Vlad Kozin

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

	Corporate ladder
Since Apr 2017	Programmer at Droit (London, UK) Same as before but with obligatory daily commute.
2015-2017	Programmer/Consultant at Droit (New York, USA - remote)
	Building an expert system for compliant trading. Sneaking Clojure/ClojureScript into unsuspecting financial glants. On any given day I could be designing DSLs, implementing compilers, parsers, rule-based engines, putting together simple browser-based GUIs and whatever else the startup life would have me do.
2014-2015	Programmer at Yandes (Moscow, Russia). Officially a member of Search Interface Development Infrastructure group, but mostly I will backent look that perform source to source compilation; parse, transform and generate code. Any given project will intervisible depend on: meeting, eportime, estress, upility-is; escode, gen, xiel. II i'm lucky and do it right frontend developers get to use my work and ge at the credit.
2009-2011 2007-2009	Equity Derivatives & Structured Products Sales at Renaissance Capital (Moscow, Russia). EM Structured Solutions and Derivatives Sales at Barclays Capital (London, UK).
	Projects
Racket	Author of ometa-racket, a mostly complete Racket implementation of OMeta - OD 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' wonderfulsosh to Racket. scsh is a non-interactive Unix shell embedded within Scheme (originally Scheme48).
Clojure JavaScript	Author of several closed-source products: FpML message parser, financial derivatives classifier based on ISDA taxonomies, legal annotation tools, etc. Author of benthml-syntax, a syntax converter for BENHTML - an XSLT inspired templating
SavaScript	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 xist-more, an XJST-based compiler for BEMHTML templates that facilitates incremental compilation of templates potentially on the Client. WIP.
	Contributor to ometa-js, a JavaScript implementation of OMeta. Contributor to bem-xist, XJST-based compiler for BEMHTML templates.
Emacs Lisp	Author of jslime, a minor mode that sends JavaScript code to Node is repl - silly little thing with very few features that does make iterative JavaScript development in Emacs sane.
	Contributor to whatever little thing I find annoyingly broken in my daily Emacs use. Could bug reports or tiny fixes to upstream packages.
	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	How I was introduced to programming. Assorted solutions to HtDP. Programming Languages, [Certificate]
	Brown University
	Brown University 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
2014	Brown University How I was introduced to creating PLs. Taught by Shriram Krishnamurthi based on his wonderful PLN lext. My solutions - a sequence of interpreters for progressively more complex languages: all the way to ODP, OPS transforms and type checkers. Hardware/SORIVAR interface, Inc. Exitificate 89.8%I
2014	Brown University How I was introduced to creating PLs, Taught by Shriram Krishmanurthi based on his wonderful PLA text. My solutions - a sequence of interpreters for progressively more complex languages at the way to GOR, PS transforms and type checkers, Hardware Software Interface, Derifficate 88.8% University of Wishman for Courses How I was introduced to systems powering. Essentially an Infroduction to Computer Systems course as tought at Compage Marriam with the same course-load and text Computer
	Brown University How I was introduced to creating PLs. Taught by Shirram Kichmanurthi based on Nis wonderful PLN text. In youthings — sequence of interpreters for progressively more complex tanguages at the way to COP, PS transforms and type checkens. Hardware Schware Interface, Derifficials 88.0%] University of Whardpring for Courses How I was introduced to opsteme programming. Essentially an Introduction to Computer Systems course as laught at Carnege Median with the same conservation of the Computer Systems course be subject of Computer Systems course be subject of Computer Systems course bearing at Carnege Median with the same conservation of the Computer Systems Course bearing at Computer Systems Course bearing at Computer Programming 1, Exciticates 19.1% In Proceedings of Computer Programming 1, Exciticates 19.1% In Proceedings of Computer Programming 1, Expertitions of Computer Systems Course Section (Edition 19.1%).
