

# The **PCLinuxOS** magazine

Volume 150

July, 2019



*De-Googling Yourself – Part 3*

*Google To Block Modern Ad Blocking  
On Chrome*

*Repo Review: RedNotebook*

*Short Topix:*

*S. Korea Gov't Switching To Linux*

*PCLinuxOS Friends & Family  
Spotlight: Aragorn*

*PCLinuxOS Recipe Corner:  
Queso Taco Pockets*

*GIMP Tutorial: Pressed Text*

*ms\_meme's Nook: Linux Boogie*

*Casual Python – Part 6*

*Ruby Programming Language: More  
Ruby Concepts for Data Handling*

*And more inside ...*

# In This Issue...

- 3 From The Chief Editor's Desk...**
- 4 Screenshot Showcase**
- 5 De-Googling Yourself, Part 3**
- 9 Screenshot Showcase**
- 10 PCLinuxOS Recipe Corner: Queso Taco Pockets**
- 11 ms\_meme's Nook: Linux Boogie**
- 12 Short Topix: S. Korea Gov't Switching To Linux**
- 15 Screenshot Showcase**
- 16 GIMP Tutorial: Pressed Text**
- 18 uBlock Origin + Privacy Badger: Say Goodbye To Ads, Tracking In Firefox**
- 20 PCLinuxOS Family Member Spotlight: Aragorn**
- 22 Screenshot Showcase**
- 23 Repo Review: RedNotebook**
- 24 Google To Block Modern Ad Blocking On Chrome**
- 26 The Ruby Programming Language: More Ruby Concepts For Data Handling**
- 30 Screenshot Showcase**
- 31 PCLinuxOS Bonus Recipe Corner: Teriyaki Chicken Meatballs**
- 32 Casual Python, Part 6**
- 42 Screenshot Showcase**
- 43 ms\_meme's Nook: Who Can Say**
- 44 PCLinuxOS Puzzled Partitions**
- 48 More Screenshot Showcase**

## The PCLinuxOS magazine

The PCLinuxOS name, logo and colors are the trademark of Texstar.

The PCLinuxOS Magazine is a monthly online publication containing PCLinuxOS-related materials. It is published primarily for members of the PCLinuxOS community. The magazine staff is comprised of volunteers from the PCLinuxOS community.

Visit us online at <http://www.pclosmag.com>

This release was made possible by the following volunteers:

**Chief Editor:** Paul Arnote (parnote)  
**Assistant Editor:** Meemaw  
**Artwork:** ms\_meme, Meemaw  
**Magazine Layout:** Paul Arnote, Meemaw, ms\_meme  
**HTML Layout:** YouCanToo

**Staff:**

ms_meme	Cg_Boy
Meemaw	YouCanToo
Gary L. Ratliff, Sr.	Pete Kelly
Daniel Meiß-Wilhelm	Smileeb
daiashi	Alessandro Ebersol

**Contributors:**  
Aragorn

The PCLinuxOS Magazine is released under the Creative Commons Attribution-NonCommercial-Share-Alike 3.0 Unported license. Some rights are reserved.  
Copyright © 2019.



# From The Chief Editor's Desk...

Woof!

The “dog days of Summer” have arrived, and with a vengeance. Around where I live in the Kansas City area, we’ve been under heat advisories, on top of the flooding that has been plaguing our area. Just when the water levels were on their way down, the U.S. Army Corps of Engineers was forced to release more water from upstream reservoirs. That sent already overflowing rivers and streams back to flood stage levels.

Our temperatures have been relatively mild up through the third week of June. But during the last week of June, things started heating up. Now, we’re having temperatures in the mid to upper 90s (°F), and low temperatures staying in the 80s (°F) for several days. You would think that low temperatures dipping into the lower 70s (°F) would offer some relief, but anyone who’s been to the U.S. Midwest knows that the humidity is atrocious during the summer. All of that humidity can push the “Real Feel” or heat index (what the temperature feels like when you add in the humidity levels) over 100°F!

