Lisp in a Startup:
The Good, The Bad & The Ugly

Vsevolod Dyomkin
Franz Inc.
2018-04-17 @ ELS

A Bit about Me

(m8n)

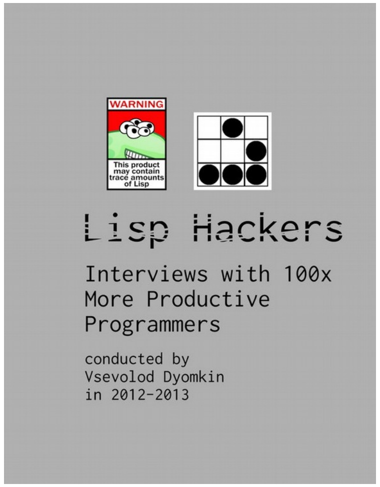
Copyleft Solutions

FACT or FICTION

FRANZ INC.

grammarly

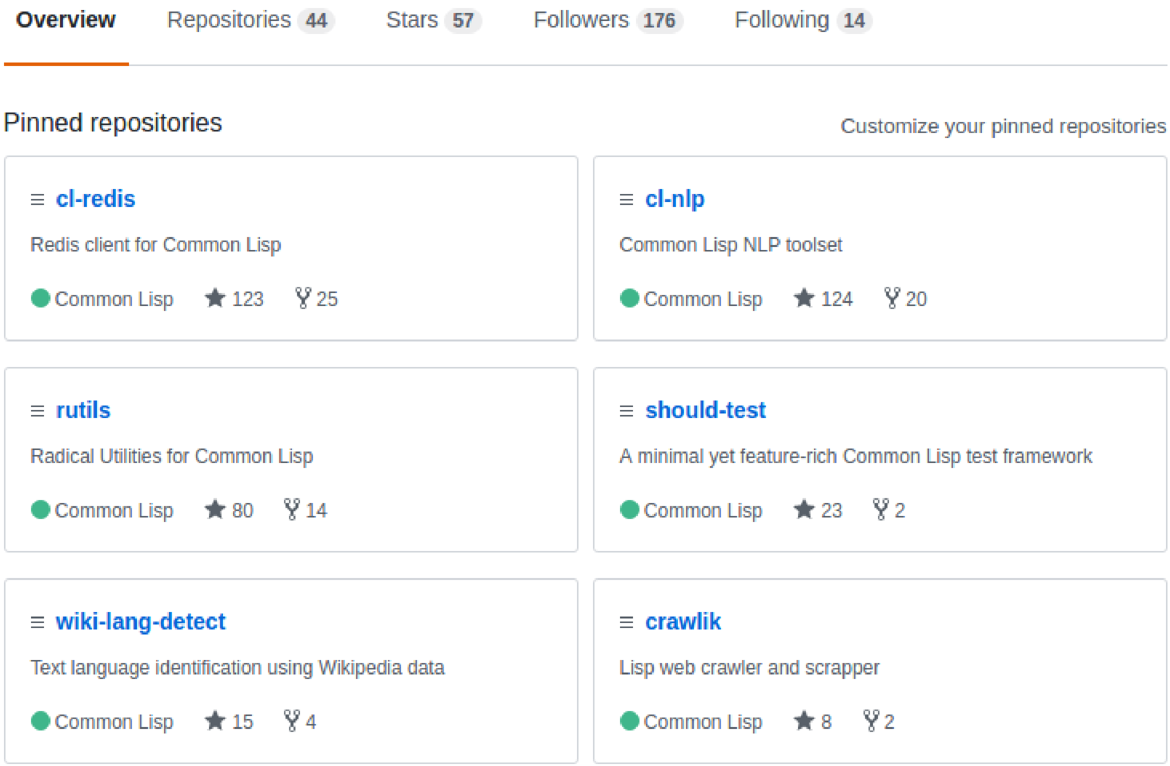
LISP, THE UNIVERSE AND EVERYTHING



Lisp Hackers

Interviews with 100x More Productive Programmers

conducted by Vsevolod Dyomkin in 2012-2013



Pinned repositories

Repository	Description	Language	Stars	Forks
cl-redis	Redis client for Common Lisp	Common Lisp	123	25
cl-nlp	Common Lisp NLP toolset	Common Lisp	124	20
rutils	Radical Utilities for Common Lisp	Common Lisp	80	14
should-test	A minimal yet feature-rich Common Lisp test framework	Common Lisp	23	2
wiki-lang-detect	Text language identification using Wikipedia data	Common Lisp	15	4
crawlik	Lisp web crawler and scrapper	Common Lisp	8	2

https://vseloved.github.io

(m8n)

