# Vlad Kozin

**JavaScript**, **OMeta**, **Racket**, **meta-programming**  
Fall'13 [Recurse Center](https://www.recurse.com/) (aka Hacker School) alum

## Corporate ladder

Since 2014

*Programmer* at [Yandex](https://company.yandex.com/) (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](https://github.com/veged/ometa-js), [esprima](http://esprima.org/), [estraverse](https://github.com/estools/estraverse), [uglify-js](https://github.com/mishoo/UglifyJS), [escodegen](https://github.com/estools/escodegen), [xjst](https://github.com/veged/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](https://www.rencap.com/InvestmentBanking/About/) (Moscow, Russia).

No programming of any kind unless you count Excel.

2007-2009

*EM Structured Solutions and Derivatives Sales* at [Barclays Capital](http://investmentbank.barclays.com/markets.html) (London, UK).

No programming of any kind unless you count Excel and Powerpoint.

## Code to frown upon

Racket

*Author* of [ometa-racket](https://github.com/vkz/ometa-racket), a mostly complete Racket implementation of [OMeta](http://en.wikipedia.org/wiki/OMeta) - OO pattern-matching language that extends PEGs with ability to handle left-recursive rules and match structured data.

*Author* of [skish](https://github.com/vkz/skish), a mostly futile attempt at porting Olin Shivers' wonderful [scsh](http://scsh.net/) to Racket. scsh is a non-interactive Unix shell embedded within Scheme (originally Scheme48).

JavaScript

*Author* of [bemhtml-syntax](https://github.com/vkz/bemhtml-syntax), a syntax converter for [BEMHTML](https://en.bem.info/) - an XSLT inspired templating language - part of [BEM methodology](https://en.bem.info/method/) of frontend development.

*Author* of [bemhtml-source-convert](https://github.com/vkz/bemhtml-source-convert), a *best effort* compiler from [BEMHTML](https://en.bem.info/) templates to [BH](https://github.com/bem/bh) templates.

*Author* of [xjst-more](https://github.com/vkz/xjst-more), an [XJST](https://github.com/veged/xjst)-based compiler for BEMHTML templates that facilitates incremental compilation of templates potentially on the Client. WIP.

*Contributor* to [ometa-js](https://github.com/veged/ometa-js), a JavaScript implementation of [OMeta](http://en.wikipedia.org/wiki/OMeta).

*Contributor* to [bem-xjst](https://github.com/bem/bem-xjst), XJST-based compiler for BEMHTML templates.

Emacs Lisp

*Author* of [jslime](https://github.com/vkz/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](https://github.com/bbatsov/projectile), a tiny fix for persp/projectile when used with Helm.

## Formal education

2004–2006

[Keldysh Institute of Applied Mathematics](http://keldysh.ru/index.en.shtml) (Moscow, Russia)  
 *PhD track in Applied Mathematics, dropped out*

1999-2004

[Lomonosov Moscow State University](http://www.msu.ru/en/info/struct/depts/mechmath.html) (Moscow, Russia)  
*MS in Theoretical Mechanics and Applied Mathematics.*

## Autodidacticisms

2012

[How to Design Programs](http://www.ccs.neu.edu/home/matthias/HtDP2e/index.html) by Matthias Felleisen et al.

How I was introduced to programming. [Assorted solutions to HtDP](https://github.com/vkz/HtDP).

2012

[Programming Languages](http://cs.brown.edu/courses/cs173/2012/), [[Certificate](http://cs.brown.edu/courses/cs173/2012/OnLine/Certification/687898716/)]  
Brown University

How I was introduced to creating PLs. Taught by [Shriram Krishnamurthi](http://cs.brown.edu/~sk/) based on his wonderful [PLAI](http://cs.brown.edu/courses/cs173/2012/book/) text. [My solutions](https://github.com/vkz/PLAI) - a sequence of interpreters for progressively more complex languages: all the way to OOP, CPS transforms and type checkers.

2014

[Hardware/Software Interface](https://www.coursera.org/course/hwswinterface), [[Certificate](http://bit.ly/1FBVMks) 89.6%]  
University of Washington for Coursera

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](http://csapp.cs.cmu.edu/) by Bryant and O’Hallaron.

2014

[Paradigms of Computer Programming 1](https://www.edx.org/course/paradigms-computer-programming-louvainx-louv1-1x-0), [[Certificate1](http://bit.ly/1F2svxU) 94%]  
[Paradigms of Computer Programming 2](https://www.edx.org/course/paradigms-computer-programming-louvainx-louv1-2x-0), [[Certificate2](http://bit.ly/1cEu4Lu) 97%]  
Université catholique de Louvain for edX

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](https://www.info.ucl.ac.be/~pvr/cvvanroy.html) and is based on his classical [Concepts, Techniques, and Models of Computer Programming](https://www.info.ucl.ac.be/~pvr/book).

2015

[Introduction to Probability](https://www.edx.org/course/introduction-probability-science-mitx-6-041x-0), [[Certificate](http://bit.ly/1FHykkR) 94%]  
MIT for edX

Because it's awesome.

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