

ON FUNCTIONAL PROGRAMMING

with examples (primarily in Clojure)

Michael S. Daines
November 2015

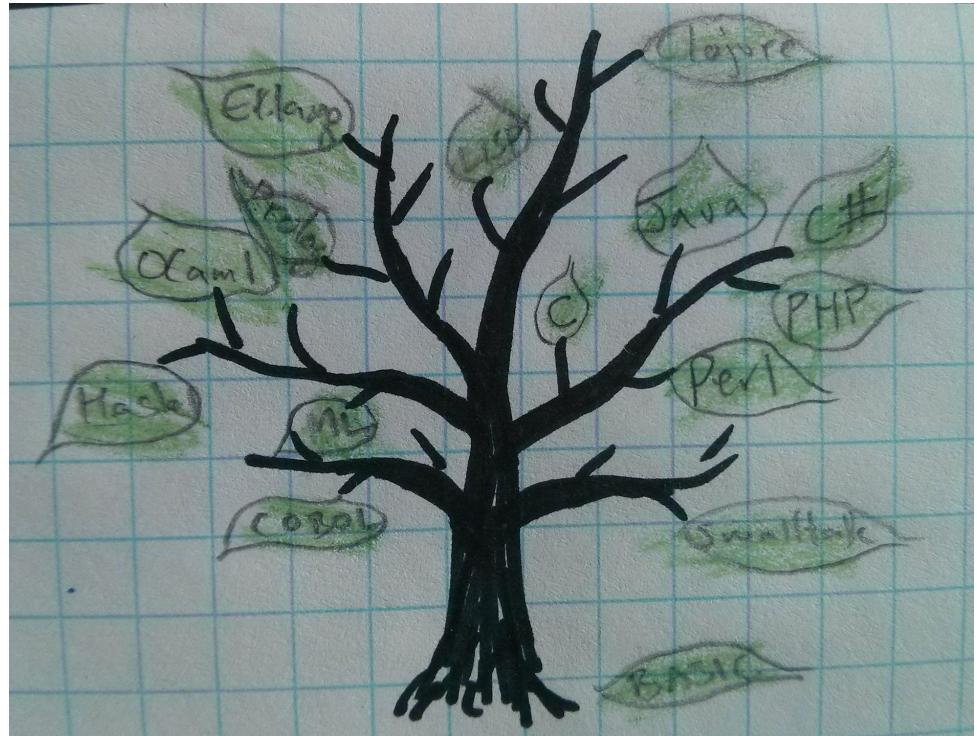
ONCE UPON A TIME



A younger programmer had learned several languages, tools, and frameworks.

THE C FAMILY

But they were all strikingly similar.



PROCEDURAL PROGRAMMING



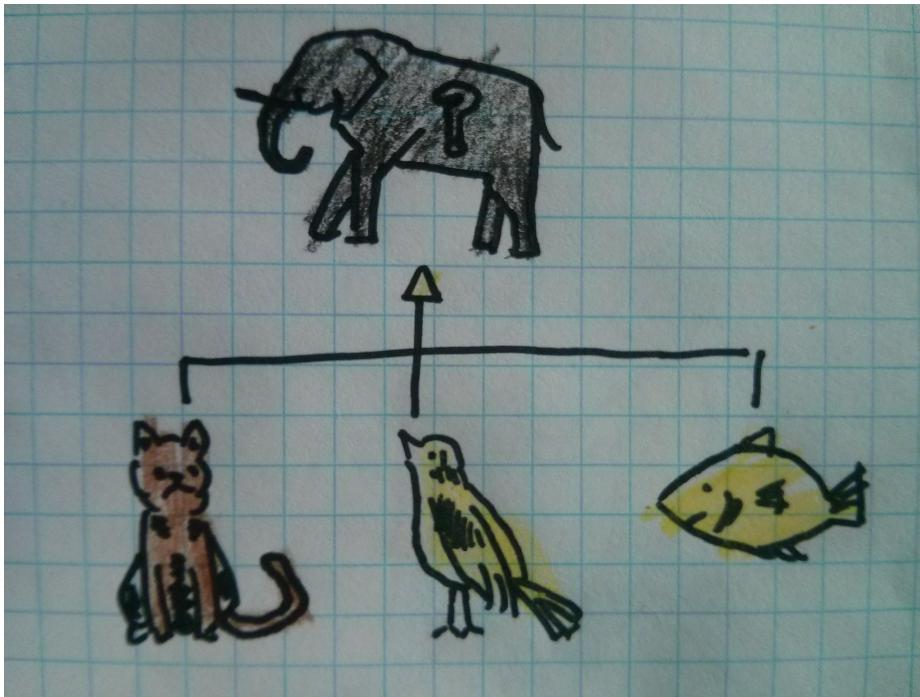
He could tell the computer what to do.

