
Subject: Re: Optional parameters

Posted by [Craig Markwardt](#) on Wed, 25 Feb 2004 05:50:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

JD Smith <jdsmith@as.arizona.edu> writes:

```
> I'd offer another real life example where read-write keyword variables
> can get you into trouble. When you call a function several times in a
> row, expecting the same results each time, but in actuality the
> keyword variable is defined after the first invocation, which can
> change the function's behavior:
>
> a=myfunc(test,FUNNY_VARIABLE=foo) ; foo undefined
> a=myfunc(test,FUNNY_VARIABLE=foo) ; foo defined
> a=myfunc(test,FUNNY_VARIABLE=foo) ; foo defined
```

Yes, I've encountered this kind of problem before as well, and it's one of the most evil and difficult problems to solve.

That's why I define all of my functions as

```
pro myfunc, value=value0

;; Default value ...
if n_elements(value0) EQ 0 then value = 1 $
;; ... or requested value
else          value = round(value0(0))
...
end
```

The "0" variable is the pristine variable that never gets changed, and the regular variable is the one that is internal to the procedure. I do that as a matter of habit now and it's not too labor intensive.

I do kind of wish that there would be an automatic way for IDL to peel off a local copy of a variable for internal calculations.

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
