
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