	Brown University How I was introduced to creating PLs. Taught by Shriram Krishnamurthi based on his wonderful PU I see. It has been a sequence of interpreters for progressively more complex inarguages at the way to DCP. Post inardoms and type Cerchoders. Hardware Software Interface, Derifficial 80.056] Hardware Software Interface, Derifficial 80.056] New I was introduced to systems programing, Essentially an Introduction to Computer Systems course as taught at Clarange Melton with the same course-load and text Computer Systems course as taught at Clarange Melton with the same course-load and text Computer Systems. A Programm's Perspective by Systems and Oristations of Computer Programming 1, Escriticates 91% Paradigms of Computer 1, 100% Pa
2014	Brown University How I was introduced to creating PLa. Taught by Shriram Krishnamurthi based on his worderful PLN last. My Solutions – a sequence of interpreters for progressively more worderful PLN last. My Solutions – a sequence of interpreters for progressively more land to the progressive of the progressive of the progressive of the PLN last introduction. Destinate all Sections (Section My Vision I was a sequence of the PLN last introduction to Computer Systems course as taught at Camege Melton with the same course-load and lead Compute Systems course as taught at Camege Melton with the same course-load and lead Compute Systems. A Progressive Play System and O'Faldaccian Progressive Order and Camege Melton (Section I was a progressive of Camege Progressive Order and Camege Melton (Section I was a progressive Order and Camege Progressive Ord
2014	Brown University How I was introduced to creating PLs. Taught by Shriram Krishnamurthi based on his wonderful PUN text. It by solutions –a sequence of interpreters for progressively more complex languages at the way to COP. O'Ps brandform and type forchedens. Hardware Schlaws Interface, Sectificate 88.0H; Never I was introduced to optient programming, Essentially an introduction to Computer Systems course as taught at Cornega Mellow with the same course-load and text Computer Systems can course be sought at Cornega Mellow with the same course-load and text Computer Systems. An Opportune? Perspective by Systems and Orbitation Programming 1, Sectificate 97.9[Université catholique de Louvain for atX Hardware Cornega (Post Cornega Mellow of the State Cornega Mellow of the State Cornega Mellow of the State Cornega (Post Cornega Mellow of the State Cornega Mellow of the State Cornega (Post Cornega Mellow of the State Cornega Mellow of the State Oromopter, Textringues, and Models of Computer Programming, Introduction to Problemble, Certificates 9449] Mill for ed.X
2014	Brown University How I was introduced to creating PLs. Taught by Shriram Krishmanurthi based on his wonderful PU I set. It his youthrist -a sequence of interpreters for progressively more complex languages at the way to 0.00°, PC brandforms on type forchedren. Hardware Schlares (Particus, Partificiale 98.0%) Hardware Schlares (Particus, Partificiale 98.0%) Hardware Schlares (Particus, Partificiale 98.0%) Particus (Particus, Partificiale 98.0%) Particus (Particus, Particus, Partificiale 98.9%) Paradigms of Computer Programming 1, Partificiale 98.9%) Paradigms of Computer Programming 1, Partificiale 98.9% Paradigms of Computer Programming 1, Partificiale 98.9% Université Earthique de Louvair for etX. How I was introduced to concurrency, multi-paradigm programming and eligitatis and the particus of this classical Conceptis, Techniques, and Models of Computer Programming. Introduction to Probability, (Partificiale 9849) Mil' for edX. Mil' of edicing (Particus) Recuse it is assected.
2014	Brown University How I was introduced to creating PLs. Taught by Shriram Krishnamurthi based on his wonderful PLN text. My solutions – a sequence of interpreters for progressively more complete languages and the vey to COP. O'D structions and type checkens. University of Washington for Ouzersa How I was introduced to systems programming. Essentially an introduction to Computer Systems course as taught at Camege Melton with the same course-load and text Compute Systems. An Organizaria Perspective by Dipart and O'Halamon of Computer Programming 1, Escriticated 19/6) Paradigment of Computer Programming 1, Escriticated 19/6 (Industrial Systems). An Organizaria Perspective by Dipart and O'Halamon of Computer Programming 1, Escriticated 19/6 (Industrial Systems). An Organizaria Systems of Computer Programming 1, Escriticated 19/6 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1 (Industrial Systems). An Organizaria Systems of Computer Programming 1
2014 2015 2017	Brown University How I was introduced to creating PLa. Taught by Sinstram Krishmanuthi based on his wonderful PLA text. by Solutions – a sequence of interpreters by organized who wonderful PLA text. by Solutions – a sequence of interpreters by organized who wonderful PLA text. by Solutions and type checkens. Hardware Software latentians, Carefiticals and Solutions and type checkens. University of Washington for Courses How I was introduced to bystems programming. Essentially an introduction to Computer Systems course as taught at Camega Mellon with the same course-load and text Compute Systems course as taught at Camega Mellon with the same course-load and text Compute Paradigm of Computer Programming 2, Cartificated 94%; How I was introduced to concurrency, mill-paradigm programming and designful introduction to Probability, Cartificates 94%; Mill of the Camega Computer Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Computer Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Computer Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Computer Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Computer Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the Camega Camega, American Camega, and Models of Computer Programming, Introduction to Probability, Cartificates 94%; Mill of the