Thankfully, the milder temperatures earlier in the month afforded us the chance to take the kids to the Kansas City Zoo, as well as a free air show at Whiteman Air Force Base in nearby Knob Noster,

Missouri (the home base for the majority of the U.S.’s 20 B-2 stealth bombers). At the latter, we witnessed flight demonstrations and static displays of many aircraft, including a B-29 Superfortress, P-51 Mustang, A-10 Thunderbolt, F-22 Raptor, B-52 Stratofortress, and B-2 Spirit Bomber. Ryan and Dad took in a baseball game, which somehow managed



to dodge passing thunderstorms to get the game played. Plus, every Thursday night, the local movie theater on the town square shows a family-friendly movie outdoors on the side of the building. We take the kids, along with some lawn chairs, drinks and snacks, and take up temporary residence in the theater’s parking lot for a few hours to watch the free

movie. The kids always have a great time, even if they aren’t always that into the movie that is playing.

\*\*\*\*\*

Also, during the month of June, we both got new smartphones. We went with the Google Pixel 3. Yes, there are a LOT of problems being tied so closely with Google. I acknowledge that. For us, it was a hardware decision. We use our smartphones ... A LOT ... to take pictures of the kids. So, it makes sense for us to be able to get the smartphone that has the best camera. I did ... briefly ... consider going with an iPhone, for one – and only one –

reason: Apple does an exemplary job of protecting user privacy, which is something that Google totally sucks at. But, the closed Apple/iOS ecosystem, coupled with Apple’s stranglehold on the iOS App Store, were reasons enough to quickly put the idea of switching to the iPhone to rest. Even though the latest iPhone has a really great camera, it isn’t as good as the camera in the Google Pixel 3.

\*\*\*\*\*

This month’s cover comes from our visit to the Kansas City Zoo, where they have viewing windows where you can view the penguins swimming underwater. The cover image is of some of those penguins swimming underwater, taken through one of the large viewing windows with the camera on the Google Pixel 3. Hey, a penguin is the mascot of Linux, so taking pictures of the penguins at the zoo was a no-brainer.

Until next month, I bid you peace, happiness, serenity and prosperity.

Chat with PCLinuxOS users  
from all over  
the World.

**Sign up  
TODAY!**

**PCLOS-Talk**  
<http://pclostalk.pclosusers.com>

# Disclaimer

1. All the contents of The PCLinuxOS Magazine are only for general information and/or use. Such contents do not constitute advice and should not be relied upon in making (or refraining from making) any decision. Any specific advice or replies to queries in any part of the magazine is/are the person opinion of such experts/consultants/persons and are not subscribed to by The PCLinuxOS Magazine.
2. The information in The PCLinuxOS Magazine is provided on an "AS IS" basis, and all warranties, expressed or implied of any kind, regarding any matter pertaining to any information, advice or replies are disclaimed and excluded.
3. The PCLinuxOS Magazine and its associates shall not be liable, at any time, for damages (including, but not limited to, without limitation, damages of any kind) arising in contract, tort or otherwise, from the use of or inability to use the magazine, or any of its contents, or from any action taken (or refrained from being taken) as a result of using the magazine or any such contents or for any failure of performance, error, omission, interruption, deletion, defect, delay in operation or transmission, computer virus, communications line failure, theft or destruction or unauthorized access to, alteration of, or use of information contained on the magazine.
4. No representations, warranties or guarantees whatsoever are made as to the accuracy, adequacy, reliability, completeness, suitability, or applicability of the information to a particular situation. All trademarks are the property of their respective owners.
5. Certain links on the magazine lead to resources located on servers maintained by third parties over whom The PCLinuxOS Magazine has no control or connection, business or otherwise. These sites are external to The PCLinuxOS Magazine and by visiting these, you are doing so of your own accord and assume all responsibility and liability for such action.

#### Material Submitted by Users

A majority of sections in the magazine contain materials submitted by users. The PCLinuxOS Magazine accepts no responsibility for the content, accuracy, conformity to applicable laws of such material.