In fact, he had to....

// Micromanagement of array.

```
for (int i = 0; i <= arr.length; i++) {  
    workWith(arr[i]);  
}
```

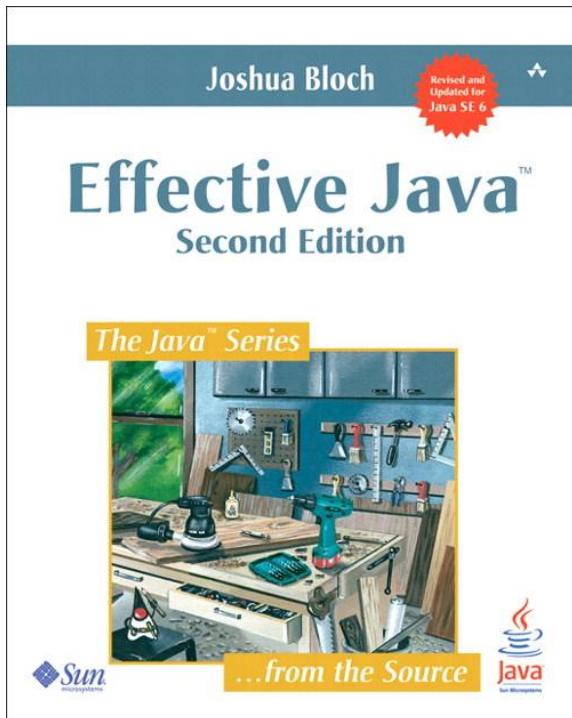
OBJECT ORIENTED PROGRAMMING



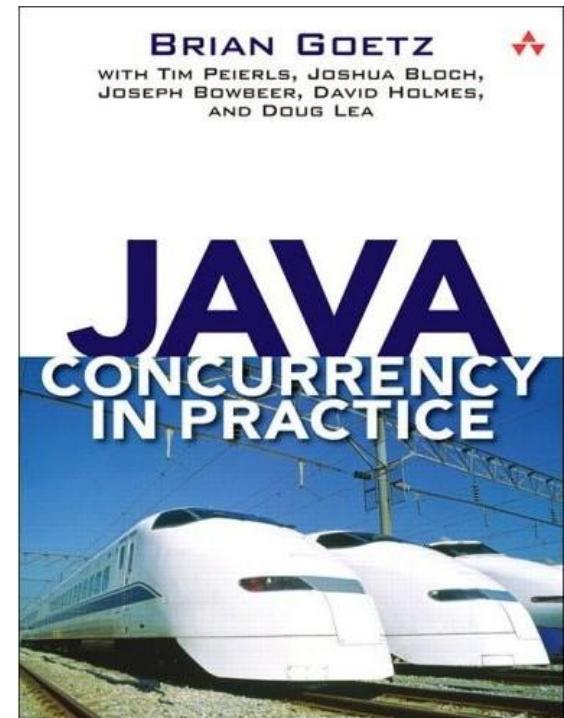
State is constrained inside objects,
but still needs to be coordinated.

Inheritance, abstraction, re-use.

