

Functional Programming for no Rhyme or Reason

- Venkatesh Choppella

Programs are Expressions,
Expressions are reducible.
In the crucible of the Interpreter
the Meaning is deducible.

Syntax is superficial.
Pick one
that's easy 'n sensible.
While you're at it,
it may, for good measure,
be extensible.

Values are what you have,
when Expressions are normalizable.
Identifiable and immutable.
Plain and irreducible,
Their existence, irrefutable.

Semantics is simplification,
compositional and substitutional.
Depending on what you like,
axiomatic, operational or denotational.

Bindings are persistent,
and lexical.
Those recursive? Circular,
and paradoxical.

For, circularity and self-application,
you conclude, after some meditation,
are at the heart of
Recursive Functions that are Nameless.
and except when baseless (or tasteless),
are begotten, you realise somewhat later
from Y, the mysterious
Fixed Point Combinator.

Referential transparency,
is not about efficiency.
Young Man!
'Tis a Call to Reason!
Unlike Assignment,
which is forbidden,
and tantamount to treason!

Still, if Effects is what you might,
Try this thing called Monadic.
I'm told they're tricky to get right,
it's best when they are sporadic.

Functions are values,
applicable, and higher-order.
It's *what* lets you write
programs that are *shorter*.

Functions have Types.
Types are polymorphic.
And isomorphic,
to propositions,
that are logical.
Programs are like proofs,
of theorems mathematical.

The moral of this allegory:
Lambda is a Category,
that is Closed and Cartesian.
If you don't believe me,
ask a logician,
or a type theoretician.

Iteration is Recursion.
Recursion in tail position.
Recursion is Iteration,
just pass the Continuation!

Be warned! In colling with
the current continuation,
exercise caution,
lest there is confusion,
or a catastrophic situation.

An Object is a Closure.
(I learned this from a Hoosier.)
Formals, body and bindings of
free variables that are mutable.
Methods are functions over Self,
in a position that's suitable.

Data is algebraic, and inductive.
Processes, co-algebraic, and co-inductive.
Now, just in case,
you find this
so charming and seductive,
there's also reasoning
that's abductive and deductive.

Logic is Magic.
Magic is Logic.
Ask Smullyan, who did both,
besides Puzzles and Music.

Evaluation and Abstraction
the Yin and Yang of computation.,
The former, Application,
the latter, Implication,
with Value the destination,
by most estimation.

Alpha is permutable.
Beta, Eta reducible.
Lambda, the Ultimate.
and Omega, perpetual.

"You know its value,
but do you know its cost?"
Laments the Garbage Collector,
and borrows from Frost:
The lists are lovely, long and deep.
But I have promised to mark the Heap,
And bytes to go before I sweep.
And bytes to go before I sweep.

Ah, a whole class in verse!
Now, isn't that perverse?
But, what is a program,
if not a poem?
At least, this is what
I wish to show 'em.

I could go on and on,
waxing along, right or wrong.
But now it's time
to close this song.
A functional program
would never be so long.