#### Entire Agreement

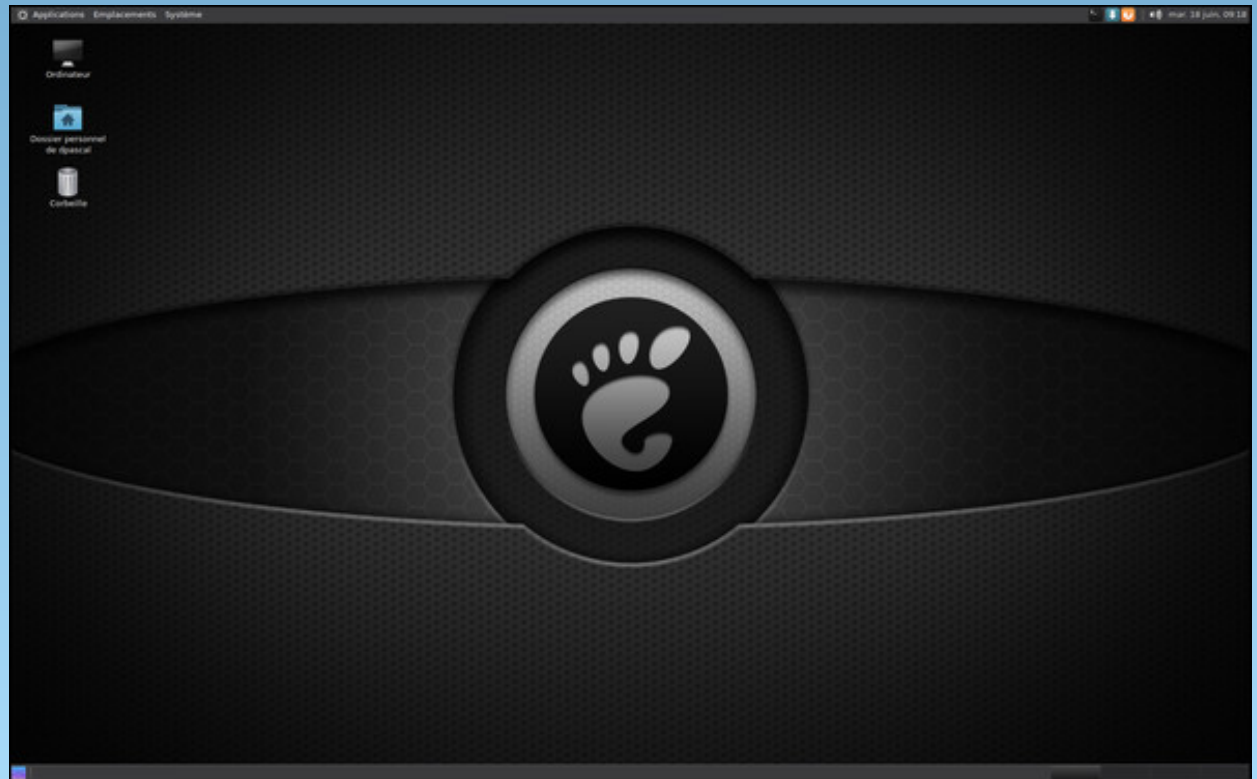
These terms constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes and replaces all prior or contemporaneous understandings or agreements, written or oral, regarding such subject matter.



**Like Us On Facebook!**  
**The PCLinuxOS Magazine**  
**PCLinuxOS Fan Club**



# Screenshot Showcase



*Posted by dpascal, on June 18, 2019, running Mate.*

# De-Googling Yourself, Part 3

by Alessandro Ebersol (Agent Smith)



## Recapping

I wrote about the genesis of Google (1st part), how Google captures and stores the information of its users (part 2), and now I will discuss how Google can harm business, YouTubers, and the ordinary citizen (me and you).

## How Google Hurts Businesses

A study, partially sponsored by [Yelp](#), and led by professors Michael Luca of Harvard Business School and Tim Wu of Columbia University, showed that Google, with its search algorithm, mainly favors services that it provides.

The research combines statistical tests with detailed economic and legal analysis to examine the ramifications of Google's practice of promoting its own specialized search services, such as restaurants or local doctors, at the expense of rivals like Yelp and TripAdvisor.