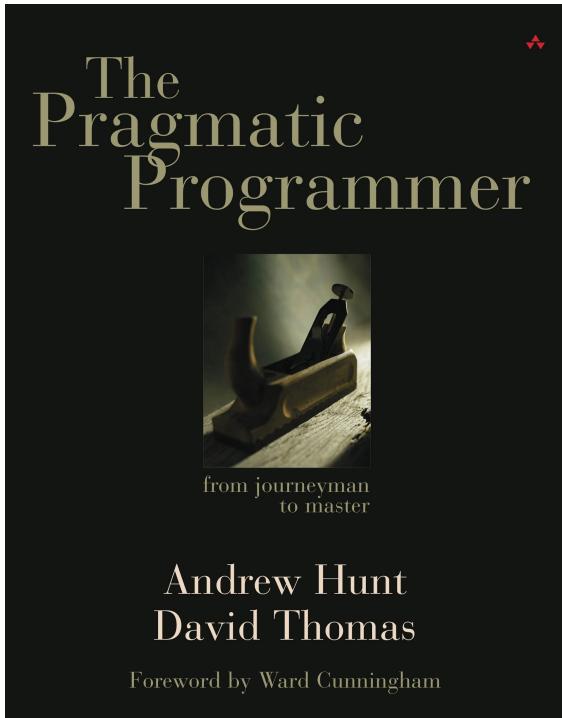
COMPLICATED PROBLEMS



Solving bigger problems
meant new approaches.



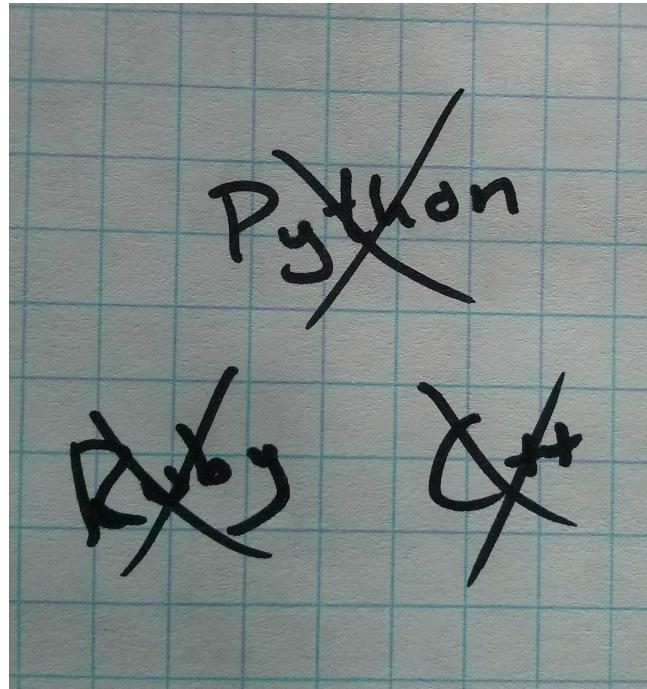
LEARN A NEW LANGUAGE EVERY YEAR



He hadn't read it, but there was this advice from "The Pragmatic Programmer" floating around.

CHALLENGING THE WAY YOU THINK

Not just about new skills.



CLOJURE



Clojure

A LISP which runs on the JVM
(and browsers, CLR, ...)

LISP

First lisp was 1959.



ELEGANCE



Minimal syntax yet great power.

FUNCTIONAL PROGRAMMING

"It is better to have 100 functions operate on one data structure than 10 functions on 10 data structures."

— Alan Perlis, *Epigrams on Programming*, 1982

INTERACTIVE DEVELOPMENT

In praise of the REPL.

SIMPLER CODE

Writing simpler code.

Single responsibility.

Tends to isolate state changes.

DATA IS KING



Functional Programming is all about
the flow of data.

TELL DON'T ASK

Break apart dependencies

Code is independent.

.....
... STEP INTO THE CODE HERE ...
.....

WHAT DID HE LEARN?

