

A Zen for Scheme

In all life there must be balance. Python has The Zen of Python aka

```
import this
```

It is a powerful unifier that helps people graft libraries that feel pythonic. I try to write something in the same spirit (but not in [the same form](#)) for [Scheme](#).

Birds Eye

RR Remove limitations to Reduce the number of features you require,
*but **OM**: *Optimizability Matters*.*

FI Freedom for the Implementation and from the Implementation,
*but **CM**: *Community Matters: Join the Scheme you choose*.*

SL Mind the Small systems!
And the Large systems!

ES Errors should never pass silently,
*unless speed is set higher than safety.*¹

¹Thanks for the error-handling principle goes to John Cowans [Satire in the same form as Python](#).

On the Ground

HA Hygiene reduces Anxiety,
except where it blocks your path.

PP Practicality beats Purity,
except where it leads you into a dead end.

3P 3 Pillars of improvement:
Experimentation, Implementation, Standardization.

Submerged in Code

WM Use the Weakest Method that gets the job done,
but know the stronger methods to employ them as needed.

RW Recursion Wins,
except where a loop-macro is better.

RM Readability matters,
but nesting works.