The study found that users are 45 percent more likely to click on search results generated organically by Google's own search engine than in results in which Google defends its own services as it does now.

Google restaurants

Web Images Maps Shopping More Search tools

About 1,440,000,000 results (0.53 seconds)

**Best Deals and Discounts on the Best Local Restaurants - Save ...**  
[www.restaurant.com/](#)  
Saving money on dining out is easy with Restaurant.com gift certificates for only \$10!  
Find Restaurants - Redeem Certificate - New York Restaurants

**"Commitments 1"  
Proposed: 3/2013  
Rejected: 7/2013**

Search on Site1, Site2, or Site3

**Cascal**  
[www.cascalrestaurant.com/](#)  
Zagat: **23** / 30 - 1,112 Google reviews

**XANH**  
[www.xanhrestaurant.com/](#)  
Zagat: **22** / 30 - 902 Google reviews

**Tied House Brewery & Cafe**  
[tiedhouse.com/](#)  
Score: **15** / 30 - 718 Google reviews

**Chez TJ**  
[www.chezjtj.com/](#)  
Zagat: **24** / 30 - 459 Google reviews

**Sushitomi Restaurant**  
[www.sushitomi.com/new\\_english.html](#)  
Score: **22** / 30 - 427 Google reviews

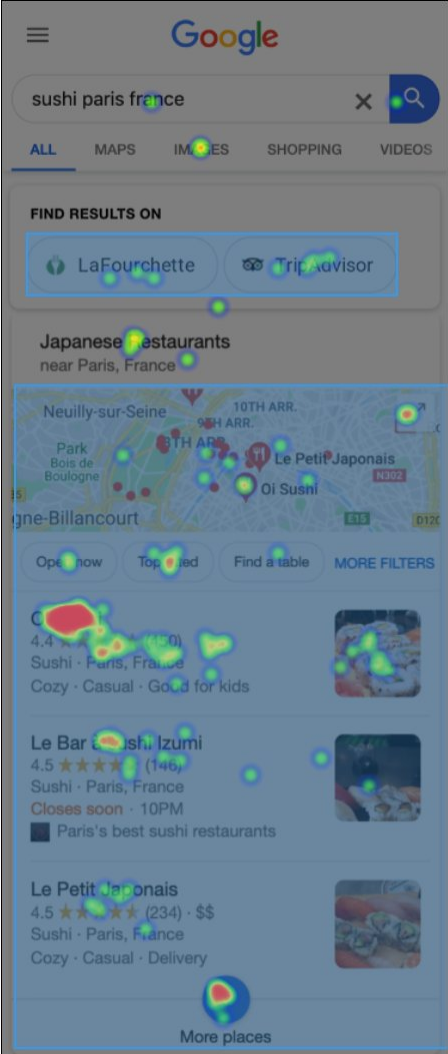
**Ristorante Don Giovanni**  
[www.dongiovannis.com/](#)  
Score: **18** / 30 - 58 Google reviews

**Le Petit Bistro**  
[www.lepetitbistro.biz/](#)  
Score: **22** / 30 - 41 Google reviews

More Google Places near Mountain View, CA »

**San Francisco » Restaurants - Yelp**  
[www.yelp.com/c/sf/restaurants](#)  
Best Restaurants in San Francisco. Mosaic; List. 1. Gary Danko. 4.5 star rating. 3114 reviews. user photo · Christina L.: He had the 5-course Tasting Menu with ...

This suggests that by leveraging the domain in the pursuit of promoting its internal content, Google is reducing social welfare - leaving consumers with lower quality results and worse matches. "The study provides empirical evidence" that Google favoring its own products in some cases harms Google users. The study was presented as evidence in the antitrust case filed by European Union regulators. The EU antitrust director Margrethe Vestager formally accused Google in April 2015, for diverting the results to favor its price comparison service, prolonging the five-year investigation. The question is whether Google uses its share of 90% of online searches in Europe to squeeze rivals in related markets in which it also competes.



