Subject: Re: "foreach" loops in IDL
Posted by rtk on Tue, 20 Jan 2009 22:54:39 GMT
View Forum Message <> Reply to Message

On Jan 20, 2:11 pm, JD Smith <jdtsmith.nos...@yahoo.com> wrote:
> foreach elem in x
> print, my_function(elem)
> end
>
> When treating a vector as a list, requiring an extra loop variable is
> pure syntactic overhead. Not to mention that the loop variable could
> overflow, could get changed in the body of the loop or, most commonly,
> risks nested sub-loops accidentally re-using the same loop variable.
> None of these happens with a foreach construct.

Exactly! Of course, at this point, we are talking Python:

```
for elem in x:
    print my_function(elem)
```

Lists are a natural extension/complement to IDL's arrays. Then, once you have lists, you will naturally want higher-order functions to operate on those lists. Lists would add great power to IDL but perhaps if people are not familiar with lists they will not appreciate what lists have to offer. To prove my point, look at what happened with Python. It supports lists out of the box but folks added powerful array processing via the numpy package. Well, IDL is just the opposite. It has powerful array processing already, now it just needs lists.

The package I referenced above has a list DLM that is quite useful on its own. It supports all IDL data types except structures. A truly integrated version would naturally add helpful syntax and support all data types.

Ron
(Disclaimer: while I do work for ITT VIS, these opinions are my own and should not be thought of as those of ITT VIS...)