

Brit Butler

CONTACT INFORMATION

404-718-9378
866 Woodland Ave SE
Atlanta, GA 30316

E-mail: brit@kingcons.io
Code: <https://github.com/kingcons/>
Web: <https://blog.kingcons.io/>

WORK EXPERIENCE **October 2018 – Present**

Lead Instructor

Flatiron School

I taught 80+ full-time students Ruby/Rails and React while managing and mentoring TAs. I developed bonus lectures and additional curriculum on unit testing, test coverage, and refactoring. I participated in internal discussions to improve our assessments and developed custom plans for struggling students.

June 2017 – October 2018

Senior Software Engineer

Showcase IDX

I took over Backend duties and DevOps for a Real Estate Search startup powering over 1000 websites and helped scale the system to 1M+ page views a day. Major initiatives included moving the postgres instance from kubernetes to a dedicated Cloud SQL instance, writing a migration tool for customers on the legacy version of the product, improving the ingestion of data from MLS listing feeds, rebuilding the Elasticsearch cluster after a canary deploy went wrong, optimizing queries on our 30M+ record leads table, and other assorted production work. I did some “Programmer Archaeology” to relearn how to deploy the legacy system and switched it to using LetsEncrypt for SSL. Since I was the sole ops engineer, I thoroughly documented the architecture, its performance characteristics, and some gotchas before leaving.

December 2014 – May 2017

Lead Instructor

The Iron Yard

I taught immersive, full-time courses in Backend Engineering using Ruby/Rails and Frontend Engineering using Javascript/Angular. I was promoted to Lead Instructor after 6 months. As an instructor, I iterated on curriculum, lectured, graded assignments, and worked 1-on-1 with students during lab time. As a lead, I mentored a dozen new instructors at different campuses and served as a resource for classroom issues, struggling students, and other problems.

May 2013 – October 2014

Software Engineer

Emcien

I worked on several data analysis products written in Ruby/Rails and C. I added allocation tracking machinery to a modern 20k SLOC C project to aid in finding memory leaks and reducing the overall memory footprint. I took over maintenance of a legacy product, Mix, migrating from Ruby 1.8.7 to 1.9.3 and overseeing numerous point releases. I also contributed substantial work to the primary product, Patterns, including the report download builders and storing report attributes in SQL shards.

May 2011 – September 2012

Software Engineer

CMGdigital

I worked on a 160k SLOC Python/Django project to serve Newspaper, TV, and Radio publishers. I had particular focuses on admin customization and data migration. I oversaw the migration of over 100K users from multiple markets to our new CMS. I made broad improvements to our Brightcove video import scripts using celery and memcached. I also gave talks on “[Programmer Archaeology](#)” and [the Economics of Open Source](#).

EDUCATION

January 2009 – May 2011

B.Sc. in Computer Science

Southern Polytechnic State University

I transferred to SPSU in August 2007 to pursue Computer Science. After the first semester, I took a year off to work full-time and self-study. I returned in January 2009 and graduated in May 2011.

January 2008 – August 2008

Self-Study

Southern Polytechnic State University

After 3 years at Oglethorpe University studying literature I decided to self-study programming while working full-time. I did this throughout 2008 and have written about it, notably [here](#) and [here](#). Example work from self-study can be found [here](#).

July 2018 – Present**Author**[study-group](#)

I have organized and led an [SICP](#) study group for myself and former coworkers and students. In addition to providing code feedback, I prepare the schedule, a meeting location, and host weekly discussion both online and in person.

August 2016 – September 2016**Author**[nescavation](#)

A NES simulator written during a semester break to practice a bigger project using ES6 classes and modules. I stalled out working on the graphics card when classes resumed. However, the light exposure to Canvas and requestAnimationFrame was valuable and I reused the decomposition of CPU opcode metadata from instruction definitions in a more recent hobby project called “clones”.

November 2014 – Present**Author**[salty-runbooks](#)

A set of Ansible playbooks to better familiarize myself with configuration management and help automate the setup and administration of my personal servers. It includes roles for retrieving SSL certificates with LetsEncrypt, setting up Postfix as a mail server, running an IRC bouncer, a blog generated with Lisp and served by nginx, and various web applications written in PHP, Python, and Node for streaming music and hosting media.

August 2012 – November 2014**Author**[coleslaw](#)

coleslaw is static blogware a la Jekyll, written in Common Lisp. It supports publishing via git push, markdown with code highlighting extensions, extensible content types, theming, and various functionality through plugins all in under 1000 lines of code. It is also a good example of Object Oriented Programming in Lisp. Since “you are what you document” there is a thorough [Hacker’s Guide to Coleslaw](#) explaining its internal design in addition to documentation of the Plugin API and extensions.

May 2011 – May 2014**Author**[cl-6502](#)

cl-6502 is a MOS 6502 emulator, assembler, and disassembler written in Common Lisp. Inspired by Luke Gorrie’s call for “Readable Programs” there is an [annotated book](#) of the source code. Some of the motivations behind cl-6502’s creation are described [here](#). There is also a recorded talk on the project and related ideas called “[On Programmer Archaeology](#)”.

Contributor

Miscellaneous

I have also contributed feature, portability, and documentation patches to:

- Spacemacs, A community driven Emacs distribution
- Vacietis, a C to Common Lisp compiler
- st-json, a Common Lisp JSON library
- kardboard, a virtual Kanban web app made with Python and Flask
- pybrightcove, a Python interface to Brightcove’s API

SKILLS

Programming Languages (intermediate): Common Lisp, Python, Ruby, Javascript

Programming Languages (novice): C, Scheme, Haskell, Factor, Elisp, Clojure

Markup Languages: HTML, CSS, SASS, LaTeX

Operating Systems: Mac OS X, Windows 3.11-7, Various Linux distributions esp. Debian, Archlinux, Guix

Preferred Tools: Emacs, Git, Steel Bank Common Lisp, Debian

INTERESTS

When I’m not in front of my computer, I like to make cocktails, spend time with my partner and our goofy dogs, play Smash Brothers Melee, and noodle on a modular synth. As far as Comp Sci topics, I am fascinated by the implementation of dynamic, reflective languages such as Lisp and Smalltalk and the careful interplay between compiler and runtime to make them fast.

REFERENCES

Available on request.