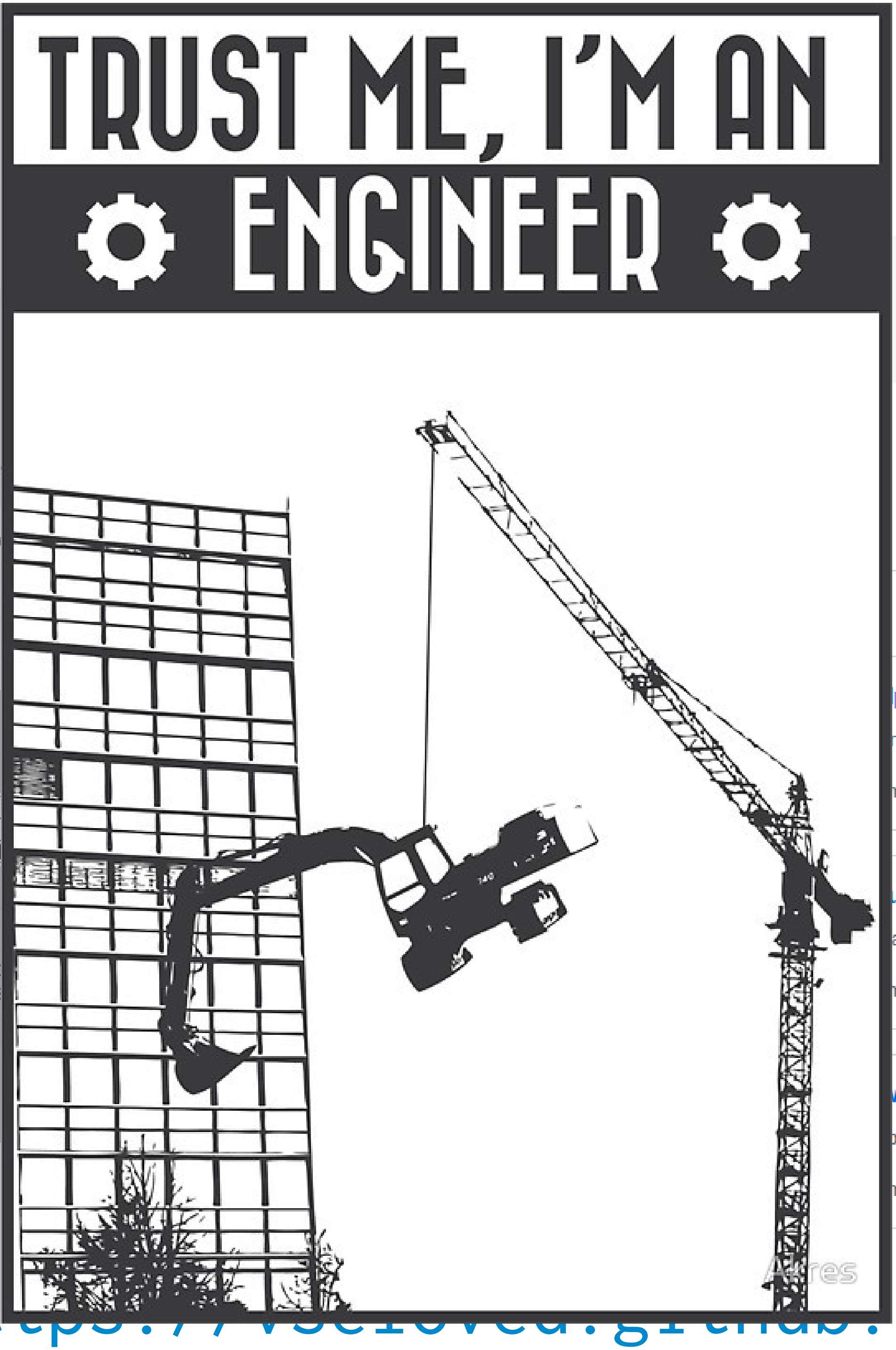
Copyleft Solutions

gramm

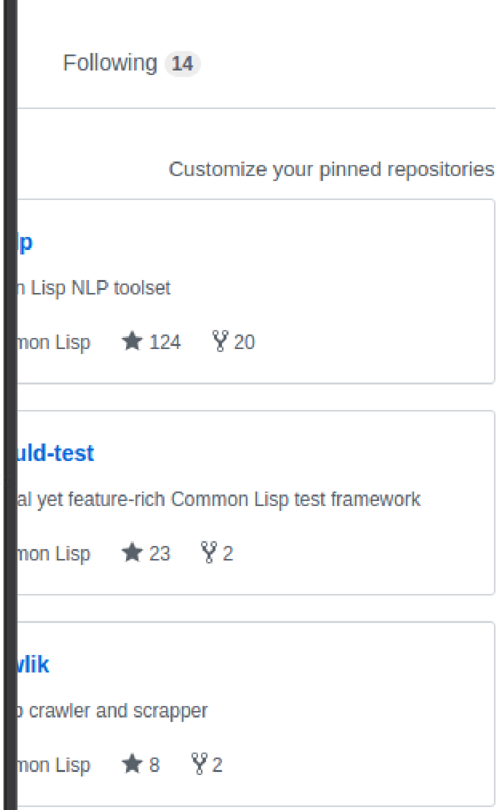
Lisp H

Interviews
More Produc
Programmers

conducted by
Vsevolod Dyomki
in 2012-2013



IVERSE AND


Following 14

Customize your pinned repositories

lp
n Lisp NLP toolset
mon Lisp ★ 124 ⑂ 20

uld-test
al yet feature-rich Common Lisp test framework
mon Lisp ★ 23 ⑂ 2

wlik
o crawler and scrapper
mon Lisp ★ 8 ⑂ 2

https://vseloved.github.io

Startup Requirements

* Support research & experimentation
* Flexibility

Not:

* Safety
* Ease of reusing existing solutions
* User-friendliness
* Whatever

The Good: The Language

* Dynamism
* Interactivity
* Uniformity

Dynamism

Not just:

* Dynamic typing
* OO with late binding

Dynamism

Dynamism

Not just:
* Dynamic typing
* OO with late binding

What can (should?) also be redefined:
* syntax
* program flow
* type signatures & class hierarchies
* executable code
* namespaces
* dependencies
* you name it

(signal ‘commited)

I'M JUST OUTSIDE TOWN, SO I SHOULD BE THERE IN FIFTEEN MINUTES.
ACTUALLY, IT'S LOOKING MORE LIKE SIX DAYS.
NO, WAIT, THIRTY SECONDS.

THE AUTHOR OF THE WINDOWS FILE COPY DIALOG VISITS SOME FRIENDS.

