

Quickstart Python

This is a new attempt to write a quick introduction to Python (Python 3). Current content includes a basic intro chapter.

- [intro](#)

There are several math-related chapters.

- [factors](#)
- [primes](#)

From this point, the code segments get a bit longer.

I freely admit I find it very hard to read other people's code, and almost never do so. And yet these examples have non-trivial amounts of code: 50 or 100 *loc* (lines of code).

My suggestion is that the value in this repo is the problem statements, understand these, and then try to write your own code. My solutions can serve as a guide if you run into difficulty.

The first shows how to compute the square root of 2 by successive approximation. Actually, this approach can be generalized to any function that returns an irrational number.

Next is a much better way to calculate the square root of 2, due in part to the ancient Babylonians and partly to Isaac Newton.

Finally, we calculate the logarithm of a number to a given base. In the example, the base is 2, but the code could be easily modified to any base. However, because of the simple formula for changing the base of a logarithm, a general solution really doesn't add anything.

- [successive approximation](#)
- [square root of 2](#)
- [logarithm](#)

We can also exercise the Genetic Code. AI also wrote a short chapter on sorting. I love sorting.

- [wordle](#)
- [genetic code](#)
- [sorting](#)

There is a discussion of Caesar ciphers, a write-up about the math for public key crypto (original version based on Euler's totient function), and a demonstration of proof-of-work as it's used in Bitcoin.

- [ciphers](#)
- [public key math](#)
- [proof-of-work](#)

Wordle is a very popular word game on the web (Jan 2022). The first writeup is about a "cracker" for Wordle. At the end of this README, you will find a Python script for my version of Wordle that runs from the command line.

- [Wordle](#)

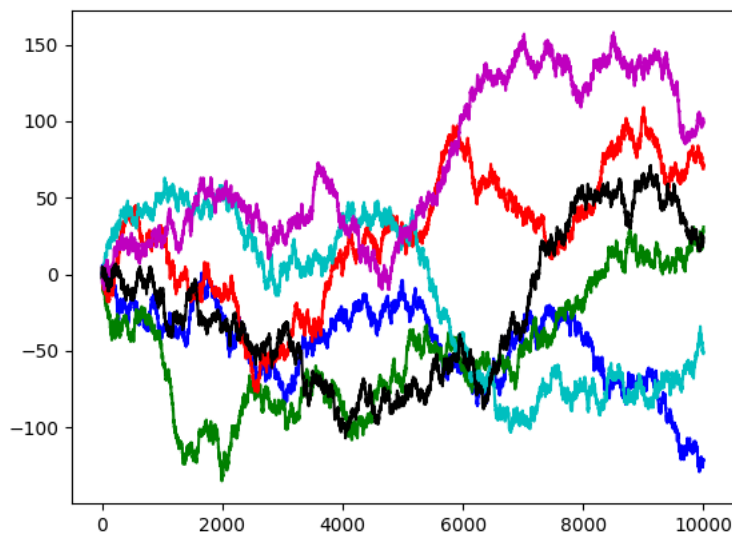
I wrote a script to check a buch of github project directories and report their status

- [git check](#)

Finally, there is code using the plotting library Matplotlib to demonstrate trials of a 1D random walk.

- [walk](#)

The output:



Bare scripts

Addition and multiplication for binary numbers represented as strings of 0 and 1 .

- [binary operations](#)