So I finally got around to upgrading my NextCloudPi to version 20 with the hub and all. I really like it so far.

Different web browsers use the same default fonts on the same operating system. However, the default fonts differ between operating systems and few fonts are available everywhere. In this article, I'll focus on making the default fonts look and behave the same across operating systems.

The default fonts for the generic serif and sans-serif (without serifs) fonts are metrically compatible across macOS, iOS, Windows, and Chrome OS. There are good options available for Linux too, but those are rarely set as the default fonts. Android only has one font, so you get what you get.

Metrically compatible means that each individual character in one font has the same width as the same characters in another font. Metrical compatibility means you can substitute one font for another in a document without affecting its layout. E.g. a headline and paragraph should fill the same number of lines of text. This isn't the same as a fixed-width font (monospace), where all the characters in the font have the same width.
It hasn't been long since the last Web progress report, but it's finally time for the blog entry you have probably been waiting for... time to talk about integrating Godot with third-party JavaScript APIs on the Web.

**Considering the Next Critical Tech Talent Shortage** [5]

But, COBOL expertise is just one example of a potential talent shortage. Where should we expect to find future crises? Loukides asks. What other shortages might occur? The key to understanding or predicting such shortages, he says, involves looking at critical infrastructure.

**Code as Infrastructure** [6]

The Next Critical Talent Shortage Won't Be Fortran

A few months ago, I was asked if there were any older technologies other than COBOL where we were in serious danger of running out of talent. They wanted me to talk about Fortran, but I didn't take the bait. I don't think there will be a critical shortage of Fortran programmers now or at any time in the future. But there's a bigger question lurking behind Fortran and COBOL: what are the ingredients of a technology shortage? Why is running out of COBOL programmers a problem?

The answer, I think, is fairly simple. We always hear about the millions (if not billions) of lines of COBOL code running financial and government institutions, in many cases code that was written in the 1960s or 70s and hasn't been touched since. That means that COBOL code is infrastructure we rely on, like roads and bridges. If a bridge collapses, or an interstate highway falls into disrepair, that's a big problem. The same is true of the software running banks.

**Calculate the mean or average of a single column of numbers in a text file** [7]

**Rakudo compiler, Release #147 (2021.06)** [8]

On behalf of the Rakudo development team, I'm very happy to announce the June 2021 release of Rakudo #147. Rakudo is an implementation of the Raku1 language.

The source tarball for this release is available from https://rakudo.org/files/rakudo. Pre-compiled archives will be available shortly.