DLL Hell

**“Loading Multiple Versions of
an ASDF System in the Same Image”**
https://www.youtube.com/watch?v=VN58mZsHWXk

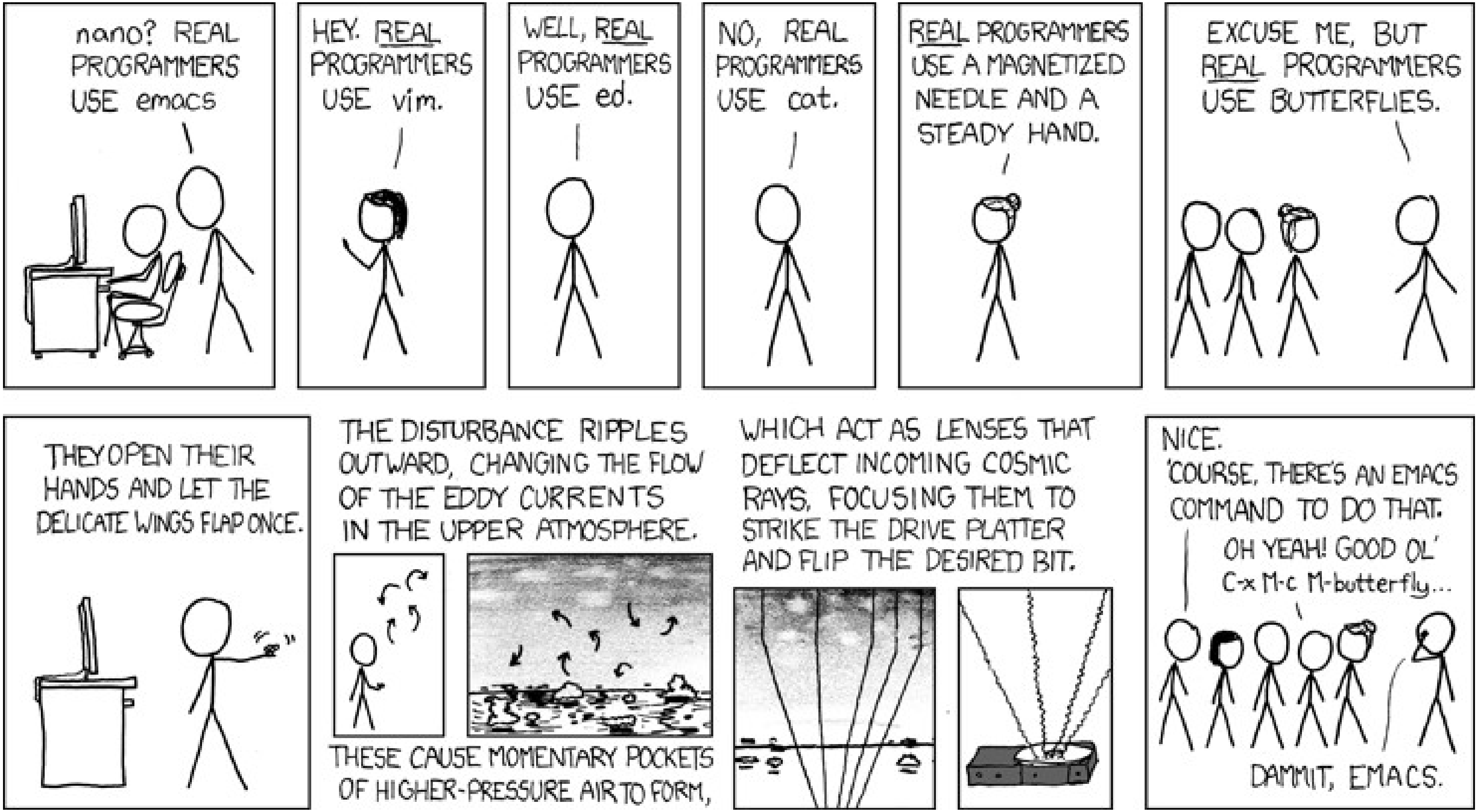
DSLs

```
(defmethod crawlik:scrape ((site geo) source)
  (crawlik:match-html
   source
   '(>> (div :class ”lastcol ...”)
        (>> (div :class ”portlet_content”)
            (ul (li)
                (>> (span :class ”counts”) ($ gds))
                (>> (span :class ”counts”) ($ gse)))))))
```

```
(defmethod match-expr ((head (eql :$)) &rest tail)
  ”Match variable: ($ name &optional expr).”
  (when (or (single tail)
            (let ((*tree* *tree*))
              (apply ’match-expr (rest tail))))
    (:= (? *vars* (first tail)) *tree*
        *tree* nil)
    t))

(defmethod match-expr ((head (eql :>>)) &rest tail)
  ”Match by tree depth-first search:
   (>> tag &rest contents).”
  (or (when (apply ’match-expr tail)
        !!!)
      (when (listp *tree*)
        (let (matched)
          (dolist (*tree* (rest *tree*) matched)
            (when (apply ’match-expr :>> tail)
              (if *match-multiple*
                  (:= matched t)
                  (return t))))))
      (void *tree*)))
```

Interactivity


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

The “Hardest” Bug

```
Socket error in "socket": EPROTONOSUPPORT (Protocol not supported)
  [Condition of type SB-BSD-SOCKETS:PROTOCOL-NOT-SUPPORTED-ERROR]
…
```

the problem (analysis by John Fremlin):

The shared-initialize calls getprotobyname which is not thread safe.

There is little need to call this function at runtime anyway as these proto numbers are quite stable, but getprotobyname_r is a safe alternative (on GNU/Linux).

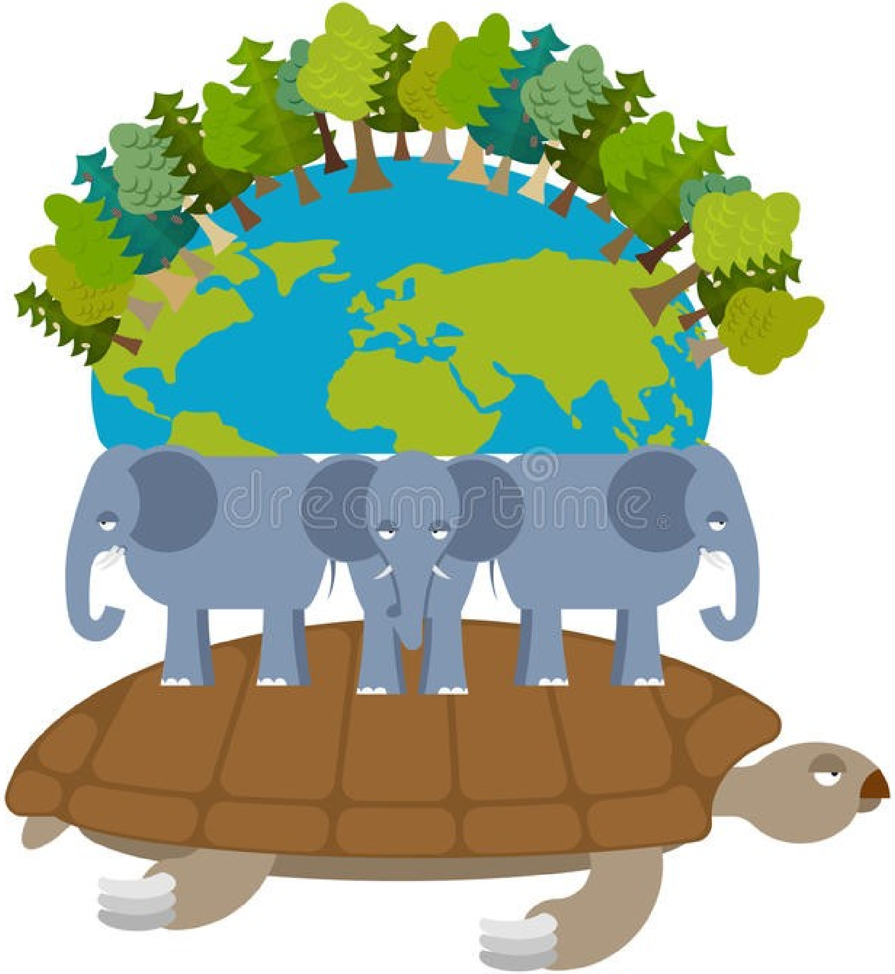
“Fix”:

```
(defun sb-bsd-sockets:get-protocol-by-name (name)
  (case name
    (:tcp 6)
    (:udp 17)))
```

