

## Re: Python's simplicity philosophy

**Source:** <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2003-11/2618.html>

---

**From:** Alex Martelli (*aleax\_at\_aleax.it*)

**Date:** 11/14/03

Date: Fri, 14 Nov 2003 10:08:54 GMT

Terry Reedy wrote:

...  
>> *personally argued that reduce() should be removed from the language,*  
>> *but I do agree that it does not have to be part of "core" Python,*  
>> *and could easily be relegated to a module.)*  
>  
> *If the builtins are reduced in 3.0, as I generally would like, I would*  
> *be fine with moving apply, map, filter, and a repaired version of*  
> *reduce to a 'functional or hof module. But the argument of some*  
> *seems to be that this batteries-included language should specifically*  
> *exclude even that.*

A functional module would be neat. A great way to enhance the chance that there will be one would be starting one today (e.g. on sourceforge), ideally with both pure-Python and C-helped (or pyrex, etc) implementations, and get it reasonably-widely used, debugged, optimized. There's plenty of such ideas around, but gathering a group of people particularly keen and knowledgeable about functional programming and hashing out the "best" design for such a module would seem to be a productive endeavour.

Also, I advocate that 3.0 should have a module or package (call it "legacy" for example) such that, if you started your program with some statement such as:

```
from legacy import *
```

compatibility with Python 2.4 or thereabouts would be repaired as much as feasible, to ease running legacy code, and to the expense of performance, 'neatness' and all such other great things if needed (e.g., no "repaired" versions or anything -- just compatibility).

One reasonably popular counterproposal to that was to have it as

```
from __past__ import *
```

by analogy with today's "from \_\_future\_\_". I'd also like to make it easy to get this functionality with a commandline switch, like is the case today with -Q specifically for `_division_` legacy issues.

But mostly, each time I mention that on python-dev, I'm rebuffed with remarks about such issues being way premature today. Somebody's proposed starting a new list specifically about 3.0, to make sure remarks and suggestions for it made today are not lost about more day-to-day python-dev traffic, but I don't think anything's been done about that yet.

Alex