KDE Planet - Latest News

- KStars v3.8.0 is Released (2025/12/03 19:24)
  KStars v3.8.0 is released on 2025.12.03 for Windows & Linux. MacOS release is expected in one week due to build issue on KDE CI infrastructure.For Linux users, it's highly recommended to use the official KStars Flatpak hosted at Flathub. You can install the stable flatpak or try out new features by downloading the KStars Nightly Flatpak for x86_64 and arm64 architectures.Live Stacker: LRGB StackingJohn Evans implemented generation of RGB and LRGB images from individual mono subframes. Watch R, G, B and optionally L directories and combine the individual stacks into RGB or LRGB images.Add directories for R, G, B and optionally L subs. These are monitored and a single color image is displayed.RGB images are combined with a Linear Fit type algorithm. LRGB images are combined with a LRGB Combination type algorithm. SNR algorithm has been rewritten. Appears to work better but is more resource intensive.Live Stacker: ImageMM Stacking MethodJohn Evans added an implementation of the ImageMM stacking method. This implementation strikes a balance between speed (it needs to be a Live Stacker) and fidelity.To use: select ImageMM as the stacking method and play with the available controls. This method is considerably more resource intensive than "regular" stacking because it uses an iterative approach.Live Stacker: Live Stacker MonitorJohn Evans implemented the Live Stacker monitor. Live Stacking Monitor window is a popup from Live Stacker that shows a table of subs that match the chosen directory in Live Stacker.The purpose is to allow analysis of Live Stacker, for example to allow investigation of bottlenecks in the stacking process.When a sub is added to the watched directory, the sub is added to the Monitor's table. As the sub is processed by each step of the process information is updated in the table:Waiting to load. The sub is in the queue to be processed but Live Stacker is busy with other subs.Loading. The sub is loaded into memory.Plate solving. The sub is undergoing plate solving (if appropriate).Waiting to stack. The sub is waiting to be stacked (e.g. currently there are insufficient subs loaded to start a stack).Calibration. Dark / Flat calibration.Alignment.Stacking.Table columns and sort order are configurable. Changing cells can be highlighted (or not)Task Queue systemObservatory startup and shutdown steps are now replaced by the new highly configurable Task Queue system. The Task Queue System is a modern, flexible automation framework that replaces traditional startup and shutdown scripts with a template-based, configurable task execution system. It provides a visual interface for building sequences of automated operations that can control your observatory equipment through INDI. It is accessible from Ekos Scheduler.Why use it? The Task Queue system offers several advantages over traditional scripting:Visual Management: Build and monitor task sequences through an intuitive graphical interfaceReusability: Use pre-built templates for common operations without writing codeError Handling: Built-in retry logic and configurable failure responsesDevice Compatibility: Automatic matching of templates to available devicesProgress Monitoring: Real-time status updates and detailed execution logsCollections: Pre-defined task sets for startup, shutdown, and other common scenariosFlexibility: Combine templates or create custom variations without programmingSafety MonitorKStars scheduler now fully supports INDI Safety Monitor driver released part of INDI v2.1.7. A standalone driver may be used (independent of the equipment profile) that is running on a different INDI server to provide 24/7 safety monitor updates to the scheduler. No observatory operations shall take place unless it is deemed safe by the safety monitor.The INDI Safety Monitor can listen to any number of sources including weather stations, UPS (uninterruptible power supply) monitors, and any auxiliary device that support the INDI's standard SAFETY_STATUS property.Push-To Assistant Wolfgang Reissenberger added an incredibly useful tool for users

with manual mounts: Push-To Assistant. Just attached a camera to your dobsonian and use this tool to center the target in the eyepiece. This tool assumes that both the camera center and eyepiece center are already aligned.The new push-to assistant is intended as plate solving support for mechanical mounts in combination with a digital camera on a finder scope.Setup: Create an optical train with the Telescope Simulator as mount and configure your combination of finder scope and digital camera.UsageStart Ekosopen the Push-to Assistant located in the Tools menumove your scope as good as possible to the target you want to findselect the target from the catalog or enter its coordinates manually and press "Select target"press "Solve position" to determine the position your scope is currently pointing atas soon as the position has been solved, the assistant displays hints in which direction you should move your mount to be closer to the targetcorrect your mount position and press "Solve position" againif you want to automatically repeat plate solving, configure the delay and press the "Repeat" button.Bug FixesFix crash when connecting to remote host because the clientManager was not set.When restarting capture, update job captured frame map from module state.Fix issue with J2000 getting set by default in Mount Control Panel.Fix regression where capture format and encoding are reset whenever they are changed in INDI.Use the offline HIPS path since it can be a custom locationPrevent premature alignment run if there are no more light frames in the jobDo not cover dust cap when capturing sky flats.Fix setFrame when loaded from XML as DARK was incorrectly decoded as BIAS frame.Remove parents from constructors of objects pointed to by QSharedPointerFix issue in FITS Solver, need to disconnect signals in callbacksFix bug in one-pulse-dither, update loggingFix issue in analyze stats font sizePlaceholder %cam removed

- FreeBSD 14 and KDE (2025/12/02 23:00)
  Huh, I started a blog post on November 11th, and then got distracted. What I was going to write was about KDE Plasma 6 on FreeBSD 14, how it does with Wayland, what kind of configuration surprises there are. And since then FreeBSD 15 has been released, which makes this title kind of moot, and I still need to write all the things down. The summary is short: with AMD graphics, KDE Plasma 6 Wayland is just fine for my workflow.
- Last Two Weeks in KDE Apps (2025/12/02 21:11)
  Performance improvement in Krita, Trust and Safety in NeoChat and files actions in PhotosWelcome to a new issue of "This Week in KDE Apps"! Every week (or so) we cover as much as possible of what's happening in the world of KDE apps. We are still doing our fundraisers and in the past 48 hours, thanks to the crazy support from our users we managed to raise more than €90,000. Keep it going and if you can afford it, donate at kde.org/donate! Any amount helps. Getting back to all that's new in the KDE App scene, let's dig in! Travel Applications Volker Krause published a blog post about the current progress of KDE Itinerary in October and November. This includes an improved journey search page, fine-grained deletion control of tickets, altitude information in the live status view, and more! You can read all of that on his blog. Grapics Applications Okular View and annotate documents Mohammad Kazemi added a “Copy Without Line Breaks” action to remove line breaks when copying text (26.04.0 - link). Quinten Kock added native pinch gestures with a touchpad in Okular (26.04.0 - link). Photos Image Gallery Noah Davis added more standard file actions in Photos when viewing a picture (26.04.0 - link). KPhotoAlbum KDE image management software Randall Rude made the metadata extractor also extract the creation date and time for videos (link). Creative Applications Krita Digital Painting, Creative Freedom Agata Cacko improved the performance of the Liquify Transform tool making it a lot more smooth to use (link). Agata also added a knife tool prototype to Krita (link). Joshua Goins removed the error dialog when cancelling an export (link). Utilities Applications Konsole Use the command line interface Matan Ziv-Av added two keyboard actions in Konsole for focusing on the next/previous view in split view mode (26.04.0 - link). Sune Vuorela added an option to enable or disable whether Konsole listens to zmodem terminal codes, which might happen accidentally when outputting a binary file. (26.04.0 - link) Kate Advanced text editor Héctor Mesa Jiménez added some default configuration for netcoredbg, a

standalone debug server for .NET Core. (26.04.0 - link) Alligator RSS feed reader Oula V improved the feed group feature of Alligator. Now when creating a feed group, you will get an error if another one exists with the same name. They also cleaned up the list of feed groups (26.04.0 - link). Oula also fixed some crashes in Alligator after editing a feed (25.12.0 - link) and Stephan Seitz fixed some conformance issues with the OPML export feature (25.12.0 - link). Salvo Tomaselli reordered the buttons in the menu, and now opening the current article in an external browser is the first button (25.12.0 - link). System Applications Dolphin Manage your files Alex Hermann made KIO-powered applications like Dolphin keep the permissions of files copied from an SFTP server (link 1, link 2, link 3). Social Applications NeoChat Chat on Matrix Joshua Goins continued efforts to improve Trust and Safety in NeoChat and added support for reporting rooms and users (26.04.0 - link). "renner 03" fixed the KRunner integration of NeoChat when running the application in Flatpak (25.12.0 - link) Browsers Konqueror KDE File Manager & Web Browser Stefano Crocco added a configuration page to configure Speed Dials in Konqueror. These speed dials are buttons that allow you to quickly open pre-configured links (26.04.0 - link). Falkon Web Browser Juraj Oravec added support to add items in the sidebar menu to the Falkon plugin API (link). Angelfish Webbrowser for mobile devices Rinigus Saar fixed an issue with retrieving the last visited entries (25.12.0 - link) PIM Applications Trojitá IMAP E-mail Client Sandøy Hustad started pushing some work to make Trojita support Qt 6 (link). Third-party Applications Deskflow - Keyboard and mouse sharing app Chris Rizzitello released Deskflow 1.25.0! The main changes are support for a symbolic tray icon which is recolored correctly even when using Plasma's Twilight theme; support for changing the application's language without restarting it; and initial support for the wl-clipboard Wayland protocol. EasyEffect Giusy Digital continued working on unifying the wording of the various physical units (e.g. dB, Hz, ...) all over the application (link). Wellington Wallace ported some overlay sheets to Kirigami dialogs (link). …And Everything Else This blog only covers the tip of the iceberg! If you’re hungry for more, check out Nate's blog about Plasma and be sure not to miss his This Week in Plasma series, where every Saturday he covers all the work being put into KDE's Plasma desktop environment. For a complete overview of what's going on, visit KDE's Planet, where you can find all KDE news unfiltered directly from our contributors. Get Involved The KDE organization has become important in the world, and your time and contributions have helped us get there. As we grow, we're going to need your support for KDE to become sustainable. You can help KDE by becoming an active community member and getting involved. Each contributor makes a huge difference in KDE — you are not a number or a cog in a machine! You don’t have to be a programmer either. There are many things you can do: you can help hunt and confirm bugs, even maybe solve them; contribute designs for wallpapers, web pages, icons and app interfaces; translate messages and menu items into your own language; promote KDE in your local community; and a ton more things. You can also help us by donating. Any monetary contribution, however small, will help us cover operational costs, salaries, travel expenses for contributors and in general just keep KDE bringing Free Software to the world. To get your application mentioned here, please ping us in invent or in Matrix.

- Choosing AnyType: A Platform for Scalable, Secure Governance (2025/12/02 07:00)
  STF needed a scalable, secure, and asynchronous collaboration system for representatives across multiple time zones. This article explains why AnyType was selected, how it is used today, and which features and challenges matter most as the Software Transparency Foundation.
- Boosting PySide with C++ models (2025/12/01 13:03)
  In a recent series of blog posts, we have demonstrated that Python and Qt fit together very well. Due to its accessibility, ease-of-use and third-party ecosystem, it is really straightforward to prototype and productize applications. Still, Python has one significant disadvantage: It is not necessarily the most performant programming language. Continue reading Boosting PySide with C++ models at basysKom GmbH.
- October/November in KDE Itinerary (2025/11/29 10:00)

