

Vlad Kozin

JavaScript, OMeta, Racket, meta-programming
Fall'13 Recurse Center (aka Hacker School) alum

Corporate ladder

- Since 2014 *Programmer* at [Yandex](#) (Moscow, Russia).
Officially a member of *Search Interfaces Development Infrastructure* group, but mostly I write backend tools that perform source to source compilation: parse, transform and generate code. Any given project will inevitably depend on: [ometajs](#), [esprima](#), [estraverse](#), [uglify-js](#), [escodegen](#), [xjst](#). If I'm lucky and do it right frontend developers get to use my work and get all the credit.
- 2009-2011 *Equity Derivatives & Structured Products Sales* at [Renaissance Capital](#) (Moscow, Russia).
No programming of any kind unless you count Excel.
- 2007-2009 *EM Structured Solutions and Derivatives Sales* at [Barclays Capital](#) (London, UK).
No programming of any kind unless you count Excel and Powerpoint.

Code to frown upon

- Racket *Author* of [ometa-racket](#), a mostly complete Racket implementation of [OMeta](#) - OO pattern-matching language that extends PEGs with ability to handle left-recursive rules and match structured data.
Author of [skish](#), a mostly futile attempt at porting Olin Shivers' wonderful [scsh](#) to Racket. [scsh](#) is a non-interactive Unix shell embedded within Scheme (originally Scheme48).
- JavaScript *Author* of [bemhtml-syntax](#), a syntax converter for [BEMHTML](#) - an XSLT inspired templating language - part of [BEM methodology](#) of frontend development.
Author of [bemhtml-source-convert](#), a *best effort* compiler from [BEMHTML](#) templates to [BH](#) templates.
Author of [xjst-more](#), an [XJST](#)-based compiler for [BEMHTML](#) templates that facilitates incremental compilation of templates potentially on the Client. WIP.
Contributor to [ometajs](#), a JavaScript implementation of [OMeta](#).
Contributor to [bem-xjst](#), [XJST](#)-based compiler for [BEMHTML](#) templates.
- Emacs Lisp *Author* of [jslime](#), a minor mode that sends JavaScript code to Node.js repl - silly little thing with very few features that does make iterative JavaScript development in Emacs sane.
Contributor to [projectile](#), a tiny fix for [persp/projectile](#) when used with Helm.

Formal education

- 2004-2006 [Keldysh Institute of Applied Mathematics](#) (Moscow, Russia)
PhD track in Applied Mathematics, dropped out
- 1999-2004 [Lomonosov Moscow State University](#) (Moscow, Russia)
MS in Theoretical Mechanics and Applied Mathematics.

Autodidacticisms

- 2012 [How to Design Programs](#) by Matthias Felleisen et al.

2012	<p>How I was introduced to programming. Assorted solutions to HtDP.</p> <p>Programming Languages, [Certificate] Brown University</p> <p>How I was introduced to creating PLs. Taught by Shriram Krishnamurthi based on his wonderful PLAI text. My solutions - a sequence of interpreters for progressively more complex languages: all the way to OOP, CPS transforms and type checkers.</p>
2014	<p>Hardware/Software Interface, [Certificate 89.6%] University of Washington for Coursera</p> <p>How I was introduced to systems programming. Essentially an Introduction to Computer Systems course as taught at Carnegie Mellon with the same course-load and text Computer Systems: A Programmer's Perspective by Bryant and O'Hallaron.</p>
2014	<p>Paradigms of Computer Programming 1, [Certificate1 94%] Paradigms of Computer Programming 2, [Certificate2 97%] Université catholique de Louvain for edX</p> <p>How I was introduced to concurrency, multi-paradigm programming and delightful paradigms that so far seem to exist only in academic setting. Taught by Peter Van Roy and is based on his classical Concepts, Techniques, and Models of Computer Programming.</p>
2015	<p>Introduction to Probability, [Certificate 94%] MIT for edX</p> <p>Because it's awesome.</p>

Languages

Russian, English	Equally uncomfortable with both.
Racket	Would be my weapon of choice were such choice ever offered.
JavaScript	What I get to speak at work.
OMeta	What I'm fortunate to speak at work.
Emacs Lisp	Unavoidable.
C	Enough to pass the systems programming class.
Factor, OCaml, F#	Toyed with but never mastered in earnest.

Activities and interests

Most of my activities and interests these days involve boxes with lights and buttons. Even so there were reports of me cycling, bouldering, surfing, roller-skating, skiing and more. Having owned a sports car I'll choose a bicycle every time.

Lived in the UK, US, Hungary and far more exotic places. Crossed the US from Mexico to Canada twice with the current state count of 19.

NB: unless remote is ok I'd probably need a visa to work in your country

vladilen.kozin@gmail.com • +7 916 8100356 • Moscow, Russia
[pdf version](#) • [txt version](#) • [doc version](#) • [html version](#)