
> Greets Sean

Please see the following code.
Note that there is a main level example code at the bottom of the function.

Using a = 123121.22222200000d
I get the following output:
Normal print command:
	123121.22
Print command with GetFormat:
123121.222222

Cheers,
bob stockwell
stockwell@co-ra.com

```
function getformatstring,number

; this should be big enough to show everything
showallformatcommmand = '(f50.25)'

nsize = size(number,/st)
case nsize.type of
;0:  Undefined
;1:  Byte
;2:  Integer
;3:  Longword integer
4: begin;  Floating point
 s =  STRING(FORMAT=showallformatcommmand, number)
 s = strtrim(s,1) ; remove leading blanks
 slen = strlen(s)
 dpos=strpos(s,'.') ; find decimal posiiton
 if dpos eq -1 then begin
  ; this shouldn't happen
  print,'Error: no decimal place found'
  formatstring = ''
 endif else begin
  ; chop off lagging 0s
  zpos=0
  doflag=1
  while doflag do begin ; find how many lagging zeros there are
   char = strmid(s,zpos,1,/reverse)
   if char eq '0' then begin
    zpos=zpos+1
   endif else begin
    doflag=0
   endelse
  endwhile
 endelse
 end
5:  begin;  Double-precision floating
 s =  STRING(FORMAT=showallformatcommmand, number)
 s = strtrim(s,1) ; remove leading blanks
 slen = strlen(s)
 print,'slen:',slen
 print,s
 dpos=strpos(s,'.') ; find decimal posiiton
 if dpos eq -1 then begin
  ; this shouldn't happen
  print,'Error: no decimal place found'
  formatstring = ''
 endif else begin
  ; chop off lagging 0s
  zpos=0
```

```
  doflag=1
  while doflag do begin ; find how many lagging zeros there are
   char = strmid(s,zpos,1,/reverse)
   if char eq '0' then begin
    zpos=zpos+1
   endif else begin
    doflag=0
   endelse
  endwhile
 endelse
 end
;6:; Complex floating
;7:  String
;8:  Structure
;9:; Double-precision complex
;10: Pointer
;11: Object reference
;12:; Unsigned Integer
;13:; Unsigned Longword Integer
;14:; 64-bit Integer
;15:; Unsigned 64-bit Integer
else: begin
 print,'Error: Number not a supported type.'
 return,'' ; null string
endelse
endcase

d =slen-dpos-zpos-1
w = dpos+d+1 ; 1 for the decimal place
formatstring = strcompress("(f"+string(w)+'.'+string(d)+")",/rem)
return,formatstring

end




; ******    main level code here
a = 123121.22222200000d
;must become 1.22222 without leading or trailing spaces.
b=  1.00

print,'Normal print command:'
print,a
print,'Print command with GetFormat:'
print,a,format=getformatstring(a)
```

end

Subject: Re: printing floats/integer
Posted by Craig Markwardt on Thu, 08 Mar 2001 22:44:17 GMT
View Forum Message <> Reply to Message

"Sean Heukels" <sean77=cuthere=@dds.nl> writes:
> I wrote a small module for integers. The variable is formatted and returned.
>
> "1 " returns as "1"
>
> Now this doesn't work with floats. for example if I want to print "1.22221"
> I
> dont want to see it as "1.222221000000000000000" or
> "1.222222 "
>
> Does anyone know how I can solve this ??

The first thing to understand is that there are many real numbers which cannot be represented in floating point number system of computers. That's just the facts.

Second, if you simply don't want spaces around your numbers, why not try STRTRIM?

One of the most sophisticated answers might be found in the following paper by Burger and Dybvig, "Printing Floating-Point Numbers Quickly and Accurately:"

http://citeseer.nj.nec.com/28233.html

Unfortunately that's probably overkill, and too hard to implement in IDL.

You've seen Bob S's implementation. I have a program called INPUTFORM on my web page which prints a number as a string, such that it can be read again by the IDL parser. Like this:

IDL> print, inputform(1.22221)
1.22221E

You see the "E" indicates that it is a single-precision floating point number, to distinguish it unambiguously from double precision or integers. Just as Bob's code is not too pretty (sorry Bob), neither is the code in INPUTFORM. Basically it tries printing the number with both the "G" and "D" output formats and takes whichever is shorter.

Craig

INPUTFORM can be found at
http://cow.physics.wisc.edu/~craigm/idl/idl.html

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------