

# Will Pringle

[willkantorpringle@gmail.com](mailto:willkantorpringle@gmail.com) | [linkedin.com/in/will-pringle](https://www.linkedin.com/in/will-pringle) | <https://github.com/wiwichips> | +1(519)994-0408

## TECHNICAL SKILLS

**Languages:** JavaScript, Python, Rust, C, Go/Golang, Shell/Bash, Java, SQL, GraphQL

**Frameworks and Libraries:** Node.js, Express.js, React, Apollo, Django, Tensorflow, Pandas, NumPy

**Developer Tools:** Git, GitHub / GitHub Actions, Docker, AWS, Android Studio, Datadog, Kubernetes

## WORK EXPERIENCE

### Lead API & Toolchains Developer

June 2024 – Present

*Distributive*

Kingston, ON

- Designed and built a Webhook server in **Go** for accessing **AI model inferences** deployed to [DIANA](#)
- Developed an API which enables users to specify execution environments in distributed compute nodes
- Created a runtime for distributed **Python** execution on [DCP](#) using **WebAssembly** and **Pyodide**
- Designed and implemented the **Python** SDK for DCP ([Source code on GitHub](#))
- Planned Quarterly OKRs, mentored developers, and scoped MVPs for API and Toolchain products for DCP

### Software Developer

May 2023 – June 2024

*Distributive*

Kingston, ON

- Developed a versatile **REST API** integrated with **OAuth2.0** using **NodeJS** and **Express**, broadening DCP's accessibility across various programming languages and enhancing platform usability
- Implemented OAuth2.0-based SDK authorization for enhanced security and convenience for SDK users
- Created a Kubernetes **manifest.yaml** generator to configure and deploy DCP compute clusters efficiently
- Promoted the open source project [PythonMonkey](#) resulting in 300+ stars on GitHub in 1 month (800+ now)
  - \* Designed in **Figma** and implemented the [pythonmonkey.io](#) landing page using **HTML**, **JS** and **CSS**
  - \* Created [videos](#), [articles](#), and met with journalists (Featured in [The New Stack](#), Node Weekly, JS Weekly)

### Software Engineer Intern (Backend)

May 2022 – August 2022

*Arctic Wolf*

Waterloo, ON

- Implemented a feature in **Go** that removes data across a distributed system by sending **Kafka** and **RabbitMQ** messages in order to automate the deactivation of security vulnerability scanners
- Wrote a **Python** script to automate deactivating scanners, speeding up the originally manual process substantially

### Software Engineer Intern (API)

September 2021 – December 2021

*Wave Financial Inc.*

Toronto, ON

- Implemented API request caching using [Automatic Persisted Queries](#) for over 140 **GraphQL** query **HTTP** requests reducing inbound network traffic by up to 93% for queries requested millions of times a month
- Helped increase **GraphQL** literacy by contributing to an internal training project and presenting at demos

### Software Developer Intern (Fullstack)

January 2021 – August 2021

*Distributive (Formerly known as Kings Distributed Systems)*

Kingston, ON

- Implemented a scraping feature for payments, enabling a third party to earn commissions on paid compute
- Fixed race condition bugs in an in-house network protocol affecting session ID generation improving security
- Re-branded a frontend dashboard used for a research study with 1000+ participants using **CSS**

### Research Analyst Intern

June 2020 – December 2020

*Ontario Ministry of Finance*

Toronto, ON

- Automated data entry and modelling using **Python**, reducing time for a critical monthly task by 99%
- Wrote a package in **Python** and **R** that helps economists analyze data that is used to write Ontario's tax policies

## EDUCATION

### University of Guelph

September 2018 – May 2023

*Bachelor of Computing, Computer Science, Area of Application in Business Economics*

Guelph, ON

- 3.7 GPA (86% cumulative average)
- VP Event Planner for the **Society of Computing and Information Science**
- Ran and organized the **Guelph Programming Competition** for competitive programming

## PROJECTS

### ImgProBox([Try it out](#))

[github.com/wiwichips/imgprobox](https://github.com/wiwichips/imgprobox)

- Image processing toolbox in the browser using **Rust** compiled to **WebAssembly** for the core logic and **React**
- Able to efficiently processes a live webcam feed or allow for images to be uploaded