In the past two months since the last post, KDE Itinerary’s journey search UI got simplified, you got more control over deleting individual entries and altitude information is shown on the live status page when available, among many other things. New Features Improved journey search page The interface for manual public transport searches as been simplified. Filters for specific modes of transportation are now on a secondary page, and you don’t have to specify a trip to add the results to in the first step anymore. Instead, that’s now queried when actually saving a result. Simplified journey search page. Fine-grained deletion control For multi-ticket or multi-traveler reservations, it’s now possible to delete just individual tickets or travelers rather than the entire entry. Multi-ticket deletion dialog. Altitude information in live status view The live map on services with the corresponding onboard API now also shows the current altitude information when available. Live train position with altitude. Infrastructure Work Automatic geocoding for reservation data For many of Itinerary’s features to work properly we need to know geo locations of the involved places, such as departure and arrival stops of a train trip. In many cases we get those from being able to recognize stop identifiers found in e.g. ticket barcodes. There’s a bunch of heuristics as fallback (such as knowing in which areas a train company operates), but that’s also not covering all cases. To address this properly, Itinerary can now resolve those remaining locations by using OSM’s geocoder Nominatim. As this involves querying an online service, this is conditional on having online data sources enabled in the settings, same as for querying for delay information. Transitous upgrade to MOTIS v2.7 Upgrading MOTIS, the software behind Transitous brought us a number of new features, with the following ones particularly relevant for Itinerary: State and positions of currently available rental vehicles such as bikes and scooters can now be queried. Support for GBFS station booking URLs. Support for multiple language preferences. That means that secondary languages are now also considered when picking the best option for multi-lingual content such as disruption notes. Location searches include the modes of transportation served at stops now. Itinerary's station map showing a car rental station and two free-floating rental bikes. Android platform support KDE’s Android build infrastructure (which Itinerary relies on) has been updated to Android’s NDK r28, which enabled compliance with the 16kB page size requirement enforced by the Google Play Store since November 1st. While this is something that went mostly unnoticed by users, the next required update (to Qt 6.10) is unfortunately going to have some more side-effects. For the first time in many years this will require a higher minimum Android version, going from currently 21 (Android 5, from 2014, 99.8% cumulative use) to then 28 (Android 9, from 2018, 91.7% cumulative use). This means any newer build of any KDE Android app would no longer run on anything older than Android 9. It’s unclear how many of our users would be affected by this, but it unfortunately does look like we have very little choice here beyond delaying this a bit. If you have thoughts or feedback on this, feel free to join the KDE Android Matrix channel. Events There also were several events with Itinerary-adjacent topics in the past two months: The Open Transport Community Conference in Vienna. Two OSM Hack Weekends in Berlin and Karlsruhe. Itinerary also got mentioned in the whirlwind tour through the land of Wikidata-powered_apps at Wikimania in Nairobi. And more is coming up, members of the Itinerary and Transitous teams will be at 39C3 end of December in Hamburg, Germany as well as FOSDEM at the beginning of February in Brussels, Belgium. Fixes & Improvements Travel document extractor Added or improved travel document extractors for Booking.com, CFR, citycity.se, Comboios de Portugal, Eurostar, Flixbus, FooEvents, GlobalTicket, Inviton, MÁV, NH Hotels, Predpredaj, Prioticket, Ryanair, TicketCounter, United, Ventra, Wiener Linien and ZSSK. Consider bus stations and try harder to discard freight terminals when locating airport entrances. Ignore seat qualifiers (“window”, “aisle”, etc) when comparing seat numbers. Merge common parts of all elements of the same incidence such as a multi-ticket booking. Consider names with swapped given/family name as equivalent as well All of this has been made possible thanks to your travel document donations! Public transport data Add a vehicle feature flag for night trains (supported with MOTIS and Hafas backends). Add support for agency/operator URLs (supported with MOTIS and OTP backends). This can be useful as this is the most widely

available way towards actually booking something at this point. Improved onboard API support for Frecciarossa and RailJet trains. Improved performance of the location search page. All of this also directly benefits KTrip. Itinerary app Editing now affects the currently selected reservation in a multi-ticket or multi-traveler batch. Only load reservation data for the current trip group. Re-add the top-level import action. Fix performance issues and hangs when displaying journey search results. Fix online updates for standalone Apple Wallet pass tickets such as Zügli D-Tickets. Fix timer overflow in transfer monitoring with Qt 6.10. Adding a journey search result when there's no existing trip will now directly ask to create a new trip. Fix updating platform information from scheduled online data in case the platform in the original ticket was wrong. How you can help Feedback and travel document samples are very much welcome, as are all other forms of contributions. Feel free to join us in the KDE Itinerary Matrix channel.

- Kdenlive 25.12 RC ready for testing (2025/11/29 06:00)
  The Kdenlive 25.12 Release Candidate is ready for testing. We made several changes to the user interface to improve your workflow, including a new widget docking system that makes rearranging panels much easier and more powerful, an enhanced audio display in the clip monitor with a waveform overview for faster navigation and zooming, and a new Startup and Welcome screen allowing to easily select a few options when launching the program. Other highlights: Added an editing layout and safe areas for vertical formats Reordering of the menus to make them more logical Introduction of markers with a time span (GSoC 2025) Feedback Needed Now is your chance to test it and let us know if you encounter any bugs or have suggestions to help us polish the final release. Share your feedback either in the comments below or directly with the team during our online Café, where we'll be discussing this upcoming release. Join us next Wednesday, 3rd of December at 21:00 CET, on meet.kde.org Update 30th of November 2025 The original RC Appimage for Linux was broken on X11, we have now fixed it and the download link will give you the RC2 version. Download the binaries from below and give it a spin! Pre-release binaries can be downloaded here.
- This Week in Plasma: lots of cool stuff (2025/11/29 00:01)
  Welcome to a new issue of This Week in Plasma! This week saw quite a lot of feature work and user interface polish for Plasma 6.6! Have a look: Notable New Features Plasma 6.6.0 There are now global actions for seeking forwards or backwards 5 or 30 seconds in the currently playing media. These work as long as the current media player supports letting other apps control seeking via MPRIS. The actions don't have keyboard shortcuts assigned by default, but you can set them yourself. (Christoph Wolk, link) You can now configure the Window List widget to show its menu on hover, or to hide the icon and only show the name of the active window. (Shubham Arora, link 1 and link 2) You can now configure the order of the icons shown in the Lock/Logout widget. (Shubham Arora, link) Notable UI Improvements Plasma 6.6.0 Continued to polish up the XDG portal dialogs. This time the screen & window chooser dialog has been simplified and improved some more. (Harald Sitter, link) Canceling a paste of some files onto Plasma's desktop no longer produces a pointless error notification. (Nicolas Fella, link) Added pin buttons to the Web Browser and Audio Volume widgets so that if you have either them in standalone form on a panel, you can keep their popups open while you're still using them. (Alexander Lohnau and Nate Graham, link 1 and link 2) Improved the usability of searching using the Kickoff Application Launcher widget in several ways: now you can use the arrow keys to navigate from the search results view back to the search field, and new search results that come in late don't cause the selection highlight to jump around. (Christoph Wolk, link 1 and link 2) Improved the appearance and usability of the disks shown in Info Center's S.M.A.R.T. Status page. (Joshua Goins, link) Clearing the clipboard while it happens to be showing a QR code for one of the now-cleared items now dismisses the QR code, too. (Fushan Wen, link) The Kicker Application Menu widget no longer very briefly flashes a message saying "No matches found" right after you search for things. (Christoph Wolk, link) Improved the experience of rapidly moving the

pointer over top-level menu items in the Kicker Application widget; now the sub-menus appear in the same way they do for other menus in apps. (Christoph Wolk, link) When using the Kicker Application Menu widget on a right screen-edge panel, sub-sub-menus now open to the left of their parent, rather than on the right where they cover up the main menu. This also now works properly with an RTL language and a left screen-edge panel. (Christoph Wolk, link 1, link 2) Frameworks 6.21, with the full effect arriving in Plasma 6.6.0 The headers of Kirigami-based apps are now the same height as those of QtWidgets-based apps. In the process of improving this, we also managed to equalize the padding on all four sides of highlighted list items, and make them consistent with the outer padding of header items, too. (Marco Martin, link 1, link 2, link 3, link 4, link 5, link 6, and link 7) The Fifteen Puzzle widget now has a nice new icon, and also uses a symbolic icon when placed on a panel. (Martin Sh, link 1 and link 2) Frameworks 6.21 Reverted a change made a few months ago that hid .desktop files with NoDisplay=true set on them from apps' "Open with" menus. While the original change seemed technically correct, it had negative side effects outweighing its advantages. (Nate Graham, link) Notable Bug Fixes Plasma 6.5.4 Fixed a random Plasma crash. (Nicolas Fella, link) Fixed a case where turning on automatic updates in Discover would just make Discover crash in the background rather than running the updates. (Aleix Pol Gonzalez, link) Fixed an issue that broke pasting images from the clipboard into Dolphin. (Vlad Zahorodnii, link) Fixed a case where drawing using certain oddly-behaving drawing tablets would draw outside of the screen area. (Xaver Hugl, link) Fixed a case where re-mapping drawing tablet stylus buttons didn't work. (Joshua Goins, link) Plasma 6.6.0 Fixed an issue that mangled the desktop icon arrangement when dragging something to the desktop while it was using one of the automatic sorting modes. (Błażej Szczygieł, link) Fixed an issue that made certain GPUs get displayed as "llvmpipe" in Info Center. (Oleg Gorobets and Harald Sitter, link 1 and link 2) Fixed some issues that made Plasma's desktop sometimes fail to notice newly-created, -deleted, or -changed files. (Błażej Szczygieł, link) Fixed an issue that prevented re-opening the virtual keyboard immediately after closing it, but before clicking or re-focusing anything else. (Xaver Hugl, link) Frameworks 6.21 Fixed an issue that mangled the display of devices whose names contain Unicode characters in USB plug/unplug notifications. (Nicolas Fella, link) When you drag and drop an item from a Dolphin window that's accessing a network location that requires Kerberos authentication, dropping it on the desktop now successfully downloads the file. (Kai Uwe Broulik, link) Other bug information of note: 5 very high priority Plasma bugs (up from 4 last week). Current list of bugs 36 15-minute Plasma bugs (up from 31 last week). Current list of bugs Notable in Performance & Technical Plasma 6.6.0 Implemented support for XRandr emulation in KWin, which allows it to behave sensibly when running XWayland-using apps that make use of X11 APIs to change the screen resolution in a way that requires letterboxing or pillarboxing. (Xaver Hugl, link) Improved the performance of the rectangular box you can drag on the desktop to select items. (Błażej Szczygieł, link) Implemented support for the standard "reduced motion" setting that lets apps know that you'd like animations minimized. Now it's toggled on automatically when you disable animations in System Settings. (Nicolas Fella, link) Plasma 6.5.4 Fixed a memory leak in Plasma. (Vlad Zahorodnii, link) How You Can Help Donate to KDE's 2025 fundraiser! It really makes a big difference. Believe it or not, we're up to almost €90,000 raised in a month and a half. This money will help keep KDE strong and independent for years to come, and I'm just in awe of the generosity of the KDE community and userbase. Thank you all for helping KDE to grow and prosper! If money is tight, you can help KDE by directly getting involved. Donating time is actually more impactful than donating money. Each contributor makes a huge difference in KDE — you are not a number or a cog in a machine! You don't have to be a programmer, either; many other opportunities exist. To get a new Plasma feature or a bugfix mentioned here, feel free to push a commit to the relevant merge request on invent.kde.org.

- Web Review, Week 2025-48 (2025/11/28 11:57)
  Let's go for my web review for the week 2025-48. Open Source Power Tags: tech, foss, licensing, business, economics, politics, commons,

criticism This debate around licensing, politics and making our FOSS efforts sustainable need to happen. It looks like for now to some people the path forward is defensive licensing? I wish at least we'd first attempt to have more strong copyleft use…
https://blog.muni.town/open-source-power/ Open Source Has Too Many Parasocial Relationships Tags: tech, foss, maintenance, sustainability, supply-chain, commons Indeed, if you benefit from Free Software you'd better engage with it. Maintainers should stop bending backwards to please free loaders. https://pivotnine.com/blog/open-source-has-too-many-parasocial-relationships/ What They Don't Tell You About Maintaining an Open Source Project Tags: tech, foss, maintenance Want to start a new project? Here is what you're signing for.
https://andrej.sh/blog/maintaining-open-source-project/ A Message to the Computing Community About ACM's Transition to Full Open Access Tags: tech, science, research, open-access Excellent news! It is long overdue that such organisations switch to open access.
https://cacm.acm.org/news/a-message-to-the-computing-community-about-acms-transition-to-full-open-access/ Personal blogs are back, should niche blogs be next? Tags: tech, blog, web, culture This would probably be a good thing indeed. We'll see of the web culture will evolve next.
https://disassociated.com/personal-blogs-back-niche-blogs-next/ Zig: Migrating from GitHub to Codeberg Tags: tech, git, forgejo, foss, community, vendor-lockin Very good move on their part. It's time more people do so. Beside, Forgejo (powering Codeberg) looks very interesting. I plan to play with it more next year. https://ziglang.org/news/migrating-from-github-to-codeberg/ s&box is a modern game engine, built on Valve's Source 2 and the latest .NET technology Tags: tech, 3d, game, foss Nice to see another game engine go the Free Software route. This one is particularly feature packed. https://github.com/Facepunch/sbox-public libfive Tags: tech, framework, geometry, 3d Looks like a neat software library for procedural geometry. https://libfive.com/ How Cloudflare uses Rust to serve (and break) millions of websites at 50+ millions requests per second Tags: tech, cloudflare, rust, reliability, failure A bit of a shameless plug toward the end. That said the explanations of why Cloudflare is banking on Rust so much or how the recent downtime could have been avoided are spot on. https://kerkour.com/how-cloudflare-uses-rust Rust unit testing: file reading Tags: tech, rust, tests Nice approach to stub standard types in Rust. The article is a bit confusing the different types of test doubles though. https://jorgeortiz.dev/posts/rust_unit_testing_file_reading/ Linux Kernel Explorer Tags: tech, linux, kernel Looks like a neat code explorer for the kernel. It's nice that it comes with a guide to point you to the right places per topic. https://reverser.dev/linux-kernel-explorer The Input Stack on Linux — An End-To-End Architecture Overview Tags: tech, linux, kernel, system, input A long article which seems to be a good reference document on the Linux input stack. There's a lot to cover as it's quite fragmented.
https://venam.net/blog/unix/2025/11/27/input_devices_linux.html Solving Fizz Buzz with Cosines Tags: tech, programming, mathematics, funny Ever wondered if we could solve the Fizz Buzz with a Fourier series? Trigonometry is magic. https://susam.net/fizz-buzz-with-cosines.html Fifty Shades of OOP Tags: tech, object-oriented, type-systems Another post which reminds everyone what object oriented programming is about. And yes, there's indeed a variety of different tools in there, not all object oriented languages are equivalent.
https://lesleylai.info/en/fifty_shades_of_oop/ What Now? Handling Errors in Large Systems Tags: tech, reliability, failure, complexity Error handling is not easy. Having simple rules to apply for complex systems is a good thing. Of course the difficulty is to apply them consistently.
https://brooker.co.za/blog/2025/11/20/what-now.html 10 Years of thinking about Pair Programming Tags: tech, pairing, programming Gives an idea of what pair programming looks like when practiced properly.
https://salfreudenberg.wordpress.com/2013/11/16/10-years-of-thinking-about-pair-programming/ We stopped roadmap work for a week and fixed 189 bugs Tags: tech, programming, quality, craftsmanship, engineering I'm not really a fan of the leaderboard part of their approach. That said, if the maturity of the organisation allows it, having such bug squashing sessions is a good idea. https://lalitm.com/fixits-are-good-for-the-soul/