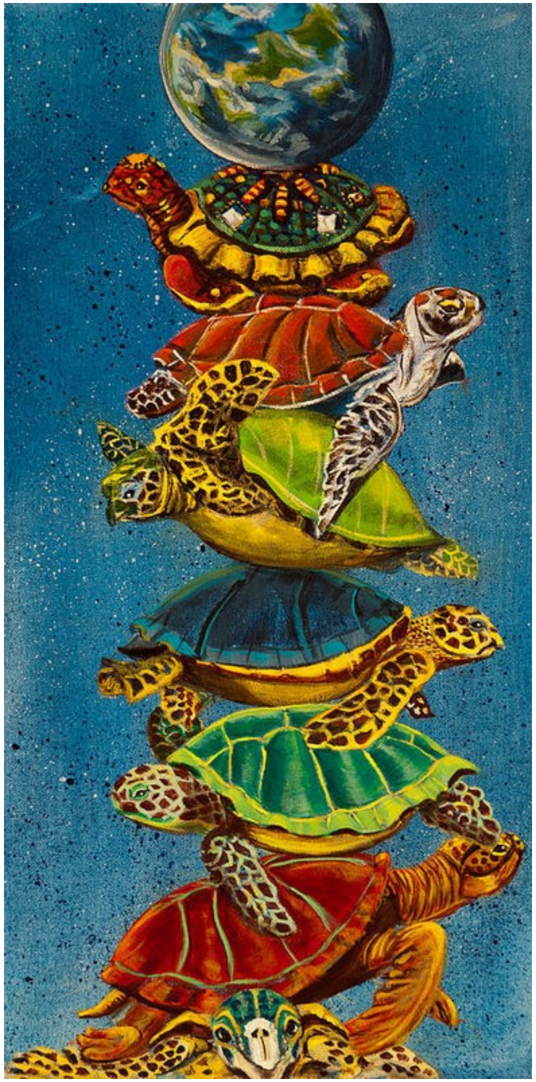
https://bugs.launchpad.net/sbcl/+bug/505497

See also: https://tech.grammarly.com/blog/running-lisp-in-production

Uniformity



=>



http://lisp-univ-etc.blogspot.com/2013/04/errors-in-your-platform.html

If I could change just 1 thing…

If I could change just 1 thing…

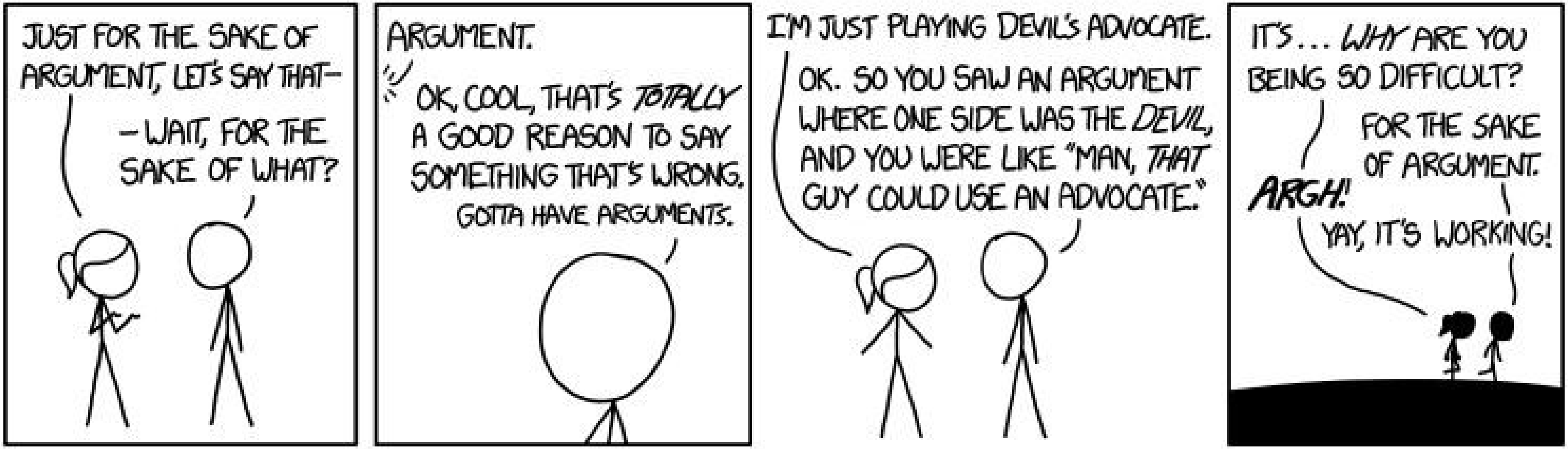
Eradicate the .-syntax for cons cells!

```
(defun |...-reader| (stream char)
  (when (char= #\. (peek-char nil stream))
    (unless (equal ’(#\. #\.)
                   (list (read-char stream)
                         (read-char stream)))
      (error ”malformed ...”))
    ’|...|))
```