MUTATING STATE

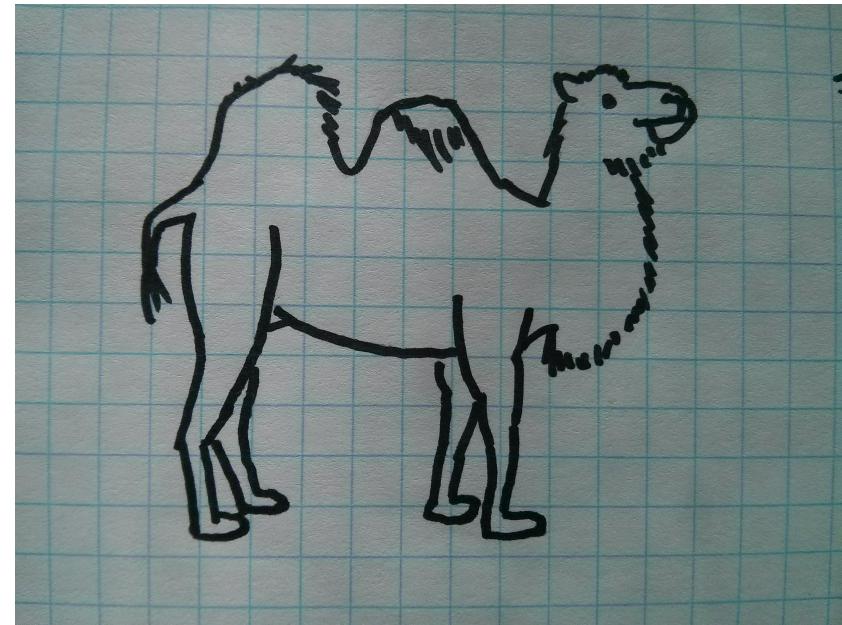
Mutating state everywhere!

Incidental complexity.



WHAT A NEW LANGUAGE CAN TEACH YOU

Objects \neq Types



ON OBJECT ORIENTED PROGRAMMING



"OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things. It can be done in Smalltalk and in LISP. There are possibly other systems in which this is possible, but I'm not aware of them."

-- Alan Kay, 2003

(he coined the term "object-oriented programming" in 1967)

SPECIAL MICRO LANGUAGES



When we drop down to the algorithm level, I think OO can seriously thwart reuse. In particular, the use of objects to represent simple informational data is almost criminal in its generation of per-piece-of-information micro-languages, i.e., the class methods, versus far more powerful, declarative, and generic methods like relational algebra. Inventing a class with its own interface to hold a piece of information is like inventing a new language to write every short story. This is anti-reuse, and, I think, results in an explosion of code in typical OO applications.

-- Rich Hickey

POLYMORPHISM

Teaching old dogs new tricks.

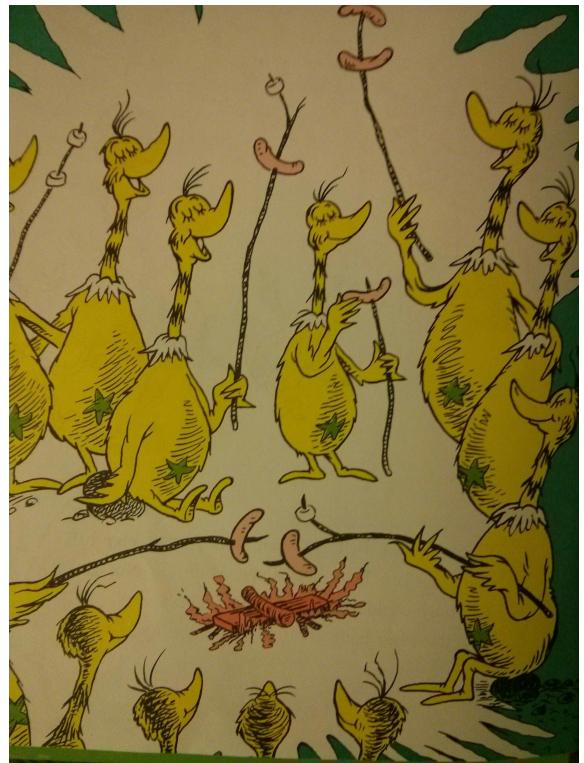


“Java has means of abstraction, like interfaces. But applying abstraction via interfaces (for example, creating an *Emailable* interface to be able to generically use *getEmail* on both *Company* and *Person*) means a) ADDING code (a new interface) and b) CHANGING existing code (*Company* and *Person*). Thus the act of abstraction both makes your code base bigger and is invasive.”

