Elliott Cable

Cell: +1 919 636.4204   (SMS, calls, FaceTime welcome)
E-mail: me@ell.io   (NOT CHECKED OFTEN – Call or text first!)
Twitter: twitter.com/ELLIOTTCABLE (DMs welcome)
GitHub: github.com/ELLIOTTCABLE
Addt'l: elliottcable.name

I'm a testing- and documentation-obsessed software engineer, most motivated by
programming-language design, tooling development, and accessibility.

I'm currently most interested in rapid software development using powerful
correctness tools like OCaml and ReasonML; PL theory and JIT design; and
embedded firmware design for RF devices. I'm most *experienced* in JavaScript
library development, Node.js, Ruby, POSIX shell, and ISO C.

In researching support software for a wide range of domains of concern, I'm
additionally at least passably familiar with a vast array of other programming
languages, tools, and platforms. (See footer.)

PROJECT EXPERIENCE

These are neither necessarily my most recent, nor most mature, pieces of work; I
instead opted to try and present a *breadth* of experience – if there's one
primary skill I possess, it's learning new things quickly and thoroughly.

2010–now: Pratchett, the programming language
My overarching work of the past ten years, Pratchett (née Paws) is the
project that has driven much of my personal development as a programmer. From
surveying prospective users' needs, staying on top of similar language work and
PLT research, to organizing a (brief) community of interest and contributors and
triaging design goals to try and keep a sense of constant progress, this has
been my largest ongoing software project. (Yes, it's currently in CoffeeScript.
Technical debt!)

 – http://ell.io/tt$Paws.js/tree/queueless+#README
 – http://ell.io/tt$Paws.js/blob/d1a1f2/Source/datagraph.coffee#L27–L135
 – http://ell.io/tt$Paws.js/blob/d1a1f2/Test/datagraph.tests.coffee#L246
 – http://ell.io/tt$Paws.js/blob/546795/Source/reactor.coffee#L114–L196
 – http://ell.io/tt$Paws.js/blob/d1a1f2/Scripts/test.sh#L3–L35
 – http://ell.io/tt$Paws.c/tree/f4fcb5/Source/Paws.o/Paws.c#L32–L94
 – http://ell.io/tt$Paws.c/blob/f4fcb5/Source/Paws.o/Ancillary/Threading/Thread/Thread.c#L97–L175

2017–now: Unnamed learning project, an R5RS-Scheme compiler in OCaml
At PWLconf 2017, I was introduced to A. Ghuloum's "Incremental Approach to
Compiler Construction" paper. This spawned a still-ongoing project
from which I am learning a substantial amount – about compiler architecture, as
well as general OCaml skill, parsing automata, and Scheme itself.

 – http://ell.io/tt$ocameel/blob/5ac70f/bin/cli.ml#L6–L93
 – http://ell.io/tt$ocameel/blob/db2dda/lib/Lexer.sedlex.body.ml#L40–L97

Various patches, or contributions to others' open-source projects
I'm a compulsive open-source developer and patcher. Although so many of my
changes stay local and never see the light of day (a bad habit!), occasionally,
I clean up and complete my explorations, and remember to submit a pull-request.

 – 2018, Python/OCaml, to OCaml's analysis engine, Merlin:
   http://ell.io/tt$merlin/blob/9b80dd/vim/merlin/autoload/merlin.py#L677–L734

 – 2017, ISO C, adding syscalls to the xv6 operating system, as an exercise:

http://ell.io/tt$xv6/blob/efd94e/README#L10–L57,
http://ell.io/tt$xv6/compare/08429c...379fc6

- 2013, ISO C, adding a history–reorganizing feature to git mainline:
  http://ell.io/tt$git/compare/b5c267...author–order+

2017–now: Giraphe, a configurable JavaScript graph–walking library
Extracted from my work on Paws.js by necessity, this is an API designed to
generate individual, optimized graph–walking iterators for various Pratchett–
related tasks. (It currently only *implements* a generic, unoptimized walker.)

  - http://ell.io/tt$giraphe#README
  - http://ell.io/tt$giraphe/blob/fe85e6/giraphe.es6.js#L122–L257

2016: `pin–cushion`, an API–client for http://Pinboard.in
A small, one–off, but complete and modern, command–line client for my
bookmarking service of choice. Included simply because it's some of the only
recent and pure JavaScript I've written, with no legacy cruft.

  - http://ell.io/tt$pin–cushion#README
  - http://ell.io/tt$pin–cushion/blob/36be70/pin–cushion#L78–L121

2008: ArchLinux image–builder for Amazon EC2
ArchLinux, my preferred distro, was unavailable on Amazon EC2 for years. I built
a fiendishly–complex, modular shell–script architecture for bundling custom
builds of Arch and publishing them to your EC2 hosts. This involved poring
through both Amazon's AWS documentation, as well as that of Arch's own build–
system and packaging. This is the largest pure–POSIX–shell program I've written.

  - http://ell.io/tt$ArchLinux–AMIs/blob/a34646/bundle.sh

2006–2009: Assorted Ruby tools
I was heavily into metaprogramming Ruby for many years — these are some of the
libraries and tools I was most proud of.

  - http://ell.io/tt$it/blob/b52441/lib/it/environmented_proc.rb#L63–L155
  - http://ell.io/tt$lobby/blob/8dc3f6/lib/lobby.rb
  - http://ell.io/tt$stringray/blob/117038/lib/stringray.rb#L3–L162
  - http://ell.io/tt$nfoiled/blob/d8e593/lib/nfoiled/window.rb#L3–L217

ADDITIONAL INFORMATION

I'm an avid maker/hacker, a U.S. General–class licensed radio operator (KL4JC,
monitoring!), and consider myself an activist for the safety and inclusion of
Black, trans, and underrepresented folks in tech. I'm also a lifelong Eagle
Scout (whether the BSA wants "people like me" or not. ;)

Finally, a keyword–oriented list of additional related skills (with which I have
at least a passing familiarity) omitted from the above, follows:

    Techniques: BDD & TDD
    Languages & platforms: CoffeeScript, TypeScript, Flow; Rails, PHP;
        Objective–C, Cocoa; Io, Lua, Potion; Racket, Guile, other R5RSes; Tulip,
        Eff; VimScript
    Other: macOS, BSD, Linux, and the POSIX/UNIX APIs; Redis & MongoDB