And this lawsuit, which has been going on for more than five years, in March 2019, Google proposed to change the way it displays the results, in order to better favor its competitors.

The second picture on the previous page is a results screen for 2013 where the results of competitors hardly appear.

And to appease regulators of the European Union, after all this time, Google has proposed to change the format of the results to those of the picture directly on the left.

The latest proposal of rival links, which was reported by various media, still receives a slightly better but ultimately marginal percentage of clicks (about 5%), while web traffic capturing has increased by approximately 20% since the original proposal, five years ago. Seven out of ten clicks go to Google (compared to about five in ten clicks of the 2013 and 2014 proposals). These trends are consistent with other empirical evidence showing that Google is eating more and more of the web to drive traffic to itself.

How Google Hurts YouTubers

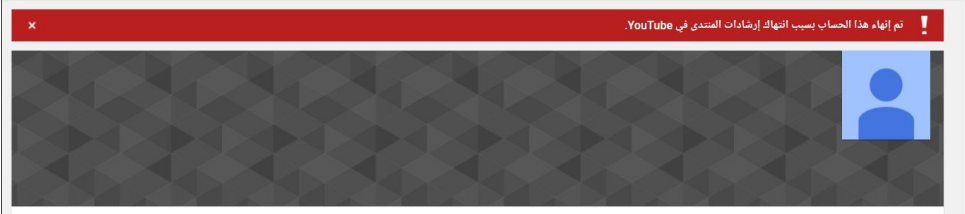
As I wrote here previously, Google, which controls YouTube, has for some time been attacking small channels in order to destroy them. Google either applies censorship, or acts in a way that does not show about channel updates, and does not put them on the first page (suggested channels), reducing their visibility.



This posture, which has been manifesting since 2012, peaked in 2017. Basically, YouTube has become another TV channel, and several channels, whose content is not attractive to sponsors, have begun to be sabotaged (some subtly, others not so much). And, all in the name of fighting extremism on YouTube ...

YouTube launched a mass censorship campaign that, in a deceptive way, called for repression of "extremist content." In fact, YouTube has been found to censor videos from respected news organizations, anti-war activists and independent journalists.

Several accounts, such as Airwars, Bellingcat, Middle East Eye and Orient News, had the videos removed or suspended as a result of past videos considered "extreme" by YouTube machine learning.



YouTube shut down Syrian Arab News Agency (SANA) channel

In addition to working with robots, YouTube is working with well-known dangerous guidance organizations that help censor YouTube videos. These organizations include the extremist/religious supremacist group ADL, the liberal, anti-Christian radical group No Hate Speech, and the Orwellian Institute for Strategic Dialogue, based in London.

And as for its copyright policy, YouTube has faced a continuing negative backlash from various individuals. The company's YouTube Rewind 2018 video is destined to be the most disliked video ever.



Several creators on the platform have spoken out against the company's copyright claim system. In recent years, large corporations have been empowered to destroy small channels by themselves. They can mark videos as infringing on their copyrights, and YouTube offers little or no support for creators, even if they're right.

To make things even worse, in July 2018, YouTube announced a program to encourage reliable sources of information and its channels on YouTube as a way to drive away conspiracy theorists and fake videos. This initiative, called the Google News Initiative, was the reason to donate US\$ 25 million in grants to news agencies that are investing in online video resources. That's a small amount for the multibillion-dollar company, but YouTube executives say it can grow over time. The funding is part of a US\$ 300 million fund to strengthen and raise quality journalism, announced by Google in March, 2018. The hope is that this funding will help news organizations build more robust video operations to compete with amateurs who would like to "trick" their audiences. That is, less space for the smaller ones and greater proliferation of the old media (Fox, CNN and the like), on a platform that should prevail for new forms of expression. We will have the same old stories, always told by the same old sources.

YouTube also sometimes agrees with the censorship requirements of foreign governments. More recently, in exchange for overturning a three-year YouTube ban in Pakistan, it has agreed to allow the Pakistani government to determine which videos can and can not be posted.

