java.next

stuart halloway
http://thinkrelevance.com

most java code is bad

repetitive, bureaucratic code

untested/untestable

stuck on an old version of _____

miles and miles of crap

java can also be part of the solution

Clojure

Groovy

JRuby

Scala

java, circa 2003

```java
public ActionForward edit(ActionMapping mapping,
                          ActionForm form,
                          HttpServletRequest request,
                          HttpServletResponse response)
    throws Exception {
  PersonForm personForm = (PersonForm) form;
  if (personForm.getId() != null) {
    PersonManager mgr =
      (PersonManager) getBean("personManager");
    Person person = mgr.getPerson(personForm.getId());
    personForm = (PersonForm) convert(person);
    updateFormBean(mapping, request, personForm);
  }
  return mapping.findForward("edit");
}
```

java.next:
2005-?

```ruby
def edit
  @person = Person.find(params[:id])
end

def new
  @person = Person.new
end
```

evolving style

convention over configuration

reasonable defaults

no checked exceptions

YAGNI

domain nouns, not language nouns

DSLs

how java.next can help

Photo credit: http://www.flickr.com/photos/uhduh/92437037/sizes/l/

common java.next features

everything is an object

easy beans

higher-order functions

unchecked exceptions

open APIs

DSLs

everything is an object

```
; is java enterprise-ready?
2000000 * 2000000
-> 1385447424
```

```
# ruby is
2_000_000 * 2_000_000
=> 4000000000000
```

easy beans

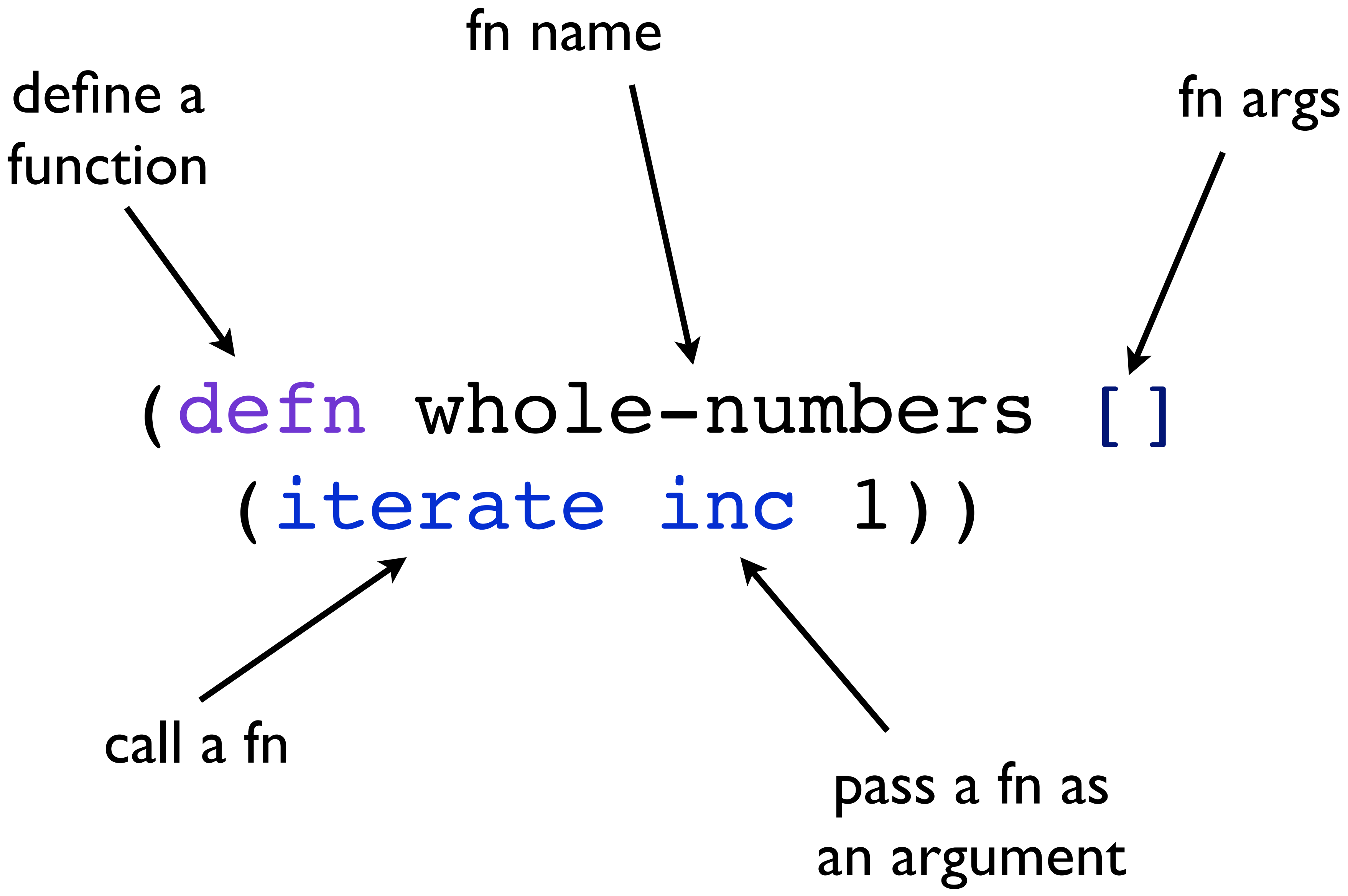
convenient constructor

immutable

```
// scala
case class Person(firstName: String,
                  lastName: String) {}
```