3 Flavors of Gains

* incremental
* game-changers
* synergy

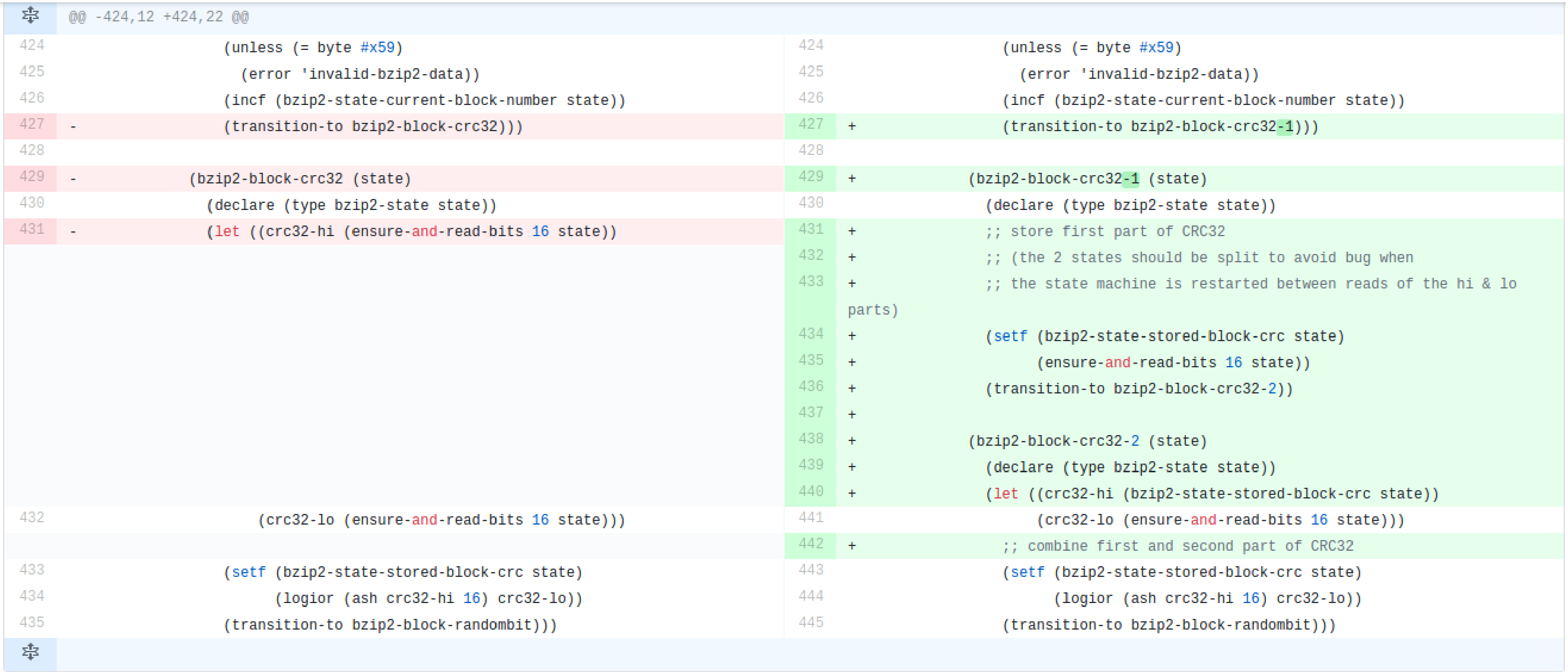
The Bad: The Environment


JUST FOR THE SAKE OF ARGUMENT, LET'S SAY THAT—
—WAIT, FOR THE SAKE OF WHAT?
ARGUMENT.
OK, COOL, THAT'S *TOTALLY* A GOOD REASON TO SAY SOMETHING THAT'S WRONG.
GOTTA HAVE ARGUMENTS.
I'M JUST PLAYING DEVIL'S ADVOCATE.
OK. SO YOU SAW AN ARGUMENT WHERE ONE SIDE WAS THE *DEVIL*, AND YOU WERE LIKE "MAN, *THAT* GUY COULD USE AN ADVOCATE."
IT'S... *WHY* ARE YOU BEING SO DIFFICULT?
FOR THE SAKE OF ARGUMENT.
ARGH!
YAY, IT'S WORKING!

The Modern Programmer

Abandonware

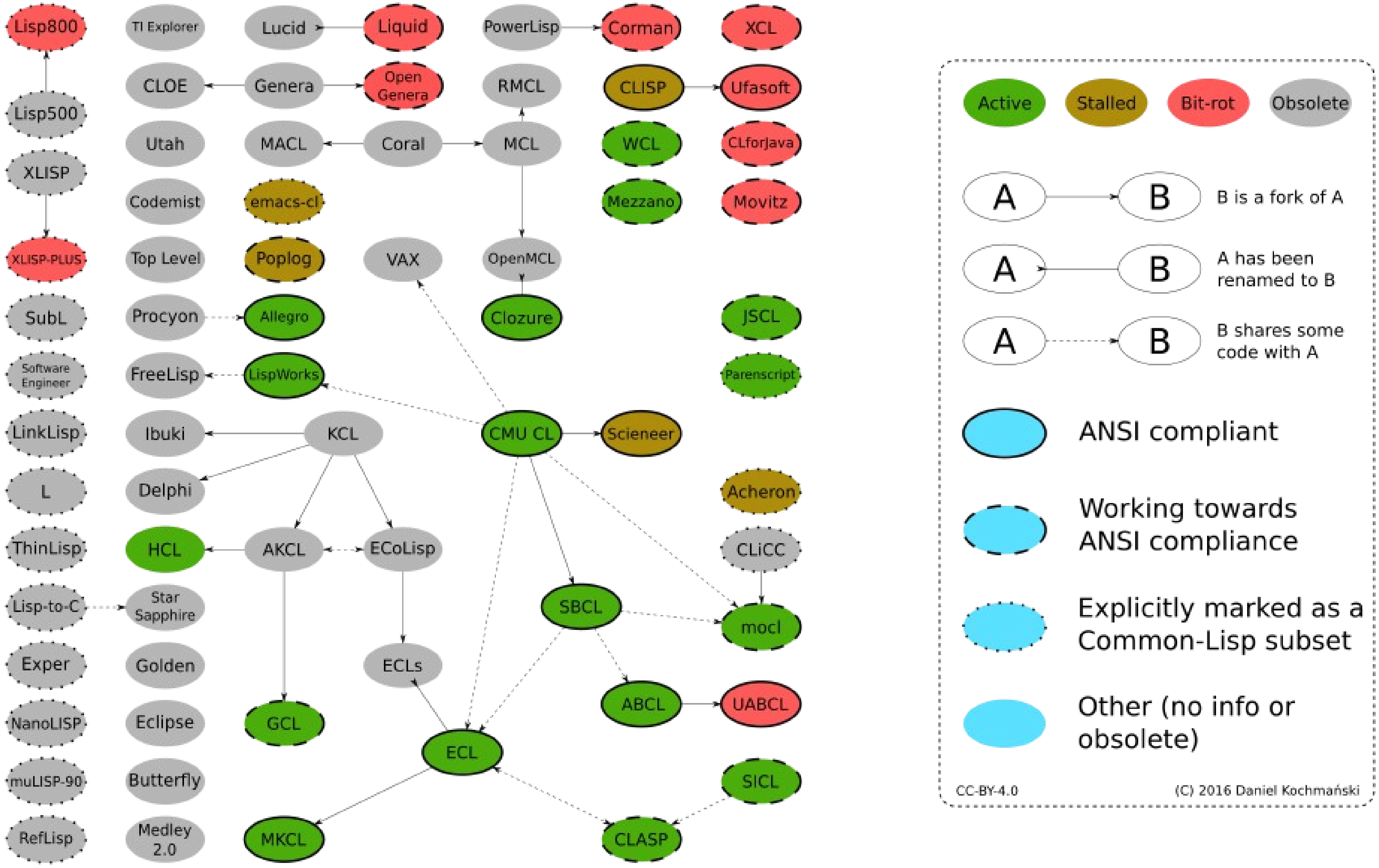
bzip2 bug



```diff
@@ -424,12 +424,22 @@
        (unless (= byte #x59)
          (error 'invalid-bzip2-data))
        (incf (bzip2-state-current-block-number state))
-       (transition-to bzip2-block-crc32)))
+       (transition-to bzip2-block-crc32-1)))

-   (bzip2-block-crc32 (state)
+   (bzip2-block-crc32-1 (state)
      (declare (type bzip2-state state))
-     (let ((crc32-hi (ensure-and-read-bits 16 state))
+     ;; store first part of CRC32
+     ;; (the 2 states should be split to avoid bug when
+     ;; the state machine is restarted between reads of the hi & lo parts)
+     (setf (bzip2-state-stored-block-crc state)
+         (ensure-and-read-bits 16 state))
+     (transition-to bzip2-block-crc32-2))
+
+   (bzip2-block-crc32-2 (state)
+     (declare (type bzip2-state state))
+     (let ((crc32-hi (bzip2-state-stored-block-crc state))
            (crc32-lo (ensure-and-read-bits 16 state)))
+       ;; combine first and second part of CRC32
        (setf (bzip2-state-stored-block-crc state)
            (logior (ash crc32-hi 16) crc32-lo))
        (transition-to bzip2-block-randombit)))
```

