

str2num

```
timerRefNum = StartMSTimer
if (timerRefNum == -1)
    Abort "All timers are in use"
endif
n=10000
do
    n -= 1
while (n > 0)
microSeconds = StopMSTimer(timerRefNum)
Print microSeconds/10000, "microseconds per iteration"
End
```

See Also

StartMSTimer, **ticks**, **DateTime**

str2num

str2num(*str*)

The **str2num** function returns a number represented by the string expression *str*.

Details

str2num returns NaN if *str* does not contain the text for a number.

str2num skips leading spaces and tabs and then reads up to the first non-numeric character.

See Also

The **char2num**, **num2char** and **num2str** functions.

The **sscanf** operation for more complex parsing jobs.

StrConstant

StrConstant *ksName*="literal string"

The **StrConstant** declaration defines the string *literal string* under the name *ksName* for use by other code, such as in a switch construct.

See Also

The **Constant** keyword for numeric types, **Constants** on page IV-51, and **Switch Statements** on page IV-43.

String

String [/G] *strName* [/N=*name*] [=*strExpr*] [, *strName* [/N=*name*] [=*strExpr*] ...]

The **String** operation creates string variables and gives them the specified names.

Flags

- | | |
|-----------------|---|
| /G | Creates a global string. Overwrites any existing string with the same name. |
| /N= <i>name</i> | Specifies a local name for the global string variable. /N was added in Igor Pro 8.00 and is available in user-defined functions only. See SVAR Creation below for details. |

Details

The string variable is initialized when it is created if you supply the *=strExpr* initializer. However, when **String** is used to declare a function parameter, it is an error to attempt to initialize it.

You can create more than one string variable at a time by separating the names and optional initializers with commas.

If used in a procedure, the new string is local to that procedure unless the /G flag is used. If used on the command line, **String** is equivalent to **String/G**.

strName can optionally include a data folder path.

SVAR Creation

In a user-defined function, you need a local SVAR reference to access a global string variable. If you use a simple name rather than a path, Igor automatically creates an SVAR:

```
String/G sVar1 // Creates an SVAR named sVar1
```