

Functional Programming, or not.

Aaron W. Hsu

`<arcfide@sacrideo.us>`

November 17, 2010

St. Louis Alt.net

A Little Background

- 3rd Year Ph.D. in Computer Science at IU
 - Programming Languages
 - Artificial Intelligence
- Member, Scheme Language Working Groups
- Primary Language of Choice: *Scheme*
- Secondary Language: *C*
- *Not* a functional programming zealot
- Love to hack

Functional Programming

- What is OOP?
 - Objects?
- Is a man judged by his name alone? (*Maybe*)
- Emphasis on Control Flow
(a.k.a. --- functions)
- Devalues explicit state change

Functional Programming is not...

- A language
- A language feature
- The One True Way
- Understood well in existing business workflows

Abstractions, abstractions,
Abstractions!

Principia Functiona

- Recursion (Church-Turing Thesis)
- Closures and Higher-order functions
- Compositionality
- Referential Transparency
- Functional Data Structures
- Continuations fly first-class

Macros!

Benefits (Who cares?)

Improves:

- Readability and Conciseness (not tersity)
- Writeability
- Reliability and Robustness
- Maintainability
- Multi-threading
- *Your Quality of Life*

Blackboxes v. Adaptability

- OOP encourages pyramids of blackboxes
 - Hard to modify
 - Must stand unchanged for many iterations
- Functional Programming (especially Lisps) encourage organic, rapidly adapting programs
- Knuth against uneditable libraries

Using it right now

- Modern languages are slowly adding FP features
- Paradigms are language independent