-- Alex Miller

COMMUNITY

New languages and tools
come with a community.



WE DON'T JUST USE "ONE" LANGUAGE, ANYWAY



No one codes in just one language.

... XML, HTML, CSS, SQL, DSLs, and
honestly any heavy framework
(Hibernate, Spring, etc.)

OTHER LANGUAGES TO CONSIDER



Seven Languages in Seven Weeks

A Pragmatic Guide
to Learning
Programming
Languages



Bruce A. Tate

Edited by Jacqueline Carter

Seven More Languages in Seven Weeks

Languages That Are
Shaping the Future



Bruce A. Tate, Fred Daoud,
Ian Dees, and Jack Moffitt

Foreword by José Valim

Edited by Jacqueline Carter



Also, Clojure has ClojureScript, so you can write in the same language on the server and client. And awesome libraries for asynchronous (CSP) processing, logic programming, etc.

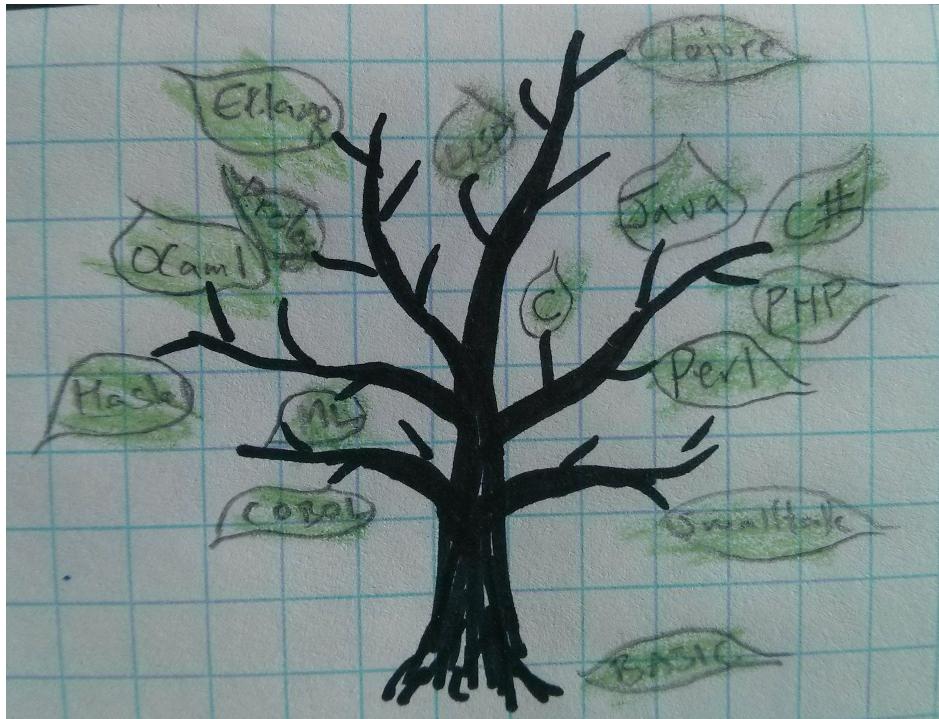
ELM



ERLANG & ELIXIR



WHAT FEELS RIGHT FOR YOU?



Until you've tried a few paradigms?

What aligns with the way you think?

乱取り?

Interested in learning together?



RECOMMENDATIONS

Rich Hickey, “Simple Made Easy”

<http://www.infoq.com/presentations/Simple-Made-Easy>

Rich Hickey, “Hammock Driven Development”

<https://www.youtube.com/watch?v=f84n5oFoZBc>