Subject: Anonymous Math Functions in IDL like Matlab
Posted by Ammar Yusuf on Thu, 24 Feb 2011 01:47:24 GMT
View Forum Message <> Reply to Message

Hi, in Matlab you can do something like:
sqrt = @(x) x.^2;

Is there a way to do this in IDL? I was just wondering.

I tried searching google but that didn't help. Thanks!

Subject: Re: Anonymous Math Functions in IDL like Matlab
Posted by Michael Galloy on Fri, 25 Feb 2011 16:16:50 GMT
View Forum Message <> Reply to Message

On 2/24/11 8:32 pm, nata wrote:
> I am just curious. What does the expression "sqrt = @(x) x.^2" means ?

It's a lambda function, or anonymous function, a way of specifying a
function without naming it, i.e., to use inline only once.

For example, see how Python does this:

http://diveintopython.org/power_of_introspection/lambda_func tions.html

Mike
--
www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: Anonymous Math Functions in IDL like Matlab
Posted by Ammar Yusuf on Sat, 26 Feb 2011 00:50:11 GMT
View Forum Message <> Reply to Message

On Feb 25, 11:16 am, Michael Galloy <mgal...@gmail.com> wrote:
> On 2/24/11 8:32 pm, nata wrote:
>
>> I am just curious. What does the expression "sqrt = @(x) x.^2" means ?
>
> It's a lambda function, or anonymous function, a way of specifying a
> function without naming it, i.e., to use inline only once.
>
> For example, see how Python does this:
>

> http://diveintopython.org/power_of_introspection/lambda_func tions.html
>
> Mike
> --www.michaelgalloy.com
> Research Mathematician
> Tech-X Corporation

That's pretty cool. I did not know that.

I know in Matlab what happens is it creates a function say sqrt. In Matlab
you can then call sqrt(4) and it would return 2.

I found this very helpful instead of making a .m file for each polynomial function
that I was doing for my homework.

Subject: Re: Anonymous Math Functions in IDL like Matlab
Posted by rtk on Thu, 03 Mar 2011 23:13:33 GMT
View Forum Message <> Reply to Message

I wrote a bunch of higher-order function stuff for IDL awhile back and included a hack that simulated a lambda function. Basically, it defined the function on the fly and returned a string with the actual name. You then used this with the higher-order function (or any function). So, for your example,

square = lambda('x:x^2')

but to use it directly:

ans = call_function(square, 4)

Outside the context of a higher-order function, this isn't really that useful. Still, if you want the code, I'll send it, just email me "oneelkruns" "hotmail" "com". The higher-order functions were DLMs for Linux and Windows, 32-bit only.

Also, you can define a procedure/function in IDL from the command line. No need to quit and type a new file:

```
IDL> .run
- function square, x
-   return, x*x
- end
% Compiled module: SQUARE.
IDL> print, square(4)
```

Perhaps what you are really looking for?

Ron