Implementation Issues



YMMV

* IDEs/editor support
* Documentation/manuals
* What else?

The Ugly: The FAD

Just like half of the world's spoken tongues, most of the 2,300-plus computer programming languages are either endangered or extinct. As powerhouses C/C++, Visual Basic, Cobol, Java and other modern source codes dominate our systems, hundreds of older languages are running out of life.

An ad hoc collection of engineers-electronic lexicographers, if you will-aim to save, or at least document the lingo of classic software. They're combing the globe's 9 million developers in search of coders still fluent in these nearly forgotten lingua frangas. Among the most endangered are Ada, APL, B (the predecessor of C), Lsp, Oberon, Smalltalk, and Simula.

Code-raker Grady Booch, Rational Software's chief scientist, is working with the Computer History Musuem in Silicon Valley to record and, in some cases, maintain languages by writing new compilers so our ever-changing hardware can grok the code. Why bother? "They tell us about the state of software practice, the minds of their inventors, and the technical, social, and economic forces that shaped history at the time," Booch explains. "They'll provide the raw material for software archaeologists, historians, and developers to learn what worked, what was brilliant, and what was an utter failure." Here's a peek at the strongest branches of programming's family tree. For a nearly exhaustive rundown, check out the Language List at *HTTP://www.informatik.uni-freiburg.de/Java/misc/lang_list.html.* - Michael Mendeno

Key

1954 **Year Introduced**

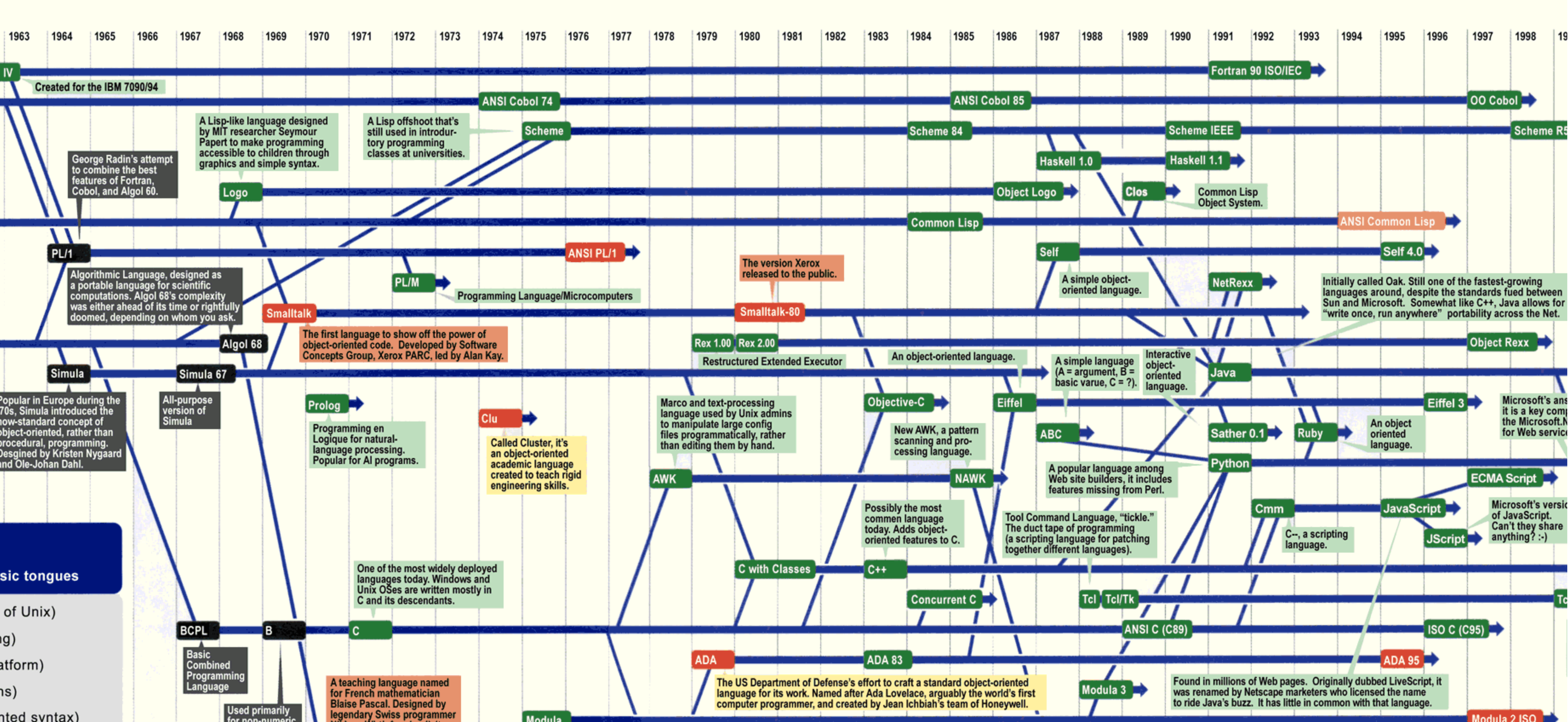
🟩 **Active:** thousands of users

🟥 **Protected:** taught at universities available

🟥 **Endangered:** usage dropping of

⬛ **Extinct:** no known active users o compilers

➡ **Lineage continues**



Created for the IBM 7090/94

A Lisp-like language designed by MIT researcher Seymour Papert to make programming accessible to children through graphics and simple syntax.

A Lisp offshoot that's still used in introdurtory programming classes at universities.

George Radin's attempt to combine the best features of Fortran, Cobol, and Algol 60.