2014 2015 2017 Russian, English	Brown University How I was introduced to creating PLa. Taught by Sintram Krishnamurthi based on his wonderful PLA I text. In y Solutions - a sequence of interpreters by progressively more wonderful PLA I text. In y Solutions - a sequence of interpreters by progressively more wonderful PLA I text. In y Solutions - a sequence of interpreters by progressively more throughout the place of the progressive of the place of
2014 2015 2017 Russian, English	Brown University How I was introduced to creating PLa. Taught by Shriram Krishnamurthi based on his wonderful PU I see. It his youthors – a sequence of interpreters for progressively more comprete languages at the way to DCP. Por brandform and type chockens. Hatel was follows the fair the way to DCP. Por brandform and type chockens. Hatel was follows the fair the way to DCP. Por brandform and type chockens. Hatel was introduced to optime programming, Essentially an Introduction to Computer Systems course as taught at Clarages Melton with the same course-load and text Computer Systems course as taught at Clarages Melton with the same course-load and text Computer Systems. An Opportunity Programming 1, Escriticated 97%] Paradigms of Computer Programming 1, Escriticated 97%] Paradigms of Computer Programming 1, Escriticated 97%] How I was introduced to concurrency, multi-paradigm programming and delegibit. How I was introduced to concurrency, multi-paradigm programming and delegibit. Mill of each of the desired of the programming of the section of the consistency of the programming introduction to Problemisity, Certificates 94%] Mill of ed. Our deleging operational semantics. The Rucket Summer School of Semantics and Languages in Salt Lake City, Utah. White targeted at Pt. Pitch a burner for universal processing semantics - what's no take? Languages Equally connected table. Equally connected table. Equally connected table. Equally connected table.
2014 2015 2017 Plussian, English Clojure Racket	Brown University How I was introduced to creating PLa. Taught by Sinstram Krishmanurthi based on his wonderful PLA I text. In y Solutions — a sequence of interpreters by opposessively more wonderful PLA I text. In y Solutions — a sequence of interpreters by opposessively more wonderful PLA I text. In y Solutions — a sequence of interpreters by opposessively more horizontal plant of the PLA I solutions of the
Russian, English Clojure Racket JavaScript OMeta Emacs Lisp	Brown University How I was introduced to creating PLa. Taught by Shriram Krishnamurthi based on his worderful PLN last. My Solutions - a sequence of interpreters for progressively more worderful PLN last. My Solutions - a sequence of interpreters for progressively more worderful PLN last. My Solutions - a sequence of interpreters for progressively more last the progressive of the progressive of the progressive of the PLN last last last last last last last last
2015 2017 2017 2017 2017 2017 2017 2017 2017	Brown University How I was introduced to creating PLa. Taught by Sinstram Krishmanurthi based on his wonderful PLA I text. It by Solutions - a sequence of interpreters by opposessively more wonderful. PLA I text. It by Solutions - a sequence of interpreters by opposessively more wonderful. PLA I text. It by Solutions - a sequence of interpreters by opposessively more wonderful. Plant I text. I see that the solution was they developed. University of Washington for Courses How I was introduced to bystems programming. Essentially an introduction to Computer Systems course as taught at Camege Melton with the same course-load and text Compute Paradigms of Computer Programming 2. Certificated 94%] Paradigms of Computer Programming 2. Expertitions 94%; How I was introduced to concurrency, mill-paradigm programming and delightful in text. I was introduced to concurrency, mill-paradigm programming and delightful in its based on his classical Concepts, Techniques, and Models of Computer Programming. Introduction to Probability, Expertitional 94%; Because its awerone. Secure of sequency and a security of the security of t
2014 2015 2017 Russian, English Clojum Racket JavaScript OMeta	Brown University How I was introduced to creating PLa. Taught by Shriram Krishnamurthi based on his worderful PLN last. My Solutions - a sequence of interpreters for progressively more worderful PLN last. My Solutions - a sequence of interpreters for progressively more worderful PLN last. My Solutions - a sequence of interpreters for progressively more last the progressive of the progressive of the progressive of the PLN last last last last last last last last

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 carl'Il choose a betyder every time. Hungary, Spain and far more exotic places. Crossed the US from Mexico to Canada twice with the current state court of 19.

vladilen.kozin@gmail.com • +44 7494979 628 • London, UK pdf version • txt version • doc version • html version