One thing to be said here, however painful, is that YouTube is not the place for free speech. It is more a TV channel that has to sell soap, and not give voice to all sectors of society (which would be wonderful). They certainly want you to think that they give everyone a voice, as evidenced by their "credo" [here](#).

### How Google harms its users (censors searches and results)

#### Autocomplete Censorship

There is a list of words and phrases deleted from the autocomplete feature in the Google search bar. The search bar instantly suggests various search options when you enter words like "democracy" or "watermelon," but freezes when you type swearing and sometimes gets frozen when people type words like "torrent" or "bisexual" or "penis". Right now, it's freezing when I type "clitoris." The autocomplete blacklist can also be used to protect or discredit political candidates.

As recently reported, at the time the autocomplete shows "Ted" (for former Republican presidential candidate Ted Cruz) when you type "lying", but does not

show "Hillary" when you type "crooked" - not even in my computer, anyway, when you type "crooked hill". (The nicknames of Clinton and Cruz are invented by Donald Trump, of course). If you add the "a", then you have "crooked hilla", you get a very strange suggestion: "Crooked Hillary Bernie". When you type "crooked" in Bing, "Crooked Hillary" appears instantly. Google's list of prohibited terms varies by region and individual, so the "clitoris" may work for you.

And recently, the site Project Veritas blew the whistle that Google would try to rig the 2020 election, to prevent the re-election of President Donald Trump. The whole story can be found [here](#). <sarcasm> And, to no surprise, the video that accompanies the text posted on YouTube was taken down. Oh, but why on earth??? </sarcasm> Luckily, it was also posted on Vimeo. [Editor's note: And has since been taken down at the Vimeo site. It is currently only hosted at the Project Veritas site.]

#### Deleted Google account

For some years, Google has consolidated several of its products - Gmail, Google Docs, YouTube, Google Wallet, and others - so you can access all of them through your Google account. If you violate Google's vague and intimidating terms of service contract in any way, you will participate in the growing list of people who are excluded from their accounts, which means that you will lose access to all of those interconnected products. As virtually no one reads this long agreement, full of legalese, however, people get shocked when they are excluded, in part because Google reserves the right to "stop providing Services for you ... at any time." And since Google, one of the largest and richest companies in the world, has no customer service department, reintegration can be difficult. (However, since all these services collect personal information about you to sell to advertisers, the loss of Google's account may be considered by some to be a blessing in disguise.)

#### Censorship in search results Google

Google's ubiquitous search engine has become one of the best- becoming the gateway to virtually all information, handling 90% of the searches in most countries. It dominates the search because its index is huge: Google indexes more than 45 billion Web pages; its closest competitor, Microsoft's Bing, indexes only 14 billion, which helps explain the poor quality of Bing's search results.

Google's search dominance is why companies large and small live in constant "fear of Google," explained Mathias Dopfner, CEO of Axel Springer, Europe's largest publishing conglomerate, in an open letter to Eric Schmidt in 2014. According to Dopfner, when Google made one of its frequent adjustments to its search algorithm, one of its subsidiaries fell dramatically in the search rankings, and subsequently lost 70% of its traffic in a few days. Even worse than the whims

of the tweaks, however, are the terrible consequences that follow when Google employees conclude that you have violated its "guidelines." You are banned for the rarely visited region of search pages beyond the first page (90% of all clicks go to links on that first page) or completely removed from the index. In 2011, Google adopted a "corrective" manual action against retailer JC Penney - punishment for allegedly using a legal SEO technique called "link building" that many companies employ to try to increase their rankings in search results of Google. Penney was downgraded 60 positions, or lower in the rankings.

So, because of changes in the algorithm, companies can dramatically lose their visibility in Google search, and for many, it could mean premature death.

**The quarantine list**

There are several lists of quarantined sites around the world, lists that place malicious or illegitimate sites and are made public. If you think your IP is blacklisted, you can check this site: <http://blacklistalert.org/>.

But in 2007, Google created the largest and best list of quarantined sites out there. Because Google is crawling the web more widely than any other, it is also in the best position to find malicious sites. In 2012, Google acknowledged that every day it adds about 9,500 new sites to its quarantine list, and displays malware notices in its responses to between 12 and 14 million search queries. It does not reveal the exact number of sites on the list, but it certainly is in the millions on a given day.

