
Subject: Test if variable is a string
Posted by [Carsten Dominik](#) on Tue, 04 Feb 2003 09:56:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I must be blind: How do I test for the type of the object contained in a variable? I am writing a procedure and want to allow one of the arguments to be either a string or a number - but the program must of course find out which it is?

This is my main problem with IDL: I am **never** able to correctly guess the name of a function for a particular purpose.

So what is the magic word for determining the type of a variable?
Thanks.

- Carsten

Subject: Re: Test if variable is a string
Posted by [R.Bauer](#) on Sat, 15 Feb 2003 13:28:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Carsten Dominik wrote:

> Hello,
>
> I must be blind: How do I test for the type of the object contained
> in a variable? I am writing a procedure and want to allow one of the
> arguments to be either a string or a number - but the program must of
> course find out which it is?
>
> This is my main problem with IDL: I am **never** able to correctly guess
> the name of a function for a particular purpose.
>
> So what is the magic word for determining the type of a variable?
> Thanks.
>
> - Carsten
>
>

Dear Carsten,

a bit late but I believe the opposite check is better to test if the string is a regular number because if you test only about types it could happen that's an exponent number is defined as string.

I often use a routine named `string_is_number`

you'll find it here:

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl_html/dbase/download/string_is_number.tar.gz

regards

reimar

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

a IDL library at ForschungsZentrum Juelich

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html

Subject: Re: Test if variable is a string
Posted by [MKatz843](#) on Sun, 16 Feb 2003 20:58:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Here's a handy little function I've been using for years.

```
var_type.pro
```

I think I added a number of new type names back in 2001.
If RSI adds more types in the future, they'll probably
be appended to the list rather than change the existing list.
In that case, you'll have to modify the list.

So the function returns the type as a number. But if
you set the `/string` keyword, you'll get a string containing
the name of the type.

For example:

```
IDL> print, var_type(10L)
```

```
3
```

```
IDL> print, var_type(10L, /s)
Longword Integer
```

```
;+
; IDL function: var_type.pro
; for a given input variable, this function returns
; the type of that variable. The returned value is a number 0-15
; similar to the second-to-last field in the "size" function,
; or if the keyword "string" is used, a string.
;-
```

```
function var_type, a, string=string
```

```
    s = ['Undefined','Byte','Integer','Longword
Integer','Floating-point', $
        'Double-precision floating','Complex
floating','String','Structure', $
        'Double-precision complex','Pointer','Object reference', $
        'Unsigned Integer','Unsigned Longword Integer','64-bit
Integer',$
        'Unsigned 64-bit Integer']
```

```
return, Keyword_set(string) ? s(size(a, /type)):size(a, /type)
end
```

I also have a closely related function called `defined.pro` that tests true-or-false, if a given variable is defined or not. It also uses `size(x, /type)`, that's why I'm including it here.

```
;+
; IDL function: defined.pro
; For a given input variable, this function returns
; 1b if the variable is defined
; 0b if the variable is undefined
;-
```

```
function defined, a
return, (size(a, /type) NE 0)
end
```

Hope this helps,

M. Katz

Subject: Re: Test if variable is a string

M. Katz wrote:

Did you recognize the tname keyword of size?

```
IDL> print,size(10L,/tname)
LONG
```

regards
Reimar

```
> Here's a handy little function I've been using for years.
> var_type.pro
> I think I added a number of new type names back in 2001.
> If RSI adds more types in the future, they'll probably
> be appended to the list rather than change the existing list.
> In that case, you'll have to modify the list.
>
> So the function returns the type as a number. But if
> you set the /string keyword, you'll get a string containing
> the name of the type.
>
> For example:
> IDL> print, var_type(10L)
>      3
> IDL> print, var_type(10L, /s)
> Longword Integer
>
> ;+
> ; IDL function: var_type.pro
> ; for a given input variable, this function returns
> ; the type of that variable. The returned value is a number 0-15
> ; similar to the second-to-last field in the "size" function,
> ; or if the keyword "string" is used, a string.
> ;-
>
> function var_type, a, string=string
>
>   s = ['Undefined','Byte','Integer','Longword
> Integer','Floating-point', $
>       'Double-precision floating','Complex
> floating','String','Structure', $
>       'Double-precision complex','Pointer','Object reference', $
>       'Unsigned Integer','Unsigned Longword Integer','64-bit
> Integer',$
>       'Unsigned 64-bit Integer']
```

```

>
> return, Keyword_set(string) ? s(size(a, /type)):size(a, /type)
> end
>
> I also have a closely related function called defined.pro that tests
> true-or-false, if a given variable is defined or not. It also uses
> size(x, /type), that's why I'm including it here.
>
> ;+
> ; IDL function: defined.pro
> ; For a given input variable, this function returns
> ; 1b if the variable is defined
> ; 0b if the variable is undefined
> ; -
>
> function defined, a
> return, (size(a, /type) NE 0)
> end
>
>
> Hope this helps,
>
> M. Katz

```

--

Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
 Forschungszentrum Juelich
 email: R.Bauer@fz-juelich.de

a IDL library at ForschungsZentrum Juelich

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html

=====