static typing

higher-order functions



```
(defn whole-numbers []
  (iterate inc 1))
```

anonymous function

```
(map (fn [x] (* x x))
     (whole-numbers)))
-> (1 4 9 16 25 36 49 64 81 100 ...)
```

sequence API

lazy evaluation

```
(filter odd? (whole-numbers))
-> (1 3 5 7 9 11 13 15 17 19 ...)
```

Ceremony is: checked exceptions.
Photo credit: http://www.flickr.com/photos/marinegirl/2036373729

closed
APIs

closed

```java
// Java (from the Jakarta Commons)
public class StringUtils {
  public static boolean isBlank(String str) {
  int strLen;
  if (str == null || (strLen = str.length()) == 0) {
    return true;
  }
  for (int i = 0; i < strLen; i++) {
    if ((Character.isWhitespace(str.charAt(i)) == false)) {
      return false;
    }
  }
}

```

open
APIs

groovy metaclass

```groovy
// Groovy
String.metaClass.isBlank = {
  length() == 0 || every{ Character.isWhitespace(it.charAt(0)) }
}
```

ruby open class

```ruby
# Ruby (from Rails)
class String
  def blank?
    empty? || strip.empty?
  end
end
```

scala implicit coercion

```
// Scala
class CharWrapper(ch: Char) {
  def isWhitespace = Character.isWhitespace(ch)
}
implicit def charWrapper(ch: Character) = new CharWrapper(ch)
class BlankWrapper(s: String) {
  def isBlank = s.isEmpty || s.forall(ch => ch.isWhitespace)
}
implicit def stringWrapper(s: String) = new BlankWrapper(s)
```

clojure generic dispatch

```
(defn blank? [s]
  (every? #(Character/isWhitespace %) s))
```

DSLs

ruby (from rails)

methods feel
like keywords

key/value options

```
class Coach < ActiveRecord::Base
  belongs_to :team
  has_many :sponsors, :as => :spokesperson
  validates_presence_of :first_name, :last_name
end
```

flexible punctuation

groovy

```
//from EasyB, www.easyb.org
given "an invalid zip code", {
  invalidzipcode = "221o1"
}

and

given "the zipcodevalidator is initialized", {
  zipvalidate = new ZipCodeValidator()
}

when "validate is invoked", {
  value = zipvalidate.validate(invalidzipcode)
}

then "the validator should return false", {
  ensure(!value)
}
```

function arguments

flexible punctuation

scala

```scala
// From Programming in Scala (PrePrint Edition)
import scala.util.parsing.combinator._
class Arith extends JavaTokenParsers {
  def expr: Parser[Any] = term~rep("+"~term | "-"~term)
  def term: Parser[Any] = factor~rep("*"~factor | "/"~factor)
  def factor: Parser[Any] = floatingPointNumber | "("~expr~")"
}
```

extremely
flexible punctuation

clojure

data is syntax

```
; blog.thinkrelevance.com/2008/9/16/pcl-clojure-chapter-3
(filter (where {:artist "Dixie Chicks" :rating 8}) (init-db))

(defn where [criteria]
  (fn [m]
    (every? (fn [[k v]] (= (k m) v)) criteria)))
```

everything is a sequence

which language should I learn?

every time you
start a green-field
Java project,
God kills a kitten

clojure considerations

+ functional

+ multimethods

+ concurrency (STM et al)

+ lisp

+ a la carte

- youngest java.next language

groovy considerations

+ easiest to learn

+ easiest bi-di interop

? more committed to reusing Java libs

- worst Java baggage

- no concurrency/multicore story

ruby considerations

+ biggest community

+ commercial support: ~~Sun~~ ~~Oracle~~ EngineYard

+ Rails

+ multiple platforms

- no concurrency/multicore story

scala considerations

+ functional

+ high performance

+ pattern matching

+ actor model

? hybrid object/functional

- hardest to learn

contact stu

Email:	stu@thinkrelevance.com
Office:	919-442-3030
Twitter:	twitter.com/stuarthalloway
Talks:	http://blog.thinkrelevance.com/talks
Blog:	http://blog.thinkrelevance.com
Java.next:	http://blog.thinkrelevance.com/2008/9/24/java-next-overview
Book:	http://www.pragprog.com/titles/shcloj/programming-clojure