And the sites that are on the list, when they appear in search results, are automatically blocked to access by browsers:



Common malicious screen site warning that appears in Firefox and is based on the Google blacklist

But it is not only so-called "malicious" sites that are blocked. Legitimate sites (and domains) are often hampered by Google's "hidden" policies.

In 2011, Google blocked an entire subdomain, co.cc, which contained 11 million sites, justifying its action on the grounds that most of the sites in that domain appeared to be "spam." According to Matt Cutts, still the leader of Google's spam team, the company "reserves the right" to take such action when it deems it necessary. (Right? Who gave Google this right?)

**Conclusions**

*"Power tends to corrupt, and absolute power corrupts absolutely" - Lord Acton.*

So we see that power so concentrated in Google is not good, is not democratic, and definitely not fair. And, that it uses that power in an unpredictable way. But we'll still see alternatives to how to dodge the ubiquitous Google next month.







 **PCLOS-Talk**  
Instant Messaging Server

Instant Messages

Sign up TODAY! <http://pclostalk.pclosusers.com>

## Screenshot Showcase



Posted by francesco\_bat, on June 1, 2019, running Trinity.

  PATREON

**DONATE TODAY**

Help PCLinuxOS Thrive & Survive

# PCLinuxOS Recipe Corner



*from the kitchen of  
youcantoo*

## Queso Taco Pockets

### Ingredients:

8 oz extra-lean (at least 90%) ground beef  
2 tablespoons taco seasoning mix  
(from 1-oz package)  
2 tablespoons water  
3/4 cup refried beans (from 16-oz can)  
4 oz Kraft™ Velveeta™ original cheese,  
cut into 1/2-inch cubes (about 3/4 cup)  
1 box refrigerated pie crusts, softened as directed  
on box  
Diced tomatoes, shredded lettuce, shredded  
cheese, taco sauce or salsa, if desired

### Directions:

1. Heat oven to 400F. In 10-inch nonstick skillet, cook beef over medium-high heat 3 to 5 minutes, stirring frequently, until no longer pink; drain. Reduce heat to medium-low; stir in taco seasoning mix and water. Stir about 1 minute or until mixed well. Remove from heat; let stand 5 minutes. Stir in refried beans and cheese cubes.

2. On surface sprinkled lightly with flour, roll each crust into 12-inch circle. Using 4-inch round cutter, cut each crust into 7 circles for a total of 14 circles. Re-roll remaining dough scraps to cut 2 more circles for a total of 16. Discard any remaining scraps.

3. Top 8 of the circles each with about 1/4 cup beef mixture, flattening mixture slightly and leaving at least 1/4 inch around edges of circles. Moisten edges of dough with water, and top with another dough circle, stretching dough slightly to fit; press

edges with fork to seal. Pierce top of each with fork to vent.

4. Place at least 1 inch apart on ungreased large cookie sheet. Bake 21 to 24 minutes or until golden brown. Serve warm with remaining ingredients.

### Expert Tips

Top remaining refried beans with shredded cheese; heat and enjoy as a side with your taco pockets! Adding 1 to 2 teaspoons diced jalapeño chiles to your filling will give your taco pockets a nice spicy kick.



# *ms\_meme's Nook: Linux Boogie*

Way down south near Houston town  
Met Texstar walking one day  
And as he jogged along  
I heard him singing a little song  
It was a ditty he learned in the city  
PCLOS PCLOS

Boot along boot along little users  
Boot along be on your way  
Boot along boot along little users  
He led them to the forum to stay  
Singing a Linux Boogie  
His own way  
PCLOS PCLOS

**MP3**



Singing his song  
He's just too much  
He's got a Texas accent  
With a Linux touch  
This OS is what you need  
Got everything yes indeed  
Singing a Linux Boogie  
His own way  
PCLOS PCLOS

**OGG**

# Short Topix: S. Korea Gov't Switching To Linux

by Paul Arnote (parnote)

## Great White North Target Of Over Half Phishing Attacks



Ok. This probably isn't necessarily a distinction ANY country wants. But, statistically, 52% of phishing attack targets are in Canada. This is according to the first quarter 2019 RSA Quarterly Fraud Report, as reported in a TechRepublic [article](#).