Common Lisp Object System.

The version Xerox released to the public.

Algorithmic Language, designed as a portable language for scientific computations. Algol 68's complexity was either ahead of its time or rightfully doomed, depending on whom you ask.

Programming Language/Microcomputers

A simple object-oriented language.

Initially called Oak. Still one of the fastest-growing languages around, despite the standards fued between Sun and Microsoft. Somewhat like C++, Java allows for "write once, run anywhere" portability across the Net.

The first language to show off the power of object-oriented code. Developed by Software Concepts Group, Xerox PARC, led by Alan Kay.

Restructured Extended Executor

An object-oriented language.

A simple language (A = argument, B = basic varue, C = ?).

Interactive object-oriented language.

Popular in Europe during the 70s, Simula introduced the now-standard concept of object-oriented, rather than procedural, programming. Designed by Kristen Nygaard and Ole-Johan Dahl.

All-purpose version of Simula

Programming en Logique for natural-language processing. Popular for AI programs.

Called Cluster, it's an object-oriented academic language created to teach rigid engineering skills.

Marco and text-processing language used by Unix admins to manipulate large config files programmatically, rather than editing them by hand.

New AWK, a pattern scanning and processing language.

An object oriented language.

Microsoft's ans it is a key comp the Microsoft.N for Web servic

A popular language among Web site builders, it includes features missing from Perl.

Possibly the most commen language today. Adds object-oriented features to C.

Tool Command Language, "tickle." The duct tape of programming (a scripting language for patching together different languages).

C--, a scripting language.

Microsoft's versio of JavaScript. Can't they share anything? :-)

One of the most widely deployed languages today. Windows and Unix OSes are written mostly in C and its descendants.

Basic Combined Programming Language

A teaching language named for French mathematician Blaise Pascal. Designed by legendary Swiss programmer

Used primarily for non-numeric

The US Department of Defense's effort to craft a standard object-oriented language for its work. Named after Ada Lovelace, arguably the world's first computer programmer, and created by Jean Ichbiah's team of Honeywell.

Found in millions of Web pages. Originally dubbed LiveScript, it was renamed by Netscape marketers who licensed the name to ride Java's buzz. It has little in common with that language.

sic tongues

of Unix)

g)

atform)

ns)

ted syntax)

Fight FAD?

* Killer App?
  well, we already have Emacs

* Success Story?
  ITA ($1B), Grammarly, etc.

Fight FAD?

* Killer App?
  well, we already have Emacs

* Success Story?
  ITA ($1B), Grammarly, etc.

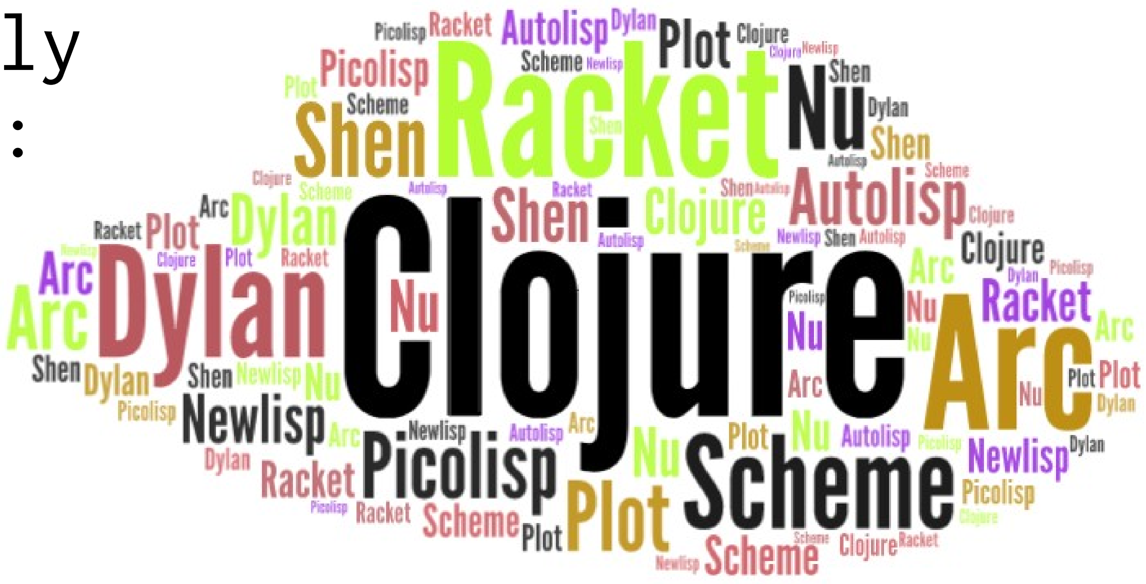
… good, but not enough -
  we need other positive directions

Rebranding?