"Good engineering management" is a fad Tags: tech, engineering, management Interesting thinking, indeed expectations are changing quite a bit for engineering managers over time. Thus the proposed list of core and growth skills is interesting. It is likely a good framing for the job, then the art is finding the right balance for your organisation. https://lethain.com/good-eng-mgmt-is-a-fad/ Solve problems with experiments Tags: tech, team, organization, agile, problem-solving You can also have experiments on your organisation. This is actually a good thing and probably should be done when something keeps popping up as a problem. https://www.viktorcessan.com/solve-problems-with-experiments/ Invert, always invert Tags: tech, planning, risk, project-management Interesting approach I didn't know about. Definitely worth trying. I like how it seems to bake risk management in. https://www.theengineeringmanager.com/growth/invert-always-invert/ Bye for now!

- KDE @ IndiaFOSS 25 (2025/11/26 23:20)
  Its been quite a while since I attended the IndiaFOSS 25 event late September. I have been meaning to write the blog post for a while but didn't get time. A lot has happened since then, I have moved continents and am now in Germany pursuing my masters at Philipps University of Marburg. I'll write lot more about my new experiences in Germany in upcoming blog posts but for now back to IndiaFOSS 25. This was a very exciting event for me since this was the first time I was representing KDE booth at a conference. This event occurs yearly in the city of Bangalore which is also known as the Silicon Valley of India. I discovered IndiaFOSS after watching a talk by Kovid Goyal, creator of Calibre and Kitty - two programs I have used heavily in my day to day life so regardless to say I was excited about the conference. Day 0 Nearby area of bangalore airport as seen from the sky I arrived a day before the conference in Bangalore and met with Bhushan at the airport as we continued our journey towards the city. The Kempegowda International Airport is located outside the city about 40km north of Bangalore's city center. It took us around 2 hours to reach the airbnb I was sharing with the my friends who were also attending the event. After encountering the much dreaded city traffic jam I took a much needed rest and spent my rest of the day exploring the area of Jayanagar and Indiranagar. Day 1 I spent most of my time at the booth and met a lot of people who were already using KDE software, specifically from the state of Kerala. It was made popular through the ICT initiative. I have to be honest here, I did hear about this initiative from Subin earlier this year at cki25 but didn't know it was this widespread. Booth photo with Bhushan and Advaith with Raspberry Pi running KDE Plasma powering the TV screen behind them. There was general amazement among the booth visitors when we showed them KDE powered Laptops, mobile phones, steamdeck and even a rasberry pi powering the giant tv screen at our booth. The steamdeck was the most popular device at the booth since its still not officially available in India. Some people already knew about the steamdeck while others who didn't were amazed to find out that gaming on handheld that too on a linux powered device had come such a long way. Booth photo with me and Advaith Some of the people who we met at the booth were also interested in helping us spread the word and gave us testomonials which I have forwarded to Aniqa. Additionally we were able to help few people with creating their invent profiles and there were others who showed interest in helping out with code contributions and translation support. Few also asked about India specific KDE events and we were happy to tell them about Conf KDE India (cki 25). Booth attendee doodling with Krita We also connected with James Reilly from AlmaLinux (can be reached on Matrix), who is looking for a KDE Community member to create a KDE bootc-image built on CentOS base and has also offered to mentor a new contributor in tackling this project. KDE x Inkscape Our booth was also located right next to Inkscape's booth and seeing Rishi make Konqi on Krita led them to battle it out to see who can make a Konqi faster. It was a friendly battle ofcourse and all of us enjoyed the drawing contest complete with cheer and laughter. It was a pretty eventful day and we met a lot of users of KDE software at our booth. Day 2 The second day of the conference saw the attendance dwindle a bit but this also gave me a chance to checkout the other booths. Some of the most interesting ones that I found were (in no particular order):- Mecha Comet Mecha Comet Internals Breakdown

It is a handheld modular Linux computer. It has a magnetic snap interface powered by a standard GPIO serial port which allows hardware extensibility. Additionally Its made up of repairable and recyclable parts. The particular distro that it runs is called Mechanix OS, which is based on Debian. It packs in a PCIe 3.0 M.2 B-Key, Wifi 6, a dedicated Secure Element (CC EAL 6+) for security with a 3.92 inch screen. The project is still in Kickstarter phase but its definitely one of the most exciting projects I found at the conference. You can find more about the project here. SFLC India SFLC Logo Software Freedom Law Center, India is a donor supported legal services organization that promotes Open access to knowledge, Free Speech, Online Privacy, Innovation and Equality. It was because of them, I became aware of the dubious practices surrounding DigiYatra's mandatory biometric data collection in India. Unfortunately, at the time I had already surrendered my data as I was forced to comply, otherwise the airport staff denied me entry. SFLC India at the time had started a huge social media campaign against this and had even gathered huge media attention to bring a change against this. Sadly, the DigiYatra gates still exist but are atleast no longer mandatory. If you wanna know more about them, you can visit them here. Absurd Shop CoryDora DIY keeb assembled by my friend Manik Rana I found their CoryDora DIY Kit to be an interesting project, a fully opensourced 3x3 macropad with hotswappable switches. You can use it to control media, set-up scripts and write macros. A nice project if your just starting with soldering and still getting the hang of things. You can find the CoryDora DIY kit here. PCB CUPID They provide opensource pcb and sensors at affordable prices and also have guides and tools for you to test and play with. Growing up I always wanted to tinker with pcb boards and build with them but always found the boards out of budget so seeing them change this, is definitely for the positive. OpenstreetMap India OpenStreetMap India logo I was happy to see a thriving presence of openstreetmap indian community, I have volunteered in two of their Delhi meetups previously where I got to know how to map data and mapped areas around lesser known of parts of Delhi so it was nice to see some familiar faces. and there were many more… unfortunately I can't cover all the interesting booths that were there at the conference but regardless to say there were plenty! Final Thoughts KDE India users, contributors and booth volunteers together All in all, I enjoyed my stay at Bangalore and was happy to meet the oss enthusiasts from all over India. I was lucky to get a chance to represent KDE at IndiaFOSS and meet with its diverse users and hopefully future contributors. I would also like to thank KDE for making this possible by sponsoring my travel for the event.

- Going all-in on a Wayland future (2025/11/26 00:01)
  Well folks, it's the beginning of a new era: after nearly three decades of KDE desktop environments running on X11, the future KDE Plasma 6.8 release will be Wayland-exclusive! Support for X11 applications will be fully entrusted to Xwayland, and the Plasma X11 session will no longer be included. For most users, this will have no immediate impact. The vast majority of our users are already using the Wayland session, it's the default on most distributions, and some of them have already dropped — or are planning to drop — the Plasma X11 session independently of what we decide. In the longer term, this change opens up new opportunities for features, optimizations, and speed of development. Because we're certain that many people will have questions about this change, the Plasma team has prepared the following FAQ: Plasma 6.8 means the X11 session will be supported by KDE until…? The Plasma X11 session will be supported by KDE into early 2027. We cannot provide a specific date, as we're exploring the possibility of shipping some extra bug-fix releases for Plasma 6.7. The exact timing of the last one will only be known when we get closer to its actual release, which we expect will be sometime in early 2027. What if I still really need X11? This is a perfect use case for long term support (LTS) distributions shipping older versions of Plasma. For example, AlmaLinux 9 includes the Plasma X11 session and will be supported until sometime in 2032. Will X11 applications still work? Outside of rare special cases, yes, they will still work using the Xwayland compatibility layer. It does a great job of providing compatibility for most X11 applications, and we provide several additional compatibility

features on top, namely improved support for fractional scaling and (opt-in) backwards compatibility with X11 global shortcuts and input emulation. In certain cases, 3rd-party applications doing specialized tasks like taking screenshots or screencasting need to be adjusted to work as expected on Wayland. Most have already done so, and the remaining ones are making progress all the time. Does X11 forwarding still work? Yes, Xwayland supports it. Waypipe exists for similar functionality in Wayland native applications as well. Can I still run KDE applications on X11 in another desktop environment? Yes. There are currently no plans to drop X11 support in KDE applications outside of Plasma. This change only concerns Plasma's X11 login session, which is what's going away. What about gaming? Games run better than ever on the Wayland session! Adaptive sync, optional tearing, and high-refresh-rate multi-monitor setups are all supported out of the box. HDR gaming works with some additional setup, too! What about NVIDIA GPUs? While Wayland support in the proprietary NVIDIA driver was quite rocky a few years ago, it has matured tremendously. Graphics cards still supported by the manufacturer work just fine nowadays, and for very old NVIDIA GPUs, the open source Nouveau driver can be used instead. What about accessibility? Accessibility is a very broad topic, so it's hard to make any definite statements, but we're generally on par with the X11 session. All the basics already work as expected, including screen readers, sticky & bounce keys, zooming in, and so on. Some things are better, like touchpad gestures for adjusting the zoom level, and applying systemwide color filters to correct for colorblindness. And even more improvements are expected by the time Plasma 6.8 rolls around. However, accessibility features provided by third-party applications may be worse in some aspects. Please open a bug report if you have any special requirements that we don't cover yet! This is an active topic we're very interested in improving. What about automation? Many tools can be used for automation in the Wayland session; for example wl-copy/wl-paste, ydotool, kdotool, kscreen-doctor, and the plasma-apply-* tools. Generally Plasma is extensible enough that you can add what's still missing yourself, for example through KWin scripts or plugins. What about the Significant Known Issues? While we can't promise all problems will be completely gone (some depend on application support), we're actively working on addressing the last stragglers on that Wiki page. Some of them are really close to being fixed; for example, the issues around output mirroring will be gone in Plasma 6.6. Session restore and remembering window positions are also being actively worked on. What about Plasma on the BSDs? FreeBSD is already shipping a working Wayland session, so there should be no upstream problems on that front. If there are any remaining issues we can help with upstream, please reach out to us! What about the kwin_wayland and kwin_x11 split? In Plasma 6.4, we split KWin into separate X11 and Wayland versions. This allowed KWin to go all-in on Wayland earlier, without being held up so much with legacy support for X11. For users with remaining edge-case requirements for X11, we put in the extra effort to keep X11 support for the rest of the desktop since then. While the split helped a lot, KWin is only one piece of the puzzle. The Plasma desktop as a whole has many places where development is held back by the need to support the lowest common denominator of the two window systems. The bottom line This is happening because we believe that eventually dropping the Plasma X11 session will allow us to move faster to improve stability and functionality for the majority of our users — who are already using Wayland. If we want to keep producing the best free desktop out there, we have to be nimble enough to adapt to a rapidly changing environment with many opportunities, without the need to drag forward legacy support that holds back a great deal of work. The Wayland transition has been long, and at times painful. But we're very close to the finish line. Passing it will unlock a lot of positive changes over the next few years that we think folks are going to appreciate!

