

Skills: Node.js, SQL, Chrome extensions development, browsing automation (Puppeteer), React, HTTP, WebSocket; JavaScript, HTML, CSS; No-code (bubble.io), Linux, Git

Areas of programming where my skills are most developed:

core JavaScript, DOM manipulation, creating user interfaces, general problem solving, coming up with and implementing algorithms when necessary

Experience

Lecturer @ Tel-Ran.de GmbH (Dec 2021 – Feb 2022)

I developed and taught a course in web programming (front-end)

Node.js developer @ Intego-press (May 2019 – Oct 2019)

I developed a data post-processing system for video content analysis (**Node.js, SQL, git, bitbucket**)

Web applications developer @ self-employed (2017 – Present)

Full list of 20+ projects: iloveprogramming.org/portfolio

Selected projects:

1. "doit.js" — full-stack framework and interactive development environment. Inspired by Smalltalk environment and bubble.io platform. (**Node.js, SQL, JavaScript / HTML / CSS**)
2. Chrome extension "Read aloud with auto language detection" — text-to-speech web reader. 1000+ users from 60+ countries. (**JavaScript / HTML / CSS, Chrome extensions API**)
3. Chrome extension "Code editor connector". Connect any web-based code editor to any opened website and execute any code selections in this websites's console. (**JavaScript, Chrome extensions API, WebSocket**)
4. Multi-user web apps (**bubble.io, JavaScript**):
"Econtrainer" — easily create training problem sets using advanced filters.
"Econquestions" — questions and answers platform which helps teachers optimally decide who should answer this or that question.
Apps were used by a team of teachers and 100+ students at Shkolkovo online school.
5. Song player with special functionality that facilitates learning to sing and play (**JavaScript / HTML / CSS, File and Directory Entries API**)
6. Chat app (**Node.js, SQL, WebSocket, JavaScript / HTML / CSS**)
7. Browsing automation, monitoring web pages and sending notifications (**Node.js, Puppeteer, JavaScript / HTML / CSS**)
8. Parsing data from web pages, generating analytics tables (**Node.js, Puppeteer, SQL, JavaScript / HTML / CSS**)
9. Tool for persistent live page editing (**IndexedDB, JavaScript / HTML / CSS**)
10. "Pick" — a simple query language for object trees (**JavaScript**)

Certification of web development skills

- LinkedIn Skill Assessment: **JavaScript** (top 5%), **HTML** (top 5%), **CSS** (top 15%)
- Angellist Assessment: **Full-Stack Web** (top 10-20%)
- Codesignal.com **Node.js** certificate: Coding Score = 843, level 10 out of 10

Certification of problem-solving skills (coding in JavaScript)

- Codesignal.com 2018: Coding Score = 731 (reference values: Oracle employees = 728, Amazon employees = 733)
- Codesignal.com 2021: Coding Score = 784, level 8 out of 10

Education

Master of Arts in Economics, New Economic School

Bachelor of Arts in Economics (with honors), Higher School of Economics

Honors and awards

- Winner of the National Olympiad in Economics
- New Economic School alumni grant for visiting US universities

Talk: [Breaking the Wall of Lie using Trust Network](#) (A.T. Kearney's Falling Walls Lab)

Motivation and programming style

Programming philosophy:

I try to avoid unnecessary complexity; I like to use beautifully designed systems where simplicity and wisely chosen constraints make you powerful and help you achieve goals with less code.

I like dynamic and highly interactive development.

I like to learn simple tools and combine them creatively.

I like systems that make it easy to modify them, so we can maintain the system in a regret-free condition.

Programming style:

interactive development with frequent use of the console and immediate testing. The result is that typically all the errors are detected and fixed during the initial testing.

What makes my code readable and maintainable:

coming up with good names for variables; following "don't repeat yourself" principle (avoiding code duplication).

Why do I love programming:

In programming you can combine creative freedom and logical thinking to create a real product that you can use; you can automate things to avoid dull repetitions; you can use systems that work consistently each time you run them; in a programming language you can simultaneously be succinct and precise up to details by using abstractions.

Direction of progress:

I have interest in the skill and technology of rapid application development, quickly implementing experimental features and minimum viable products. I also want to learn how to manage complexity and large codebases, how to work with a variety of different technologies, and the art of how to be productive in a team.

Links:

[Personal website](#)

[My blog on programming technologies](#)

[Latest version of this document](#)