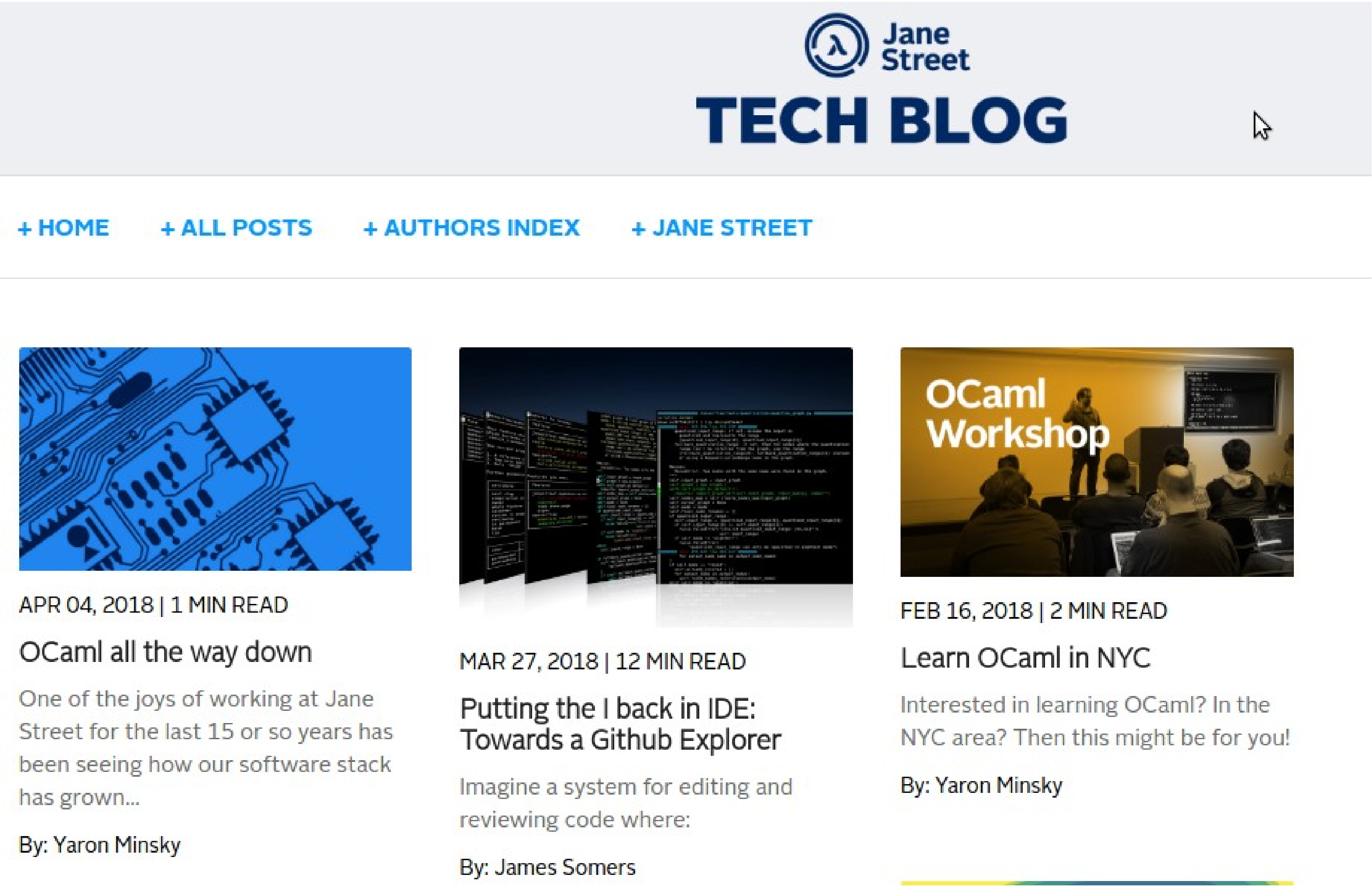
~~Common~~ Lisp

http://lisp-univ-etc.blogspot.com.es/2013/01/common-lisp-is-just-lisp.html

Lisp-family languages:



A Company-Champion?



APR 04, 2018 | 1 MIN READ

OCaml all the way down

One of the joys of working at Jane Street for the last 15 or so years has been seeing how our software stack has grown...

By: Yaron Minsky

MAR 27, 2018 | 12 MIN READ

Putting the I back in IDE: Towards a Github Explorer

Imagine a system for editing and reviewing code where:

By: James Somers

FEB 16, 2018 | 2 MIN READ

Learn OCaml in NYC

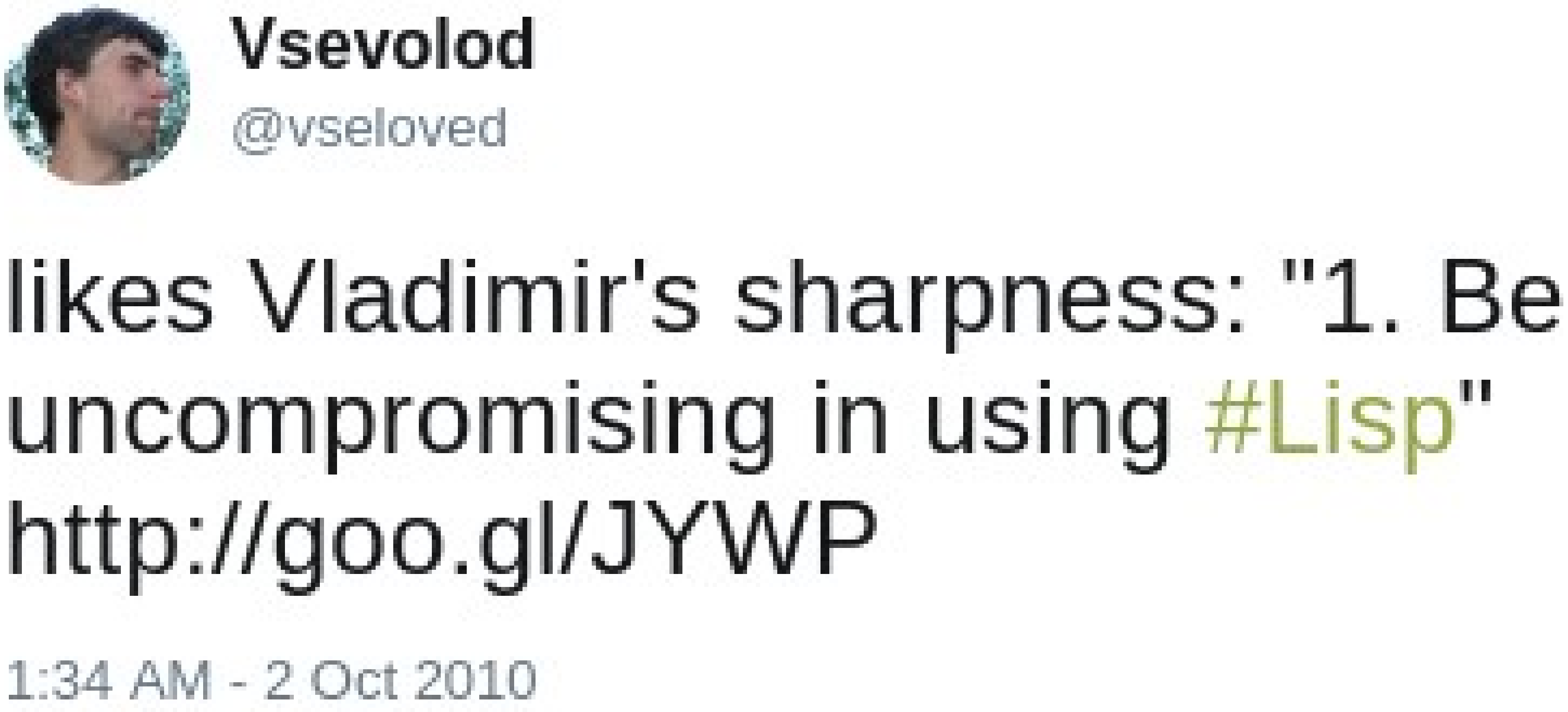
Interested in learning OCaml? In the NYC area? Then this might be for you!

By: Yaron Minsky

A Community Edition?

LispWorks

(ACL)



Vsevolod

@vseloved

likes Vladimir's sharpness: "1. Be uncompromising in using #Lisp" http://goo.gl/JYWP

1:34 AM - 2 Oct 2010