- Qt Extension 1.10.0 for VS Code Released (2025/11/25 12:45)
  We're happy to announce the release of version 1.10.0 of the Qt Extension for Visual Studio Code! This release introduces plenty of quality-of-life improvements to developing Qt solutions on Visual Studio Code.

- Thank you SCANOSS. Hello STF. (2025/11/25 07:00)
  After two years helping SCANOSS strengthen its open source and open data strategy, my focus now shifts fully to the Software Transparency Foundation. I'm grateful for SCANOSS's trust and support, and I look forward to continuing our collaboration through STF.
- Qt for MCUs vs LVGL: A Comparative Study from Design to Deployment (2025/11/24 12:10)
  When comparing Qt for MCUs vs LVGL, our independent study with Spyrosoft shows that Qt for MCUs reduces development time for microcontroller GUI by 30% compared to LVGL. The efficiency improvement comes from Qt's integrated toolchain, which facilitates better collaboration among designers (Figma to Qt), developers (Qt Creator or Visual Studio Code), and QA engineers (Squish for MCUs), making Qt for MCUs ideal for complex projects with cross-functional teams. Additionally, Qt for MCUs offers comprehensive safety certification packages for safety-critical industries such as automotive, two-wheelers, and medical applications, positioning it as a superior LVGL alternative when functional safety and regulatory compliance are essential.
- How do we keep apps maintained on Flathub? (or building a more respectful App Store) (2025/11/23 23:00)
  There have been a few discussions about what Flathub should do to push developers to maintain their apps on the latest versions of the published runtimes. But most of those lack important details around how this would actually happen. I will not discuss in this post the technical means that are already in place to help developers keep their dependencies up to date. See the Flathub Safety: A Layered Approach from Source to User blog post instead. The main thing to have in mind is that Flathub is not a commercial entity like other app stores. Right now, developers that put their apps on Flathub are (in the vast majority) not paid to do so and most apps are under an open source license. So any discussion that starts with "developers should update to the latest runtime or have their apps removed" directly contradicts the social contract here (which is also in the terms of most open source licenses): You get something for free so don't go around making demands unless you want to look like a jerk. We are not going to persuade overworked and generally volunteer developers to update their apps by putting pressure on them to do more work. It's counter productive. With that out of the way, how do we gently push developers to keep their apps up to date and using the latest runtime? Well, we can pay them. Flathub wants to setup a way to offer payments for applications but unfortunately it's not ready yet. So in the meantime, the best option is to donate to the projects or developers working on those applications. And make it very easy for users to do so. Now we are in luck, this is exactly what some folks have been working on recently. Bazaar is a Flathub first app store that makes it really easy to donate to the apps that you have installed. But we also need to make sure that the developers actually have something set up to get donations. And this is were the flatpak-tracker project comes in. This project looks for the donation links in a collection of Flatpaks and checks if there is one and if the website is still up. If it's not, it opens issues in the repo for tracking and fixing. It also checks if those apps are using the latest runtimes and open issues for that as well (FreeDesktop, GNOME, KDE). If you want to help, you can take a look at this repo for apps that you use and see if things needs to be fixed. Then engage and suggest fixes upstream. Some of this work does not require complex technical skills so it's a really good way to start contributing. This is probably one of the most direct way to enable developers to receive money from their users, via donations. Updating the runtime used by an app usually requires more work and more testing, but it's a great way to get started and to contribute to your favorite apps. And this is not just about Flathub: updating a Qt5 app to run with Qt6, or a GNOME 48 app to 49, will help everyone using the app. We want to build an App Store that is respectful of the time developers put into developing, submitting, publishing, testing and maintaining their apps. We don't want to replicate the predatory model of other app stores. Will some apps be out of date sometimes? Probably, but I would rather have a sustainable community than an exploiting one.

- This Week in Plasma: UI and performance improvements (2025/11/22 00:01)
  Welcome to a new issue of This Week in Plasma! This week there were many user interface and performance improvements — some quite consequential. So let’s get right into it! Notable New Features Plasma 6.6.0 Windows can now be selectively excluded from screen recording! This can be invoked from the titlebar context menu, Task Manager context menu, and window rules. (Stanislav Aleksandrov, link) Notable UI Improvements Plasma 6.6.0 With a dark color scheme, the blur effect now produces a blur that’s darker (ideally back to the level seen in Plasma 6.4) and also more vibrant in cases where there are bright colors behind it. People seemed to like this! But for those who don’t, the saturation value of the blur effect is now user-configurable, so you can dial it in to your preferred level. (Vlad Zahorodnii, link 1, link 2, and link 3) When clicking on grouped Task Manager icons to cycle through their windows, full-screen windows will no longer always be raised first. Now, windows will be raised in the order of their last use. (Grégori Mignerot, link) Did a round of UI polishing on the portal remote control dialog to make it look better and read more naturally. (Nate Graham and Joshua Goins, link 1 link 2, link 3 and link 4) When you open the Kickoff Application Launcher and your pointer happens to end up right on top of one of the items in the Favorites view, it won’t be selected automatically. (Christoph Wolk, link) The Kickoff Application Launcher widget now tries very hard to keep the first item of the search results view selected — at least until the point where you focus the list and start navigating to another item. (Christoph Wolk, link) Discover now uses more user-friendly language when it’s being used to find apps that can open a certain file type. (Taras Oleksy, link) You’re now far less likely to accidentally raise an unintended app when a notification happens to appear right underneath something you’re dragging-and-dropping. (Kai Uwe Broulik, link) KMenuEdit now lets you select multiple items at a time for faster deletion. (Alexander Wilms, link) The QR code dialog invokable from the clipboard has been removed, and instead the QR code is shown inline in the widget. This makes it large enough to actually use and also reduces unnecessary code. (Fushan Wen, link) Notable Bug Fixes Plasma 6.5.3 Fixed a rare case where KWin could crash when the system wakes from sleep. (Xaver Hugl, link) Worked around a QML compiler bug in Qt that made the power and session buttons in the Application Launcher widget overlap with the tab bar if you resized its popup. (Christoph Wolk, link) Plasma 6.5.4 Fixed a regression in menu sizing that got accidentally backported to Plasma 6.5.3. All should be well in 6.5.4, and some distros have backported the fix already. (Akseli Lahtinen and Nate Graham, link) Fixed a Plasma 6 regression that broke the ability to activate the System Tray’s expanded items popup with a keyboard shortcut. (Mikhail Sidorenko, link) Fixed a regression caused by a Qt change that broke the clipboard’s Actions menu from being able to appear when the configuration dialog wasn’t open. (Fushan Wen, link) Fixed a bug that could make the Plasma panel’s custom size chooser appear on the wrong screen. (Vlad Zahorodnii, link) Fixed a bug that could make the clipboard contents get sent many times when it’s being set programmatically in a portal-using app. (David Redondo, link) Fixed a memory leak in Plasma’s desktop. (Vlad Zahorodnii, link) Fixed a memory leak in the clipboard Actions menu. (Fushan Wen, link) KWin’s zoom effect now saves its current zoom level shortly after you change it, rather than at logout. This prevents a situation where the system is inappropriately zoomed in (or not zoomed in) after a KWin crash or power loss. (Ritchie Frodomar, link) Fixed a bug that made the optional Textual List representation of multiple windows in the Task Manager widget fail to get focus when using medium focus stealing prevention. (David Redondo, link) Plasma 6.6.0 Worked around a bug in some XWayland-using games that made it impossible to type text into certain popups. (Xaver Hugl, link) Clearing KRunner’s search history now takes effect immediately, rather than only after KRunner was restarted. (Nate Graham, link) With a very narrow display and a high scale factor, the buttons on the login, lock, and logout screens can no longer get cut off; now they wrap onto the next line. (Nate Graham, link) Frameworks 6.21 Fixed a bug that could confuse KWallet — when being used as a Secret Service proxy for KeePassXC — into becoming convinced that it needed to create a new wallet. (Marco Martin, link) Fixed two memory leaks

affecting QML-based System Settings pages. (Vlad Zahorodnii, link 1 and link 2) Other bug information of note: 4 very high priority Plasma bugs (Same as last week). Current list of bugs 34 15-minute Plasma bugs (Up from 31 last week). Current list of bugs Notable in Performance & Technical Plasma 6.5.3 Apps that use the Keyboard Shortcuts Portal to set shortcuts can now remove them in the same way. (David Redondo, link) You can now use Spectacle's Active Window mode to take a screenshot of WINE windows. (Xaver Hugl, link) Plasma 6.6.0 Made a major improvement to the smoothness of animations throughout Plasma and KWin for people using screens with a refresh rate higher than 60 Hz! (David Edmundson, link) Reduced the amount of unnecessary work KWin does during its compositing pipeline. (Xaver Hugl, link) When you delete a whole category's worth of shortcuts on System Settings' Shortcuts page, all the shortcuts get grayed out and cease to be interactive, and a warning message tells you they'll soon be deleted and gives you a chance to undo that before it happens. (Nate Graham, link) Frameworks 6.21 KConfig now parses config files in a stream rather than opening them all at once, which allows it to notice early when a file is corrupted or improperly formatted. This prevents freezes in several places. (Méven Car, link 1, link 2, and link 3) When using the Systemd integration functionality (which is on by default if Systemd is present), programs will no longer fail to launch while there are any environment variables beginning with a digit, as this is something Systemd doesn't support. (Christoph Cullmann, link) How You Can Help Donate to KDE's 2025 fundraiser! It really makes a big difference. Believe it or not, we've already hit out our €75k stretch goal and are €5k towards the final one. I'm just in awe of the generosity of the KDE community and userbase. Thank you all for helping KDE to grow and prosper! If money is tight, you can help KDE by directly getting involved. Donating time is actually more impactful than donating money. Each contributor makes a huge difference in KDE — you are not a number or a cog in a machine! You don't have to be a programmer, either; many other opportunities exist. To get a new Plasma feature or a bugfix mentioned here, feel free to push a commit to the relevant merge request on invent.kde.org.

- Web Review, Week 2025-47 (2025/11/21 10:46)
  Let's go for my web review for the week 2025-47. In 1982, a physics joke gone wrong sparked the invention of the emoticon - Ars Technica Tags: tech, history, culture If you're wondering where emoticons and emojis are coming from, this is a nice little piece about that. https://arstechnica.com/gadgets/2025/11/in-1982-a-physics-joke-gone-wrong-sparked-the-invention-of-the-emoticon/ Screw it, I'm installing Linux Tags: tech, linux, foss, gaming Clearly something is brewing right now. We're seeing more and more people successfully switching. https://www.theverge.com/tech/823337/switching-linux-gaming-desktop-cachyos Lawmakers Want to Ban VPNs—And They Have No Idea What They're Doing Tags: tech, vpn, privacy, law This is totally misguided… Let's hope no one will succeed passing such dangerously stupid bills. https://www.eff.org/deeplinks/2025/11/lawmakers-want-ban-vpns-and-they-have-no-idea-what-theyre-doing Learning with AI falls short compared to old-fashioned web search Tags: tech, ai, machine-learning, gpt, learning, teaching If there's one area where people should stay clear from LLMs, it's definitely when they want to learn a topic. That's one more study showing the knowledge you retain from LLMs briefs is shallower. The friction and the struggle to get to the information is a feature, our brain needs it to remember properly. https://theconversation.com/learning-with-ai-falls-short-compared-to-old-fashioned-web-search-269760 The Psychogenic Machine: Simulating AI Psychosis, Delusion Reinforcement and Harm Enablement in Large Language Models Tags: tech, ai, machine-learning, gpt, psychology, safety The findings in this paper are chilling… especially considering what fragile people are doing with those chat bots. https://arxiv.org/abs/2509.10970v1 Feeds, Feelings, and Focus: A Systematic Review and Meta-Analysis Examining the Cognitive and Mental Health Correlates of Short-Form Video Use Tags: tech, social-media, cognition, psychology Unsurprisingly the news ain't good on the front of social media and short form videos. Better stay clear of those. https://psycnet.apa.org/fulltext/2026-89350-001.html Do Not Put Your Site Behind

