Racket is a general-purpose, object-oriented, multi-paradigm, functional, imperative, logic based programming language based on the Scheme dialect of Lisp. It’s designed to be a platform for programming language design and implementation.

Racket is also used to refer to the family of Racket programming languages and the set of tools supporting development on and with Racket. It has a powerful cross-platform GUI library built in.

Racket’s core language includes macros, modules, lexical closures, tail calls, delimited continuations, parameters (fluid variables), software contracts, green and OS threads, and more. The language also comes with primitives, such as eventspaces and custodians, which control resource management and enables the language to act like an operating system for loading and managing other programs.

Racket is often used for scripting, computer science education, and research. It’s an open-source project (Apache/MIT).

Here’s our recommended tutorials to learn Racket.

Pangeo with Dask Gateway

Over the past few weeks, we have made some exciting changes to Pangeo’s cloud deployments. These changes will make using Pangeo’s clusters easier for users while making the deployments more secure and maintainable for administrators.

Going all the way back to the initial prototype, Pangeo’s cloud deployments have combined a
user interface like Jupyterlab with scalable computing. Until recently, Pangeo used Dask Kubernetes to start Dask clusters on a Kubernetes cluster. This worked well for several years, but there were a few drawbacks.

- **We are happy to announce the first release of Jaybird 4.** [4]

  We are happy to announce the first release of Jaybird 4.

  Jaybird 4 is compared to Jaybird 3 an incremental release that builds on the foundations of Jaybird 3. The focus of this release has been on further improving JDBC support and adding support for the new data types and features of Firebird 4.

- **How failure-driven development makes you successful** [5]

  My job title is senior software engineer, but that's not what my closest co-workers call me. They call me "Cherrybomb" because of all the things I blow up. My regularly scheduled failures have been tracked down to our quarterly earnings and outage times. Literally, I am the production disaster you read about that says, "what not to do ever, in any case, at any time."

  I started my career at a helpdesk where I wrote loops that wrecked servers in high-end companies. I have taken production applications down for up to eight hours without warning, and I have destroyed endless numbers of clusters in an attempt to make things better?and a couple just because I mistyped something.

  I am the reason we have disaster recovery (DR) clusters in Kubernetes. I am the chaos engineer that, without warning, teaches people how to act and troubleshoot quickly when we have an application that has never been tested for an outage recovery plan. I exist as the best example of failure possible, and it's actually the coolest thing ever.

- **Digital Making at Home: Making games** [6]

- **Code Hyper Sports? shooting minigame | Wireframe #35** [7]

- **If you've ever wished Visual Studio Code could be more open source, the Eclipse Foundation would like a word** [8]
The Eclipse Foundation has pulled back the curtains on version 1.0 of Theia, an alternative to Microsoft's developer darling of the hour, Visual Studio Code.

Except it isn't just yet. Those hoping to ditch a Microsoft-branded IDE for something more vendor-neutral might have a while to wait for something to drop from Eclipse itself, although a hop to somewhere like Gitpod will give those interested a look at the cloudy version.

Eclipse Theia is a framework on which organisations can build and brand their own products, on the desktop or online, rather than a standalone editor.

●

2020.13 NoConf Reached [9]

It’s a sad moment in time when you realize that basically all conferences have been cancelled for the foreseeable future: the Perl and Raku Conference in Houston, Perl & Raku Con in Amsterdam to name but a few. Some organizers even came to the conclusion that organizing ?in person? events is no longer a viable business model (/r/perl comments).

●

More mojibake fun [10]

The names contain reconstructable replos. A 2-byte, UTF-8 encoded character was read by a Windows program byte-by-byte to produce 2 new 1-byte characters, and those 2 1-byte characters were converted back to UTF-8 as 2-byte ones.

●

D is for Devilish Place Names [11]

The downloaded file is a format called CSV ("Comma Separated Values", though in this case they're separated by the pipe character, "|"), typically used in spreadsheets. I'm not really a spreadsheet person, and CSV files are just as easy to analyze using basic shell tools. Most Linux users are familiar with the power of the command line, but don't feel left out if you're not using Linux: the commands I'll show work fine on a Mac, and they probably work on Windows too if you use the Linux Subsystem for Windows.

I started with a basic count. I'd seen already, on the website's search page, that a lot of the names didn't actually have "Devil" in the name even though that's what I searched for, so that 1883 number is bogus. So I ran a grep -i devil to pick out the place names that actually do have "devil" in the name (-i means "ignore case", so it will find devil as well as Devil). Then I piped the result through wc, word count, using -l to count the number of matching lines:

grep -i devil GNIS_Devil.csv | wc -l