Python Regrets

OSCON, July 25, 2002

Guido van Rossum
Director of PythonLabs at Zope Corporation

guido@zope.com
guido@python.org
Relics

- Stuff that’s already being phased out
- string exceptions
- sys.exc_type etc. (use sys.exc_info())
- int/int returning int
- apply() (use f(*args, **kwds))
- coerce() (no longer needed)
- 3-way compare? (but... comparing lists)
Lexical details

- continued lines or strings with \\
  - use (...) continuation and/or string literal concatenation

- if expression: statement
  - put statement on next line

- tabs?
  - require all spaces
  - or very restricted use of tabs (not mixed)
Lambda and functional stuff

- I've never liked lambda
  - crippled (only one expression)
  - confusing (no argument list parentheses)
  - can use a local function instead

- map(), filter()
  - using a Python function here is slow
  - list comprehensions do the same thing better

- reduce()
  - nobody uses it, few understand it
  - a for loop is clearer & (usually) faster
print, str() and repr()

- drop `\'x\'` for repr(x)
  - `\'` is hard to read in many fonts
  - publication process turns 's' into `\'s`'

- do we really need both str() and repr()?
  - mostly intended to be able to special-case "print x" when x is a string
  - still not enough; need nice() that's a hybrid

- print should've been a function
  - write(x, y, z)
  - writeln(x, y, z)
  - spaces between items controlled by keyword arg
• `intern()`, `id()`: put in `sys`
• `xrange()`: make `range()` return an iterator
• `buffer()`: must die (use `bytes`, PEP 296)
• `raw_input()`: use `sys.stdin.readline()`
• `input()`: use `eval(sys.stdin.readline())`
• `callable()`: just call it, already
• `execfile()`, `reload()`: use `exec()`
• `compile()`: put in `sys`
Special cases for exec

• exec as a statement is not worth it
  – make it a function (again :-)

• perhaps shouldn't have locals(), globals(), vars()
Restricted execution

• Too many bugs to be trusted
  – didn't get enough review, and never will
• Confusion btw. `__builtins__`, `__builtin__`

• But there's a useful idea somewhere...
  – Zope uses this
Float to int conversion

• need more ways to convert float to int (round, truncate-towards-zero, floor, ceil)
• need differentiate to `__int__` which truncates and `__int__` which doesn't