Cloudflare if You Don’t Need To Tags: tech, cloud, decentralized, web Friendly reminder following the Cloudflare downtime earlier this week. https://huijzer.xyz/posts/123/do-not-put-your-site-behind-cloudflare-if-you-dont Cloudflare outage on November 18, 2025 Tags: tech, cloud, complexity, safety, rust Wondering what happened at Cloudflare? Here is their postmortem, this is an interesting read. Now for Rust developers… this is a good illustration of why you should stay clear from unwrap() in production code. https://blog.cloudflare.com/18-november-2025-outage/ Needy Programs Tags: tech, ux, notifications Kind of ignore the security impact of the needed upgrades, but apart from this I largely agree. Most applications try to push more features in your face nowadays, unneeded notifications and all… this is frankly exhausting the users. https://tonsky.me/blog/needy-programs/ I think nobody wants AI in Firefox, Mozilla Tags: tech, browser, ai, machine-learning, gpt, mozilla Looks like Mozilla is doing everything it can to alienate the current Firefox user base and to push forward its forks. https://manualdousuario.net/en/mozilla-firefox-window-ai/ DeepMind’s latest: An AI for handling mathematical proofs Tags: tech, ai, machine-learning, mathematics, google That’s an interesting approach. Early days on this one, it clearly requires further work but it seems like the proper path for math related problems. https://arstechnica.com/ai/2025/11/deepminds-latest-an-ai-for-handling-mathematical-proofs/ Production-Grade Container Deployment with Podman Quadlets Tags: tech, systemd, containers, linux, system, podman Podman is really a nice option for deploying containers nowadays. https://blog.hofstede.it/production-grade-container-deployment-with-podman-quadlets/ Match it again Sam Tags: tech, regex, rust Nice alternative syntax to the good old regular expressions. Gives nice structure to it all. There’s a Rust crate to try it out. https://www.sminez.dev/match-it-again-sam/ 10 Smart Performance Hacks For Faster Python Code Tags: tech, python, performance Some of this might sound obvious I guess. Still there are interesting lesser known nuggets proposed here. https://blog.jetbrains.com/pycharm/2025/11/10-smart-performance-hacks-for-faster-python-code/ Floodfill algorithm in Python Tags: tech, python, algorithm, graphics This is a nice little algorithm and it shows how to approach it in Python while keeping it efficient in term of operations. https://mathspp.com/blog/floodfill-algorithm-in-python AMD vs. Intel: a Unicode benchmark Tags: tech, amd, intel, hardware, simd, performance Clearly AMD is now well above Intel in performance around AVX-512. This is somewhat unexpected. https://lemire.me/blog/2025/11/16/amd-vs-intel-a-unicode-benchmark/ Memory is slow, Disk is fast Tags: tech, memory, storage, performance, system No, don’t go assuming you can use disks instead of ram. This is not what it is about. It shows ways to get more out of your disks though. It’s not something you always need, but sometimes it can be a worth endeavor. https://www.bitflux.ai/blog/memory-is-slow-part2/ Compiler Options Hardening Guide for C and C++ Tags: tech, c++, security Good list of hardening options indeed. That’s a lot to deal with of course, let’s hope this spreads and some defaults are changed to make it easier. https://best.openssf.org/Compiler-Hardening-Guides/Compiler-Options-Hardening-Guide-for-C-and-C++.html The problem with inferring from a function call operator is that there may be more than one Tags: tech, c++, type-systems, safety The type inference in C++ can indeed lead to this kind of traps. Need to be careful as usual. https://devblogs.microsoft.com/oldnewthing/20251002-00/?p=111647 There’s always going to be a way to not code error handling Tags: tech, programming, safety, failure Depending on the ecosystem it’s more or less easy indeed. Let’s remember that error handling is one of the hard problems to solve. https://utcc.utoronto.ca/~cks/space/blog/programming/AlwaysUncodedErrorHandling Disallow code usage with a custom clippy.toml Tags: tech, rust, tools, quality Didn’t know about that clippy feature. This is neat, allows to precisely target some of your project rules. https://www.schneems.com/2025/11/19/find-accidental-code-usage-with-a-custom-clippytoml/ The Geometry Behind Normal Maps Tags: tech, 3d, graphics, shader Struggling to understand tangent space and normal maps? This post does a good job to explain where this comes from.

https://www.shlom.dev/articles/geometry-behind-normal-maps/ Know why you don’t like OOP Tags: tech, object-oriented I don’t get why object oriented programming gets so much flack these days… It brings interesting tools and less interesting ones. Just pick and choose wisely like for any other paradigm. https://zylinski.se/posts/know-why-you-dont-like-oop/ Ditch your (mut)ex, you deserve better Tags: tech, multithreading, safety If you’re dealing with multithreading you should not turn to mutexes by default indeed. Consider higher level primitives and patterns first. https://chrispenner.ca/posts/mutexes Brownouts reveal system boundaries Tags: tech, infrastructure, reliability, failure, resilience Interesting point of view. Indeed, you probably want things to not be available 100% of the time. This forces you to see how resilient things really are. https://jyn.dev/brownouts-reveal-system-boundaries/ Tech Leads in Scrum Tags: tech, agile, scrum, tech-lead, leadership Interesting move on the Scrum definitions to move from roles to accountabilities. The article does a good job explaining it but then falls back into talking about roles somehow. Regarding the tech leads indeed they can work in Scrum teams. Scrum don’t talk about them simply because Scrum don’t talk about technical skills. https://www.patkua.com/blog/tech-leads-in-scrum/ How to Avoid Solo Product Leadership Failure with a Product Value Team Tags: tech, agile, product-management I wonder what the whole series will give. Anyway I very much agree with this first post. Too often projects have a single product manager and that’s a problem.
https://www.jrothman.com/mpd/2025/11/how-to-avoid-solo-product-leadership-failure-with-a-product-value-team-part-1/ Bye for now!

- FAQs (2025/11/21 00:00)
  Table of Contents Import Tool My camera is not in the list of supported devices How to use a USB Mass Storage Camera Run Time Issues Thumbnail Generation Fails on Large Files Why are my RAW images so dark? Some Properties in Sidebar are Unknown or Unavailable How can I inform you about bugs and wishes? What can I do if digiKam always crashes when doing something? How to deal with UTF-8 encoding issues? ExifTool not Found at Startup HEIF Metadata Not Updated Database digiKam does not work when the album library is on a network share How to Import Tags from Another Program? How to Migrate a Database Between Different Computers? Why Low Performances When Processing Items in Parallel on Windows With MySQL or MariaDB? Customizations How to create a color theme? How can I change the default external video player? How can I change the text font size? How to change the single-click behavior on a thumbnail? How can I change the browser used with the More Info button from the map sidebar? Is digiKam available in my language? Setup Why Are Many Data Files Downloaded at Application Startup? Why Download of Data Files Fails at Application Startup? Where Are All Configuration Files Used by digiKam Located? Why Do macOS and Windows Warn About Non-Signed Bundles? Workflow Which file format should I use? How to Copy Files Outside the digiKam Collection? How Can I Ignore or Remove Multiple Detected Faces at Once in digiKam? Where Can I Find the List of Internal Keyboard Shortcuts? How Can I Display More Photo Properties in Icon View? Contribute digiKam does not compile digiKam Mixes English and My Native Language in the Main Menu A Section of the Documentation Is Incomplete or Does Not Exist Yet Skill Level: INTERMEDIATE
- Text Tool Phase 3 (2025/11/20 10:22)
  When I started work on the text tool, I had planned it in three phases. The first phase was going to be the on canvas editor. And I handled inserting and removing text, moving the cursor around, and creating simple wrapping areas. IME support and simple copy-paste was also handled during this phase. The second phase was about rich text editing. I created a text properties docker, implemented a font database, style presets, a glyph palette and went over each property individually to make 100% sure it worked. I wrote a bunch of blogposts about this in the past (text layout, fonts, OpenType, metrics, and more). Which brings us to phase three… Removing the old Rich Text editor and shortcuts. Previously, Krita used a dialog to allow rich text editing. The dialog happened because after several years of working on Calligra (which Krita was

part of for a good while), none of the core devs were confident in their ability to conjure a full-featured on canvas rich text editor in a short amount of time. Then, the plan became to have a SVG source editor as the main way to interact with text… But then we realized that SVG text is very verbose, even compared to the pretty verbose HTML. Furthermore, we only had a few weeks left, so we had to quickly put something together that could produce SVG output without being too alienating. The old rich text editor window, next to the preview on-canvas. In a dark mode, the theming issues become especially prevalent, here the text is black both in the editor as in the preview, but the background is dark grey. This was one of the core headaches with this editor. It's interesting to reflect back on this, we had assumed at the time that the artists using our program would be quite technically competent. Over the past few years however Krita has picked up so much steam that it frequently ends up being the program to teach people that there's such a thing as "working memory" and that one can run out of said memory when they, for example, try to make a 60 fps animation at 4K resolution. The rich text editor also didn't help matters here because the conversion to and from SVG text wasn't optimal: SVG 1.1 doesn't have a concept of lines or line wrapping, being more a graphical description format like PDF than say, a rich text document. Furthermore, there were endless issues with theming, font size handling, differences between Qt's rich text implementation and HTML overall, fonts, you name it. So, given that the first two phases left us with a functional on-canvas text editor that can style text, it felt good to remove the rich text editor. The source editor remains, because it can still do some advanced tricks if you know SVG very well, but for the vast majority of text tasks it is unnecessary. That left the shortcuts. Because the rich text editor was implemented with KXmlGui, each of the entries in the toolbars was a QAction that could be configured. I went and implemented the majority of those as shortcuts in the text tool. Because the majority of these shortcuts just changed a single property value, I took inspiration here from how the old artistic text tool was implemented, which is to say, using the setData() function on QAction. The setData takes a struct that contains simple instructions on which property to edit, and how to edit it (increase it, or set a specific value, for example), and this is then used to test the toggledness on the action or generate the appropriate property setting command. The shortcuts are a little interesting in that the text tool needs to do its own shortcut matching, instead of using the global system. This is because the text tool needs to be able to discern text input from shortcuts, as well, all QKeySequence::StandardShortcuts for text navigation and selection have been implemented directly into the text tool. The latter needed to be handled directly because if you are working with vertical text you will want to have the up/down keys be for navigating forward/backwards in the text. In practice, this means that the tool first tests non-modifier arrow keys and basic input. Then it checks the direction and writing mode of the text, and ensures that any directional keys get replaced by their expected variants for said direction and writing mode. It then checks the shortcuts and finally the standard sequences. This latter order is because some standard sequences use the same shortcuts that are typically reserved for property changes (for example, deleting a line and setting underline can both have the Ctrl + U shortcut). Only a few pre-existing shortcuts are supported right now, because I was feeling the end of the project looming. It is also a very good starting point for people who are interested in hacking on the text tool, as it makes you think about how the text properties function, so I have left it alone for now. Replacing the tool options. Next up was replacing the tool options. Tool options within Krita are specifically for changing the behaviour of the current tool, as well as accessing extra functionality, and Krita's text tool had one that allowed you to set some properties for new text creation, as well as accessing the separate editing dialog. The old tool options with a "create new text with" group of widgets and an "Edit Text" button. This was a QWidgets based UI element, and one of the things I've been doing with the text tool is that every UI element would be written in primarily QML, using QWidgets only as a fallback. Beyond that, we've been trying to use Lager to keep track of data-editing, and tool configuration is a good candidate for a lager model. The new options are both longer, because there's more toggles, but at the same time also simpler to use… As for

