
Subject: Re: Error Handling Advice

Posted by [Paul Van Delst\[1\]](#) on Mon, 14 Jan 2013 16:51:57 GMT

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

hello,

On 01/14/13 10:05, David Fanning wrote:

> John Correia writes:

>

>> Is there a technical reason behind not making a keyword a required
>> argument, or just a personal preference based on experience? I find it
>> harder to interpret a long series of positional parameters as opposed to
>> a list of somewhat descriptive keywords.

>

> It is not a technical reason so much as a visceral response. :-)

Ha! Good answer!

FWIW I apply the IDL parameter/keyword type for my Fortran95/2003 code.

E.g. if I have

```
subroutine mysub(a,b,c,d)
  real      :: a, b
  real, optional :: c, d
  ...
end subroutine mysub
```

the documentation for this routine's calling sequence is always listed
sort-of like an IDL routine:

```
! call mysub(a, b, c=c, d=d)
```

to indicate that "a" and "b" are mandatory arguments, and "c" and "d"
are optional.

> I suppose I write programs differently than most people, but I've never
> had a need to define more than three required arguments in a program
> EVER, as far as I can recall. I'm sure there is a reason for the 10-12 I
> sometimes see in programs I haven't written, but I can't imagine what
> that reason is. People who come to me with those kinds of programs are
> always confused. I just assumed that was one of the reasons why. ;-)

My rule of thumb: More than 3-5 required arguments -> collect them in a
structure! :o)

cheers,

paulv
