(continued from previous page)

```
total = functools.reduce(combine, items)[1]
```

But it would be best of all if I had simply used a `for` loop:

```
total = 0
for a, b in items:
    total += b
```

Or the `sum()` built-in and a generator expression:

```
total = sum(b for a, b in items)
```

Many uses of `functools.reduce()` are clearer when written as `for` loops.

Fredrik Lundh once suggested the following set of rules for refactoring uses of `lambda`:

1. Write a lambda function.
2. Write a comment explaining what the heck that lambda does.
3. Study the comment for a while, and think of a name that captures the essence of the comment.
4. Convert the lambda to a def statement, using that name.
5. Remove the comment.

I really like these rules, but you're free to disagree about whether this lambda-free style is better.

9 Revision History and Acknowledgements

The author would like to thank the following people for offering suggestions, corrections and assistance with various drafts of this article: Ian Bicking, Nick Coghlan, Nick Efford, Raymond Hettinger, Jim Jewett, Mike Krell, Leandro Lameiro, Jussi Salmela, Collin Winter, Blake Winton.

Version 0.1: posted June 30 2006.

Version 0.11: posted July 1 2006. Typo fixes.

Version 0.2: posted July 10 2006. Merged genexp and listcomp sections into one. Typo fixes.

Version 0.21: Added more references suggested on the tutor mailing list.

Version 0.30: Adds a section on the `functional` module written by Collin Winter; adds short section on the operator module; a few other edits.

10 References

10.1 General

Structure and Interpretation of Computer Programs, by Harold Abelson and Gerald Jay Sussman with Julie Sussman. Full text at https://mitpress.mit.edu/sicp/. In this classic textbook of computer science, chapters 2 and 3 discuss the use of sequences and streams to organize the data flow inside a program. The book uses Scheme for its examples, but many of the design approaches described in these chapters are applicable to functional-style Python code.

http://www.defmacro.org/ramblings/fp.html: A general introduction to functional programming that uses Java examples and has a lengthy historical introduction.

https://en.wikipedia.org/wiki/Functional_programming: General Wikipedia entry describing functional programming.