the options themselves. Like the old options, they allow selecting the default text that new texts are being created with. However, instead of replicating the text properties docker in its entirety, it now provides a drop down to select a style preset. Beyond that, there's a toggle to use the current presets in the text properties docker. The text properties docker itself also has a button to open it inside the tool options. This is because I observed multiple people trying to use the "new texts are created with" options (that were labelled as such) in the old widget to manipulate the selected text. Hopefully this'll guide people to the text properties docker. Then, there's the two options for the text tool itself: "Paste Rich Text By Default" and "Use Visual Cursor". The former is about whether Ctrl + V should paste the rich text or the plain text, while there's separate actions for pasting plain or rich text explicitly. For some reason the majority of word processors always paste rich text by default and never allow you to configure this, even though this is not how anyone wants to use a word processor. It therefore made sense to make this a toggle. The other is for bidirectional text. The default is the "logical" order of the bidirectional text, that is, the order in which you read it. This has always been a little bit of a headache to programmers, because you end up with a cursor that skips positions and will in some cases go into the opposite direction of the key you're pressing. This is why a lot of programs offer a "visual" cursor, which will try to follow the direction the keys are pressed in. This too is configurable, as it depends per person which of the two is more intuitive. Finally, there's several buttons, some for opening the dialogs like the Glyph Palette or the Source Editor, others for toggling Type Setting Mode and finally a set of buttons for the converter actions. More on these latter two in a bit. Annoyingly, QML cannot handle popups properly while used together with a QQuickWidget: the popup is clipped by the QQuickWidget bounds (specifically, it's internal QML scene). In Qt6 there's a toggle to not have Pop ups clipped to the scene, but that crashes in QApplication when used inside a QQuickWidget. This is a problem because it makes it hard to provide the style preset dropdown (or the font dropdown for that matter). I am trying to find a solution, but in the meantime I've made the delegates much smaller. The CSS style presets now have much more compact delegates to handle being inside a dropdown. I hope I'll be able to fix the pop-up problem in the coming weeks though, as this is kinda annoying. PSD text and vector support. This was a somewhat tangential project, and I had written most of it even before I had written the on canvas text editor. PSD vectors are stored as a vector mask on a fill. Because I had previously already written support for the fills, supporting vector masks meant supporting vector shapes. Then, figuring out which layer data describes the stroke, and which describes 'meta shapes' like rectangles, ellipses and stars was the rest. The vector masks, in particular PSD's path format, are interesting in that they have a particular floating point format. Luckily, Scribus already supported loading those paths, so I was able to reference their parsing code and concoct writing code based on it. Most of the work by far was making sure the transforms were 100% correct. I was helped here by Deif Lou, who provided me with a ton of test files to check against. Text was not as simple. Text in PSD is stored as a PDF structure, which, if you are unfamiliar with PDF, is format not dissimilar to JSON. You can see the way text is stored by opening up a PSD with text as a text file in a given simple text editor. Unlike most PSD data, which is stored in binary (and is best inspected with a hex-editor), the text data is largely stored as ascii text (with actual strings being stored in 16bit BE). Some inspection reveals that the text itself is stored in a range based manner. The whole plain text is written at the start, and then a list of character "sheets" is presented. After all those, a list of numbers of equal size, each presenting what range is occupied by the sheet with the same index. Idem for paragraph properties (PS, unlike SVG 2, can have multiple paragraphs. SVG 1.2 did have multiple paragraphs, but there's only one real implementation of SVG 1.2 (Inkscape) due its complexity). PSD text is also, notably, stored and sized in pixels, regardless whether the format is saying points (this is because within the Apple ecosystem the two units are the same, while outside it, a digital point is always 1/72th of an inch). But there were still a number of mysteries. Like, where were the text paths stored? And how about the advanced OpenType features? After I asked on Krita-Artists for sample files, I found the answer: There's another chunk of advanced

text data at the end of the document. This second chunk is far more complex, and, worryingly, it uses numbers for the keys for more recent versions of the Adobe products. Thankfully, it turns out that this second chunk is in fact shared between the Adobe suite, and the Inkscape project had already done a lot of reverse engineering. While I am now able to read this data and load text from it, there's still some missing information. It seems there's like, Line and Cluster-specific positioning and glyph info present in this extra data, and if you write it without said data, Photoshop will complain it is missing. I haven't had the time to look into this properly, so we can't write advanced text objects as of now. Another wrinkle is that text and vectors are per-layer in PSD, while in Krita each vector layer is its own SVG document. So I also had to write some code to explode the Krita vector layers so each shape is written as a separate PSD layer. Similarly, despite being able to load a lot, Krita's SVG+CSS based text layout is fundamentally different from Adobe's, so text loading isn't perfect. None the less, it should provide valuable for people's archive. Because of these differences, Krita will ask whether you want to load the text layers as text shapes or whether you want to load them as raster data. I do want to eventually return as figure out that final missing data. As well, because PSD doesn't do inheritance, I need to fiddle a bit with selecting which properties to set on the paragraph, as the paragraph metrics are important for Krita's baseline aligment. Text on Path/Text in Shape. Text on path and text in shape were actually going to be tackled in the first phase. But then when Alvin, who was helping me, tried to take a stab at it, he was blocked because Dmitry was extremely unsure about the design. This was a bit frustrating, because I had already made sure that the layout algorithms for both worked fine. I myself decided to focus on getting rich text editing to work, and it wasn't until I was nearly done with rich text editing that we returned to the discussion about text on path and text in shape. I never talked about text in shape previously, as it was implemented after I wrote the big text-layout blog post back in 2022. So lets do that now: The current look and feel for a complex text-in-shape while editing the text. Yellow ellipse, pink triangle and blue polygon are "Inside" shapes. The green rotated rectangle is a "subtract" shape. Both shape padding and margin are applied. The arrows indicate the order of the flow shapes. Icon on the top right is a button for contour mode. SVG 2 allows wrapping text in shapes, and has some sophisticated toggles to configure how the text is flown into shapes. The simplest is a single text in a single shape with a single subpath. However, those single shapes can handle multiple subpaths (in which case the line is broken up), and there can be multiple shapes that text can flow into, one-by-one (much like CSS columns). These flow areas can in turn have other shapes subtracted from them. Finally, shape padding and margin can be used to modify the distance of the text to the related shapes. Most text-in-shape layout algorithms will do so by taking the shape, drawing line boxes from top to bottom, and fitting the text in the first reasonable line box. This is what SVG 1.2 specified for it's text in shape. For SVG 2 however, the first line needs to sit snugly against the border of the text wrapping area. For this I implemented an algorithm described by Hans Muller, originally devised for CSS shapes. Once we have the first position, we want to create a line box. One problem here is the question "how tall is the line box?", especially in rich text: Text can have different font sizes, and different font sizes can lead to different line widths, depending on whether those differently sized sections get onto the line or are wrapped to the next. The solution here is to estimate the line height: We check all glyphs that might fit in the next shape-bounding-box width (or height for vertical), and get the max line height from this. Then, the line box is determined, with a single line being able to hold multiple line fragments if the shape boundaries cut through the line. Text is then laid out onto these lines (logically if we take bidirectional algorithm into account), breaking where it is allowed. Finally, text is reordered so it lays out visually, text alignment and justification takes place, and the final line height is calculated. The whole text is then shifted (block wise) upwards by the difference between the estimated line height and the actual line height. I was able to figure out this solution because I kept trying to figure out what Inkscape was doing here, and was able to induce a bug that suggested it does something similar. Said bug got reported. Some oddities are present in the SVG 2 spec here. For one, it

doesn’t ever say whether to include the local transform of the shapes that is being flowed into. Text on path does require this, and usage would become incredibly annoying without it (if you want to flow text into two rectangles, you will need to apply a transform to one of them), which means it is expected, but its absence is very odd. Inkscape does do this, so I implemented this as well. Another thing that can be odd is that because the shapes get linked to the text shape, it is possible for the text to be rotated and be out-of-sync with the linked shape. The only way to have both rotate together is when they are in the same group. In the context of SVG this linking behaviour makes sense. You can easily imagine a magazine layout where a few rectangles provide columns, and a pull-out quote is laid out into a circle. Said circle then overlaps these rectangles, without overlapping the text. However, while that makes sense in a magazine context, there was a worry that it might be too complex to interact with. Especially because Krita doesn’t have an object outliner like Inkscape or other specialized vector applications have, so going in and out of groups can be frustrating. Eventually we decided to make it so that Shapes that text flows into is always a child of said text. The text is then stored as a group with shapes and text inside SVG, making it 100% compatible with Inkscape, while within Krita we could simplify the interaction, while keeping all the powerful transformation features. There’s a downside to this method though: When we resize text in shape, we resize the child shapes. However, the child shapes affect the slow text layout (in particular, the shape-offset operations, which are quite slow), but because the child shapes are children of the text, it becomes next to impossible to update this text independently from the resized shapes. Very frustrating because I did pay a lot of mind to keeping the text layout thread-safe, so had we stayed with a linking model, the text shapes could’ve just been sent to another thread to sort themselves. Thankfully Dmitry decided to take responsibility for taking care of this slowdown. We now block text layout during resizing and update it after the fact. Because of the way bounding rects are calculated however, this does mean we cannot afford to have any text drawn outside the text shapes, which means we’re forced to have our overflow to be always clipped (and truthfully, there’s no clear answer to what overflowed text should look like with SVG 2 anyway, so maybe it’s for the best…). UI wise, the simplest way to set text in shape is by clicking a shape. Clicking on a border will instead set the text on path, and set the click location as the starting point. This is necessary because right-to-left text needs to be aligned with the end of the path due the way text on path interacts with text anchor in SVG. The less simple but advanced manner is to use the context menu in the Shape Selection tool to flow texts in Shape. This method allows for setting up a complex flow structure. Similarly, the default tool allows changing the flow shape order, and setting subtract shapes. Both this and the previous method are ways in which such shapes are set up in other programs, so people should be able to find either without consulting the manual. When a text-in-shape or text-on-path is created, a new button appears that can be clicked to go into contour mode. There, each contour shape can be manipulated as needed. Within the text tool, text on path gains a handle to move the start offset, while text in shape allows dragging the text area to set the shape padding and margin. Text-on-path doesn’t have an advanced mode like text-in-shape, right now artists will only be able to create texts with a single text path. However, Krita’s text layout can handle multiple text paths in a single text, and even a mix of text and positioned paths. That kind of thing is currently limited to the SVG source editor, as I ran out of time to ensure that the interaction would be nice. Something for the future. Type Setting Mode Type setting mode is a separate mode in the text tool that allows for on-canvas fine tuning and interaction with font metrics. It differs from regular editing mode in that it will show editable font metrics when activated. When the text doesn’t auto wrap, it even shows handles so that the SVG character transforms can be modified over the selection. Type Setting Mode is kinda interesting in that at first glance it seems like an unnecessary toy mode. After all, if you want to edit the font size, the text properties docker is much more suited, right? Yet, when gathering input about what artists needed from the text tool, some expressed that they wanted to be able to edit things like font size and line height by on canvas widgets. Others protested: it would interfere with text editing, which

seemed a reasonable concern. So it was clear that if such a thing would be introduced, it needed to be optional. Then there was the issue of the Baseline features. Krita is currently one of the rare text layout implementations that implements alternate baseline alignment. But the baselines are kind of abstract, especially as font makers rarely fill out the OpenType BASE table from which these baselines are derived, meaning they frequently have to be synthesized. So there was also a need to allow people to see the available baselines on canvas. But there was one final issue. Let's talk about kerning. If we conceive of text as being comprised of glyphs, and each glyph can fit onto a little rectangle, like in (movable type) print, and we imagine printing with this. Then it's very likely that there will be huge gaps between the glyphs while printed. Therefore, font makers would make the base rectangle smaller and let parts of the letter overhang, a so-called "kern", so the glyphs would interlock a bit more elegantly. Movable type was never the only text printing technology. For posters for example, lithography was widely used, and the text printed with lithography was typically hand drawn by the artists. This meant that artists would be able to manually decide the best spacing for a given piece of text. Then, there's a number of in-between technologies. There was a particular one where designers would work with letter sheets than could be transferred onto a given piece of paper, and I seem to recall there's a similar technique that relied on clever use of photocopiers. The precise technologies aren't very important here, but rather I want to impress that there's a western practice of spacing glyphs in a text just right, and that the underlying technology greatly affects what is possible. As such, this practice is taught to students of design, and seen as one of the important details that distinguishes a well done piece of typography from a rush job. For the western typographer, to get the kerning just right is to say you care. Now, in the digital era, font makers are able to very quickly define kerns for any pair of glyphs, and while doing text layout the shaper will apply these kerns. This is generally good enough for the majority of use cases… However there's a technological limitation. See, if you do rich text layout, you first need to itemize the text into ranges where the font, direction, and script is the same before you hand it over to the shaper to shape. In the above example, you can see that the first letter is much larger than the rest, and there's no kerning. This is because the font size is different, and thus, during itemization, it's a different font, and a different glyph run. The shaper cannot apply kerning between these two different runs. Typically, this is worked around by adjusting the tracking or kerning. CSS however, only has letter-spacing, which is meant to be applied to ranges of text. Meant. In practice, the majority of implementations make it so that letter-spacing modifies the spacing to the right of clusters of glyphs. But not all: some implementations do it to the left when text is right-to-left! If that weren't enough of a headache, right now, the CSS working group is changing the way letter-spacing works all together. Using letter-spacing for this is not much of an option then. The CSS-WG suggests that if you really want to do manual kerning, you need to create a special span with reduced margins, as this will give the most control. SVG doesn't have margins though, as SVG doesn't have the CSS box-model. But we do have character transforms in SVG. Character transforms in SVG have been there since the beginning. There's 5 parts to character transforms: absolute x and y, relative dx and dy, and rotate. Absolute X and Y set the current text position in absolute coordinates to the text origin, and, these break shaping, much in the way line fragments do with auto wrapped text. When the SVG text specifications talk about text chunks, it means ranges of text that have been positioned this way (and since SVG 2.0, also other forms of line fragments). Dx and dy conversely, accumulate, starting from each text chunk start. They don't break shaping, nor does rotate, which means these three are very suited for this need for manual adjustment. They just needed to be editable. So when I was looking at these three issues (on canvas adjustments, baseline selection and character transforms) and was deciding on my design, it became clear we needed this separate mode to handle these three things, and also that it wasn't all that optional: spacing and kerning are a pretty important practice after all. I then spend 10~ days to get the character transforms right. This was because I decide it would be more useful to calculate the relative positioning from absolute positions rather than to set the relative positions directly. This way, I would only

