

Python Code Origins and Why It's Good for AI

Before we had ChatGPT, before we had self-driving cars, before AI became a household word — there was Python. And the story of how a quirky language named after a British comedy show became the backbone of modern artificial intelligence is one worth telling.

Where Python Came From

Python was born in the late 1980s inside a research lab in the Netherlands. Guido van Rossum, a Dutch programmer at the **National Research Institute for Mathematics and Computer Science (CWI)**, was frustrated with the tools available at the time. Languages like C were powerful but notoriously hard to write and read. Shell scripting was flexible but limited. He wanted something in between — elegant, readable, and practical.

His solution was Python. The **first open-source release came in February 1991**, and it was unlike anything else at the time. Code that would take dozens of lines in C could be expressed in just a few lines of Python, and a newcomer could read it almost like plain English.

Fun Fact

The name has nothing to do with snakes. Van Rossum was a fan of the British comedy series *Monty Python's Flying Circus*, and he wanted a name that was short, slightly irreverent, and memorable. The snake logo came later.

The guiding philosophy behind Python was captured in a short document called “**The Zen of Python**” — a set of 19 aphorisms that includes lines like “*Readability counts*” and “*Simple is better than complex.*” These weren't just catchy phrases. They shaped how the language evolved for decades.

Why Python Became AI's Language of Choice

Fast-forward to today, and **Python is the undisputed dominant language in AI and machine learning**. That didn't happen by accident. Here's why it became the go-to tool for researchers and engineers building the most sophisticated software in the world:

It Reads Like English

Python's syntax is clean and intuitive. A researcher with a brilliant idea for a neural network can translate that idea into working code faster in Python than in almost any other language. Less time fighting syntax means more time solving real problems. When you're trying to push the boundaries of what's possible in AI, that matters enormously.

The Ecosystem Is Unmatched

Python's real superpower is its library ecosystem. Tools like **TensorFlow** and **PyTorch** let developers build and train neural networks with just a few lines of code. **scikit-learn** provides ready-made machine learning algorithms. **Pandas** handles data manipulation. **NumPy** powers fast numerical computation. These aren't just conveniences — they represent thousands of hours of engineering work that any developer can use for free, instantly.

Write Once, Run Anywhere

Python is highly portable. A model developed on a Windows laptop can be deployed on a Linux-based cloud server with virtually no changes. In the AI world, where training often happens on local machines but inference runs in the cloud, this flexibility is invaluable.

A Community That Solves Problems Fast

Python has one of the largest and most active developer communities on the planet. Stuck on a problem? There's almost certainly a Stack Overflow thread, a GitHub repo, or a tutorial that addresses it. For AI teams under pressure to ship, this community support acts like a massive, always-available help desk.

The Best of Both Worlds

Here's something most non-developers don't know: under the hood, many of Python's most important AI libraries are actually written in C or C++. Python acts as a friendly interface to incredibly fast low-level code. You get the speed of C with the simplicity of Python — a combination that makes it uniquely suited for computationally intensive AI workloads.

The Bottom Line

Python wasn't designed for AI. It was designed to be a better programming language — simpler, more readable, and more humane. But those same qualities turned out to be exactly what AI researchers needed. When the AI boom arrived, Python was ready. Its

ecosystem had grown, its community was massive, and its philosophy matched the fast-moving, experimental nature of AI research.

The next time you interact with an AI assistant, ask for a route on a navigation app, or see a product recommendation online — there's a good chance Python is somewhere in that stack. Not because it was planned that way, but because good design has a way of finding its purpose.

Original article: <https://tonyherman.com/python-code-origins-and-ai/>

PDF version: https://tonyherman.com/python-code-origins-and-ai/?apa_pdf=1

Special Offer for Readers

1,000+ Channels • On-Demand Movies • **5 Devices**

\$69.99/mo (~\$2.33/day)

Try it for a month and see if you like it, then switch.

Start Your Trial

Click to learn more about CUE Broadcast

Tip: Get 3 friends or family to sign up and you get it for free.