URL: http://cxc.harvard.edu/sherpa3.4/faq/slangcomm_interp.html
Last modified: 21 September 2006

# How can I control whether *Sherpa* or S−Lang interprets a command?

*Sherpa* can interpret both native *Sherpa*/*ChIPS* commands and S−Lang statements. Although differences in syntax are generally sufficient to distinguish the two, ambiguous input is possible.

In the example below, the user tries to define a source model *and* a S−Lang variable "src", using the syntax `src = ...` in both cases. The syntactic ambiguity causes problems:

```
sherpa> src = const1d[c]   # Define a source model
sherpa> show src
c
const1d[c]  (integrate: on)
    Param   Type      Value        Min        Max                Units
    -----   ----      -----        ---        ---                -----
 1     c0 thawed          1          0 3.4028e+38
sherpa> src = "NGC 1068"   # Define the S-Lang variable "src"
sherpa> print(src)
NGC 1068
sherpa> show src
Model not defined.
sherpa> # S-Lang assignment erased the source model definition
```

Clearly, the best solution in this case is to choose a different name for the S−Lang variable. However, it is possible to force an input line to be interpreted as *only* a *Sherpa* command by using the `sherpa_eval` function or *only* a S−Lang statement by using the `eval` function.

The above example can be corrected by using `eval` to create the "src" variable:

```
sherpa> src = const1d[c]
sherpa> eval("variable src = \"NGC 1068\"")
sherpa> show src
c
const1d[c]  (integrate: on)
    Param   Type      Value        Min        Max                Units
    -----   ----      -----        ---        ---                -----
 1     c0 thawed          1          0 3.4028e+38
sherpa> print(src)
NGC 1068
```



URL: http://cxc.harvard.edu/sherpa3.4/faq/slangcomm_interp.html
Last modified: 21 September 2006