need to calculate where I want the glyph to be, and let the function itself sort what kind of delta positioning that requires. This required me to kind of backtrack from the final position to calculate the point at which the delta x and y are added. This was quite tricky as there's a lot of modifiers on where a particular transform ends (ligatures, utf32 vs utf16 codepoints, and of course, white space collapse), as well as going backwards, as that involves removing the text-path adjustment, text anchor calculation, absolute offset and finally the textLength offset. The actual editing of this is provided by two handles at either side of the active selection. Dragging the square handle offsets the whole selection, while dragging the round handle scales and rotates the selection, using the square one as the hinge/origin. I am not fully sold on how this is handled, especially in RTL, and I want to see if I can handle the offsetting better. With type setting mode, you can transform each glyph individually, allowing for some pretty advanced looking typography. The lines are the metric lines. Hovered line is the Descender, hence it saying "Font Size". Next up was changing on canvas properties. This is right now, limited to Font Size, Baseline Shift and Line Height. The baseline shift is modified by dragging the baseline, the font size is modified by dragging either the ascender or descender, and the line height by moving the line height markers, which are ascender and descender + the line-height on either side. Artists will be able to tell which they're modifying by the hovering name. Krita slogan in Hindi. Here, we use type setting mode to adjust the size of the second word, and then select the "hanging" baseline to align the text to the head stroke. Setting the dominant/adjustment baseline can be done by pressing Shift, which switches the visible lines to the baselines. Clicking them will set that as the dominant baseline. There's still an issue here with overlapping lines, and I need to sit down and think about which lines should have priority. This is pretty useful already, but there's still a number of unanswered questions: Right now, I add a counter transform at the end, this is because when using it, the counter transform felt more intuitive. However, it can also make sense to not have that. Maybe it needs a toggle? Similarly, the scaling/rotating code can easily only do scaling OR rotating, and it makes sense to use either. But I am unsure how to provide that in the UI. Thirdly, there's right now no way to set the Absolute transform. I got some functioning unittests for it, but some edge cases look weird and it needs more work before I can expose it to the UI. Right now, the Font Size, Baseline Shift and Line Height are all adjusted in Points. I was unsure whether to have them in relative font size, and we'll need to see if that's something people prefer. I can imagine that some people would like to see a line over the x-height, but the thing is that there's no real related metric for that. There's font-size adjust, of course, but newer versions of font-size-adjust are also possible against the capital height, or even the ideographic height. So I just left it out for now. The metrics that can be adjusted are all of a certain type, what western Typographers call "vertical metrics". There's no controls yet for Tab size, text indent, word spacing and letter spacing, though they could be easily imagined. In a similar vein, one could imagine handles for italic/slant, weight or width. These I have been avoiding because of these, only slant can be predicted, the other two are unique to the type face. Finally, there's of course associated shortcuts. I've implemented four for moving the offset in any of the four directions, but none for scaling/rotating. This is because I assumed people would definitely want to offset with the keyboard, but was unsure about the others. Part of these are because Type Setting Mode came in very late. It was always going to be added as last, because, from a surface level it sounds like a frivolous toy mode. When I expressed my intent to create it, some artists even told me they were never going to use it. Not strange: There has been over 30~ years of digital type setting that didn't need a separate type setting mode. But once you see the whole picture, and more specifically, realize that not all text layout systems are the same, the purpose starts to make a little bit more sense. Whats more, because SVG character transforms have been there since the beginning of the SVG spec, they're pretty widely supported, so it'll be very interesting to see what people will come up with. There's still a snag though: SVG relative character transforms don't apply on auto-wrapped text. There's a little note in the SVG 2.0 spec that these were considered, but ultimately seen as unnecessary. Little bit annoying, but not the end

of the world, as someone who aims to wrap in shape, but then fine tune, can do just that. Wrap in shape, convert to pre-positioned text, and fine tune the spacing… Let’s now finally talk about the converters. Conversion actions for text types. Because previous versions of Krita only supported SVG 1.1 text, it was important that there would be a way for people to convert away from that format. Similarly, if someone had put text-in-shape, or created an Inline wrapping area without intending to, there needed to be a quick way to convert. Converting away from “Pre-Positioned Text” (SVG 1.1 text with white space collapse), required first removing all collapsed white spaces. Then, inserting new lines for each SVG text chunk with absolute positioning. All of this needed to be done in reverse because insertion and deletion changes the indices. By going in reverse the indices that still needed to be modified were kept the same until modification. For inline size, the inline-width of the text is tested before conversion, and set after conversion. Converting towards Pre-Positioned text on the other hand was much easier, as it was a case of figuring out the current position and making sure it was being set as an absolute transform. Because we are just working with the layout results, we can convert from text-in-shape to pre-positioned and keep the lines positioned the same. The leftmost polygon here is a text in shape, while at the right, the text has been converted to pre positioned SVG 1.1 text. The difference is only visible because of the different selection rectangles. These actions were implemented in the text tool for single shapes and in the shape select tool for multiple ones. Wrap up And that was all of Phase 3, which means I am done for Krita 5.3. As of writing, we’re in feature freeze. This means I will be focusing on fixing bugs in the coming few months. But also writing documentation, and release notes. I am slightly worried, as I didn’t get a lot of feedback near the end, and am left wondering what kind of bugs I will see in the coming months. Some things didn’t get in from the original plan. Most notably: color and stroke setting. Krita can set these things, the controls for it need to be ported to QML, and I was told to avoid it, as it would be too big. Due these missing controls, stroke can only be set for the whole text, while the fill can only be a color for a selection, or gradient set on the full text. Beyond that, a visibility mode for formatting marks (that show where the spaces are and what type of spaces they are) didn’t get in either. It’s by itself not complex to implement, but it needs good design of the marks, and I just didn’t feel like I could give it the attention it deserves because of the amount of work that went into text in shape. It is also not yet a full implementation of SVG 2. Text-orientation is the biggest missing element here, but I told myself I wasn’t going to work on that until there was a decent enough text editor. There’s also things like better justification, hyphenation and color font support, but those were never going to be in 5.3. It’s going to be interesting to see how the usability is going to be tweaked over the coming years. A good number of properties can be found directly on the canvas and the main dockers, so I do feel everything is pretty discoverable. However, I did have to put the advanced text-in-shape actions into a right click menu, and generally people don’t find those. It will also be interesting to see what people will do with the type setting mode and how that’ll evolve. I know people want an on canvas property editor, but I had been holding off on that because the design would be tricky to get right, as well, in Qt5 Android platform integration doesn’t yet obey the enum value that asks for the copy-paste menu to be hidden. So that menu would probably float over any on-canvas property menu. Qt6 fixes this, but I don’t know if it does for Apple products as well. Overall, this was a pretty ambitious project. One thing that probably didn’t bleed through in all these blog posts is that probably at a least a third of the work was communication. I myself understood that when I started it, but I think it has a tendency to get lost when you read tech blogs, so I’ll expand a little about it. Basically, every step I took, I spend some time talking with the other Krita developers (primarily Dmitry, who reviewed all my work) what I wanted to do, and how I was going to approach it. This isn’t just to get the design sorted, but also to avoid blind-siding people. I also wrote these blog posts, and wrote little feature introductions for the people on Krita artists. The purpose of the latter was to get people to understand with what I am trying to do, I am unsure how successful I was there. The technical blogposts did appear to be pretty helpful, and I got a lot of feedback from other FOSS people that these blog posts were

useful. While text layout is not a rare programming topic, advanced text layout is only really done by a handful of people, and I imagine a lot of them are too exhausted after they got the thing to work to write a blog post about it. Anyway, when I first started, my colleague Agatha had described text layout like a “Hydra” and that every update I made felt like I was chopping those heads off one-by-one. I declare the Hydra dead: Krita has a decent text tool now. Appendix SVG Character Transforms. Just checking how widely supported the SVG character transforms are… Inkscape and Chromium do pretty well. Firefox does a little bit odd. SVG Tiny 1.2 does include the character transforms, but QtSvg doesn’t support them (or Gwenview isn’t using QtSVG). Then there’s a lot of epub readers that have varying degrees of support. The sample SVG-based epub3 files do use multiple character transforms, but that by itself doesn’t mean much. KoReader for example only supports the first transform, even with text-on-path, which is interesting given that LunaSVG, which it uses, does seem to have support for it if you look at the code. KoReader does apply textLength, which is useful, but then other epub readers I’ve tried don’t apply textLength, but only transforms. It’s a bit of a mixed bag, but the wide browser support is heartening (good enough if you just want to tweak spacing while maintaining an accessible SVG text element). Our “The Two Towers” sample in Firefox 145. Firefox doesn’t support baseline shift (but does support white-space, so we can preserve the line break… but then doesn’t support SVG units, so all units need to be suffixed with px) Chromium 142, it supports SVG 1.1 fully, but no white space, so I had to convert to pre-positioned text for this to work. Neither browser supports optical size in SVG, so the large capitals are less delicate than in Krita. The sample in Inkscape 1.2, I had to convert the font from Amstelvar to DejaVu Serif, as it didn’t seem to like Amstelvar (probably because it is a variable font). It’s by far more widely supported than SVG 2 text wrapping though, of which the only known (to me) implementations are Inkscape and Krita. Ideally, Krita, like Inkscape, would ensure there’s fallback positioning written. The absolute transforms are 100% intended to provide a fallback when the auto wrapping is not yet supported. The reason Krita isn’t doing this is because it needs a separate code path so it only saves this to exported SVG layers, and as well, I am not 100% confident in my conversion code. When that time comes it might also prove useful to save the textLength, but I’m still mulling over this. A final note is that Krita, like Inkscape, has a “convert text to paths” function. When converted to paths, text can be modified as desired, but then it loses the accessibility of being text that can be selected, which you’d want to avoid in an interactive environment.

- From Commodity Trap to Sustainable Innovation: Convincing Automotive Executives to Embrace Open Source (2025/11/19 06:00)
  Automotive companies spend too much on commodity software. Co-creating multiplies their investment through community collaboration. This frees resources for true differentiation. Read about, the concepts and ideas to master, that support this strategic journey.
- Design System Progress – November 2025 (2025/11/18 00:13)
  For the past few weeks, we have been working on a few areas around the design system for Plasma. Keep in mind that I am only speaking of the graphic side, not code. Work is ongoing with the Union engine and the team is focused on replicating our current Breeze style using Union. There have been talks about creating the first components based on the design system, but that is more in the future. Meeting with PenPot We held a meeting with the PenPot team and Pablo Ruiz, their CEO, met with us to discuss new changes in the PenPot app. This was a follow up to their recent conference PenPot Fest. Their team announced a few things that should make it much easier for the Plasma Design team to adopt PenPot. For example: New composite token: Typography Taiga #10200 Show current Penpot version Taiga #11603 Switch several variant copies at the same time Taiga #11411 Invitations management improvements Taiga #3479 Alternative ways of creating variants – Button Viewport Taiga #11931 Reorder properties for a component Taiga #10225 File Data storage layout refactor Github #7345 Make several queries optimization on comment threads Github #7506 With these additions, it was much easier to move assets into PenPot than before. There was less work we

needed to do. We begun a migration to PenPot for the second layer of basic components and also started building more complex components. Here are some screenshots: Buttons Button Groups Badges Inputs Dropdowns Toggles Checkboxes Checkbox Groups Avatars Tooltips Progress Indicators Sliders These components are shared components. Then we moved into application components and this is what we have so far. Application Components Modals Pagination Tables Video screen (Miscellaneous) Breadcrumb Tabs Alerts and Notifications Date Pickers File Upload Section Headers Content Dividers In this list, you see a lot of graphics. Each of these is supposed to represent a different state of the graphic. Users wouldn’t work with these variant sets very much, instead, they would simply search in the component catalog for what works in their design and only edit organization and labels. However, to get to that level, the designers need to create interpretations of each of these states graphically. This leads to a lot of work and a lot of graphic memory usage. There are a few more components that can be created. However, given PenPot’s reliance on the browser DOM, the more complex the components, the more lag the application runs into. Because of this issue, we have contacted PenPot to become beta testers of their new rendering engine when it comes out. They are almost at the point where they can put this out. We are eager to try and see how much faster we can go. The issue is not on PenPot but the engine that powers the editing screen. Still, we have to wait a little bit to continue. In the mean time, we can dedicate ourselves to making more application icons and completing the work there. FOSDEM Additionally, we are setting up a workshop with the PenPot team during FOSDEM 2026. This workshop will focus on brainstorming ideas on how to more easily distribute and contribute to a design system using PenPot. For example, there is a list of ideas we proposed: Exclusions and inclusions into the design system library. This way, the original copy of the design system remains consistent with the base components unalterable. This should make it easier for casual designers looking to build a quick mockup without getting bogged down by sub components that don’t need edits. This can also ensure that the many users taking the components are using a consistent copy to the original. Automatic sharing and updating to users not in the immediate instance team. Generate a review system for components as external users to the main instance propose changes. An easy way to re-publish the design system after applying suggested changes. …and a few other ideas. Hopefully, there are good ways to get this done. We are still waiting to move our icons into PenPot. Likely, this is more of a reality once the new rendering engine is in place. The team let us know that there are a number of shape manipulation improvements app In addition to all of these changes, we keep submitting bug reports and feature requests to the PenPot team to make the app even stronger.

