

PSIL -Python S-expression Intermediate Language

Greg Hewgill

<http://hewgill.com>

<http://github.com/ghewgill/psil>

What is PSIL?

- PSIL is Lisp embedded in Python
- Why Lisp? Lisp has been around for nearly 50 years and will probably be around for 50 more
- The Java people have Clojure – what do we have?
- PSIL offers Lisp with a Python flavour

PSIL Manifesto

- Best features from Lisp and Scheme
- Complete language in its own right
- Built upon the Python standard libraries
- Strong interoperability with Python code

Best features from Lisp and Scheme

- Some consider Lisp to embody the ultimate in programming language expressiveness
- Powerful macro facility provides a foundation for metaprogramming
- Opportunity to move away from historical names (setf? nconc?) while retaining the essence of Lisp such as cons and car.
- Functional focus of Scheme reflects current trend toward functional languages and immutable data structures

Complete language in its own right

- Based on Lisp, PSIL can be a complete language in its own right
- Offers functions, classes, I/O, etc
- Could use PSIL for teaching without necessarily mentioning Python

Built upon Python standard libraries

- Python's “batteries included” standard library is the best of its class
- Familiarity with the standard library allows programmers to get their job done more quickly
- Ease the transition for Python programmers interested in Lisp

Strong interoperability with Python code

- PSIL can call any Python function
- PSIL code can be embedded in Python code
- Python “callable” objects can be created within PSIL and called from Python
- Classes and modules can be implemented in PSIL

Example: PSIL calling Python

```
> (import os)
> (get os name)
posix

> (.system os "pwd")
/home/greg/src/psil
```


Example: Python calling PSIL

```
>>> from psil import psil
>>> sq = psil("(lambda (x) (* x x))")
>>> sq(5)
25
>>> [sq(x) for x in range(5)]
[0, 1, 4, 9, 16]
```