- KDE Plasma 6.5.3, Bugfix Release for November (2025/11/18 00:00)
  Tuesday, 18 November 2025. Today KDE releases a bugfix update to KDE Plasma 6, versioned 6.5.3. Plasma 6.5 was released in October 2025 with many feature refinements and new modules to complete the desktop experience. This release adds two weeks’ worth of new translations and fixes from KDE’s contributors. The bugfixes are typically small but important and include: View full changelog
- This Week in KDE Apps (2025/11/17 22:10)
  Crop tool in Photos, Sudoku in Kirigami and sprintingWelcome to a new issue of "This Week in KDE Apps"! Every week (or so), we cover as much as possible of what's happening in the world of KDE apps. Last Saturday a bunch of KDE devs (and a guest) met in my kitchen for a "Kitchen sprint". As always, we discussed and worked on quite some exciting stuff, mostly around Itinerary and public transport infrastructure in KDE, but not only. Here is a short overview of what some of us worked on: Jonah experimented with integrating maplibre in our apps, Nico demoed his new online account integration for applications, and, outside of cooking some Käsespätzle for the whole group, I spent some time packaging Merkuro as a flatpak! Outside of that, and as part of our end-of-the-year fundraiser, you can adopt one of KDE's apps and we can share with the whole world how awesome you are and how much you're doing to support us. Thanks to everyone who already donated, this is super helpful! Getting

back to all that's new in the KDE app scene, let's dig in! Multimedia/Graphics Applications Photos Image Gallery Noah Davis added a crop tool to the image editor of Photos. (25.12.0 - link). Joshua Goins improved the performance a bit in the main view (25.12.0 - link). Sytem Applications Dolphin Manage your files Nate Graham reverted a change which impacted keyboard-driven folder manipulation (25.12.0 - link). Oliver Schramm fixed trashing files from temporary folders. Now they no longer end up in your home trash bin. (KDE Frameworks 6.22 - link) PIM Applications Merkuro Calendar Manage your tasks and events with speed and ease Tobias Fella fixed setting the calendar name (25.12.0 - link). He also disabled the calendar editor when we don't have permission for it (25.12.0 - link). Social Applications NeoChat Chat on Matrix Tobias Fella simplified the process to unlock the key backup by providing only one text field (26.04.0 - link) and it is no longer behind a feature flag (link). Tokodon Browse the Fediverse Loïs Rioul fixed login with GoToSocial (25.12.0 - link). Games Pumoku Anders Lund pushed the first early alpha version of his Kirigami based sudoku application called Pumoku. It is still a bit basic but very promising. Third-Party Applications Easy Effects - Audio Effects for PipeWire Applications Wellington Wallace released Easy Effects 8.0.3 containing a bunch of fixes for regression from the major 8.0.0 release. Giusy Digital fixed some translations issues in the spinboxes (link) and the number validator (link) Carl Schwan ported the settings to KirigamiAddons ConfigurationView (link) Carl also fixed various spacing issues in the effect pages (link), ported the navigation menus to normal tool buttons (link), ported the application metadata to KAboutData and FormCard.AboutPage (link) and various other small graphical changes. …And Everything Else This blog only covers the tip of the iceberg! If you’re hungry for more, check out Nate's blog about Plasma and be sure not to miss his This Week in Plasma series, where every Saturday he covers all the work being put into KDE's Plasma desktop environment. For a complete overview of what's going on, visit KDE's Planet, where you can find all KDE news unfiltered directly from our contributors. Get Involved The KDE organization has become important in the world, and your time and contributions have helped us get there. As we grow, we're going to need your support for KDE to become sustainable. You can help KDE by becoming an active community member and getting involved. Each contributor makes a huge difference in KDE — you are not a number or a cog in a machine! You don’t have to be a programmer either. There are many things you can do: you can help hunt and confirm bugs, even maybe solve them; contribute designs for wallpapers, web pages, icons and app interfaces; translate messages and menu items into your own language; promote KDE in your local community; and a ton more things. You can also help us by donating. Any monetary contribution, however small, will help us cover operational costs, salaries, travel expenses for contributors and in general just keep KDE bringing Free Software to the world. To get your application mentioned here, please ping us in invent or in Matrix.

- A new online accounts system? (2025/11/15 09:00)
  For many years Plasma comes with its own system online accounts system, known as KAccounts. The idea is simple: In Systemsettings you log into a given online service once, and then several applications can use that login, instead of loggin in inside each application separately. The number of services available and applications making use of them changed a bit over recent years. As of right now the following services are supported: Nextcloud: This is used by Dolphin to add a shortcut for file access via webdav to the Network section, as well as the Purpose framework to allow uploading files to Nextcloud. Owncloud: Used for the same things as Nextcloud. Google: Used by Purpose for uploading to YouTube. In theory also used by kio-gdrive for browsing Google Drive, but access to this is currently blocked by Google. OpenDesktop: Used for reviewing store.kde.org content in Discover This isn’t all that much. Notably absent here is KDE PIM, which could greatly benefit from integrating with the Nextcloud and Google accounts. This is something many users have asked for over time. Plus, there’s more services that are used across applications and could benefit from a systemwide online accounts system, like Mastodon or Matrix. Overall the situation with online

accounts support in KDE is unsatisfactory, and it’s not for a lack of trying. Over the last few years there have been several smaller improvements to the system. However there have been no changes to the overall architecture. At this point I am convinced that the architecture is what’s holding us back, and we need to do something about that. The current system is based on the Accounts SSO framework. It consists of several libraries and processes, split across about a dozen different repositories. This makes for a rather complex system for what is effectively reading and writing to a sqlite database and some OAuth handling. It also receives very little development activity, to the point where it was hard to get the required patches for Qt6 support in. Using an external accounts system as based for KAccounts only makes it harder to iterate on our system, while providing no meaningful interoperability with other parties. The system also isn’t designed for a sandboxed world. Apps have direct access to the accounts database and keychain, so there is no ability to restrict which apps can use which accounts. While per-application access control wasn’t really feasible for traditional Linux packaging, with sandboxed formats like Flatpak we can and want to restrict apps to only be able to access select accounts. Having pondered the problem for a while I came to the conclusion that we need a fresh start for our online accounts story, a new system that fullfils the following goals: It’s actually used by relevant KDE and third-party software Easy to hack on and extend Easy to be integrated into consumer software, with minimal dependencies Can be extended with third-party providers Account data is stored reasonably securely, with per-application access control (for sandboxed applications at least) Based on these goals I have developed a prototype for how such a system could look like. At its core there is a daemon process that manages the accounts, and exposes them via a DBus interface. Applications uses this DBus interface to list availble accounts as well as their parameters and credentials. Only accounts the app has been granted access to are visible that way. Application authentication works in a way that’s inspired by how xdg-desktop-portal works. An application can trigger a request for accessing a new account. The dameon will then handle the whole login flow and, if successful, will return a handle to the new account. Alternatively the user can log into a given service in the systemsettings module and give access to relevant apps though that. Currently the following services/apps are supported: Nextcloud: Used by Purpose and KDE PIM Mastodon: Used by Tokodon Google: Used by KDE PIM and Purpose You can find the code at https://invent.kde.org/nicolasfella/konlineaccounts. It is still very much a prototype, which is by no means ready for production, but it shows the basic concept. If you have input on this please get in touch, for example by filing an issue.

- OSM Hack Weekends October and November 2025 (2025/11/15 07:15)
  Last weekend I once again attended the bi-annual OSM Hack Weekend in Karlsruhe hosted by Geofabrik. I’ve also been at the OSM Hack Weekend in Berlin hosted at Wikimedia Deutschland a couple of weeks ago and haven’t written about that yet, so this is the combined report for both events. Transitous Transitous, our community-run public transport routing service, has been the topic of several discussions: Ways to deal with GTFS static feed rotation happening out of sync with corresponding realtime feeds. This results in time periods where available realtime information cannot be matched to base schedule data and thus gets needlessly discarded. How to best configure GBFS provider groups as supported by MOTIS v2.7. Integrating GBFS data from Citybikes, which would substantially increase the amount of available rental vehicle data. Assessing what it would take to add Transitous as an additional routing option to the OpenStreetMap website. Investigating how far along the OSM Road Closures GSoC project is, as that kind of data is obviously very interesting to integrate eventually. Exploring whether FOSSGIS e.V. would be a suitable organisational home for Transitous. Available rental vehicles shown on Transitous' map view. KPublicTransport KPublicTransport, KDE’s client library for accessing different journey planning services used by Itinerary and KTrip, got a few improvements to catch up with Transitous and MOTIS v2.6 and v2.7 changes: Access to agency/operator URLs. Querying available station-bound and free-floating rental vehicles from MOTIS. Support for direct booking URLs for station-based rental vehicles. Station-bound rental vehicles other than bikes are

now also displayed with the correct vehicle icon on the map. Itinerary's station map showing a car rental station and two free-floating rental bikes. Indoor mapping Indoor mapping was of course also on the agenda: I got to try Tobias's JOSM patches improving level filtering. Especially the option to filter on elements without a level tag is helpful for fixing level tagging in existing buildings for me. We talked about ongoing tagging discussions from TU Munich's BIM import, in preparation for the next quarterly OSM Indoor Meetup. We discussed whether we should have another in-person Indoor tagging workshop following the one from 2022, in order to have some time to work on finalizing tagging proposals and updating the current indoor tagging documentation. Emergency and weather alerts At the CAP Implementation Workshop two weeks ago a WFS/OGC feature layer for CAP alerts was mentioned, and presented as something so far only offered by a commercial entity. With my almost non-existent GIS knowledge this looked like something that shouldn't be too hard to provide by our CAP alert aggregation service as well. And thanks to the input from the right people I got a basic prototype set up in less than an hour. All the magic is provided by pg_featureserv, which can expose a PostGIS database (which we already have) in a way it can be consumed by e.g. QGIS. QGIS with a CAP alert message layer. One important difference here is that unlike its proprietary counter-part this doesn't expose many CAP fields yet, as we hold only the bare minimum as dedicated database columns right now. However, should anyone actually need this, adding more columns isn't a big deal. Event planning We also looked at upcoming events in 2026 and how we could have Transitous specifically and the Open Transport community more generally represented there: 39C3, 27-30 Dec in Hamburg Germany. We'll try to have some kind of Transitous meetup there. FOSDEM, 31 Jan-1 Feb in Brussels, Belgium. The CfP for the Railways & Open Transport track is still open and we have poked a few people to submit talks. FOSSGIS-Konferenz, 25-28 Mar in Göttingen, Germany. The CfP is already closed, a few proposals have been submitted. Still further out is next year's State of the Map which will be end of August in Paris, France. That's obviously something where Transitous should be present as well, and where we might have the option of a travel-optimized adjacent Transitous sprint along the way. Ideas for a 2026 edition of the Open Transport Community Conference are also floating around already, volunteers to drive this still very much needed though. You can help! Hack weekends how this is called in the OSM community or sprints as this is known in the KDE community are immensely valuable and productive. There's a great deal of knowledge transfer happening, and they are a big motivational boost. However, physical meetings incur costs, and that's where your donations help! KDE e.V. and local OSM chapters like the FOSSGIS e.V. support these activities.