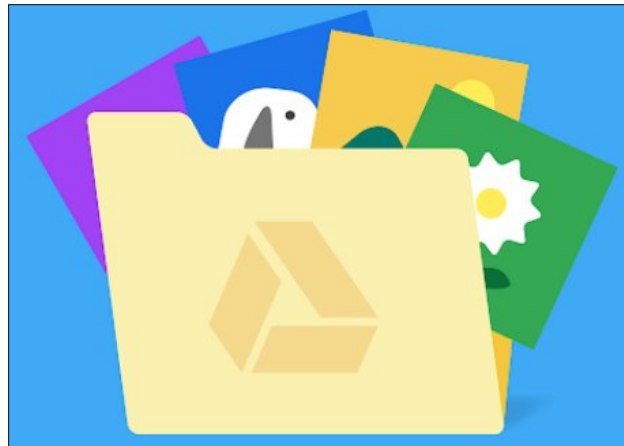
To put things into perspective, residents of Spain were the targets of 16% of phishing attacks (second place), residents of the Netherlands were the targets 10% of the time (third place), while residents of the U.S. and India tied with being the targets of phishing attacks only 6% of the time.

The list of targets of phishing attacks is completely different than the list for the country of origin for phishing attacks. The top ten countries leading the pace for the origins of phishing attacks are, in order, U.S., India, Russia, Canada, France, Germany, Poland, Malaysia, U.K., and the Netherlands. Surprisingly, China dropped out of this infamous "top 10" list in this quarter. It's also quite surprising that Nigeria isn't in this list, either, given the frequency of

the Nigerian scams users report being trapped by spam filters.

Phishing attacks exploit users by leveraging social engineering mechanisms (like making a user/target feel sympathy for a cause or individual's plight). Thus, the best defense against phishing attacks is to continually educate users so they don't become targets ... and victims.

## Changes Come To Google Drive/Photos Synchronization



Like many smartphone users, I rely on the phone's camera A LOT. It's just quick and easy to use to grab photos of the kids. In fact, it was one of the factors that weighed heavily in our decision when we purchased new phones (we ended up with the Google Pixel 3, which has the best camera of all the smartphones). Even though I have several digital cameras (including a digital SLR), the camera on my smartphone is just way more convenient for those spur-of-the-moment shots.

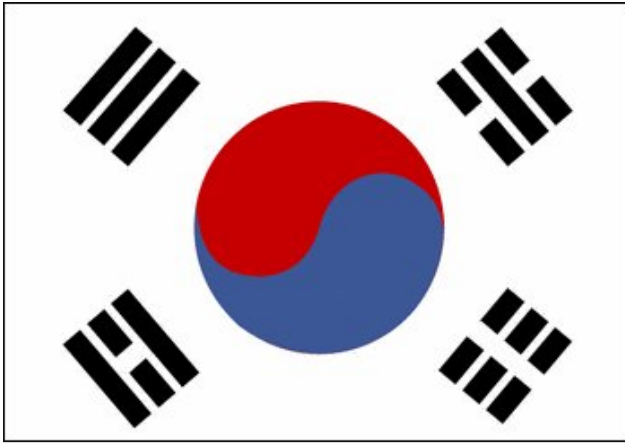
Despite all of its faults, Google has made using the Android platform quite attractive for those of us who take a lot of pictures with their smartphones: unlimited, free storage. Those photos are automatically[a][b] backed up to Google Photos when the camera and phone aren't in use. Plus, currently, all of my photos show up in Google Drive, with both services linked together without me having to do anything special to link them.

But come July, all of that is about to change, according to a Google [blog](#) entry. Google Photos will no longer automatically show up in Google Drive, and all photos in Google Drive will no longer be backed up in Google Photos. Once the change occurs, photos you delete from Google Drive will not be automatically deleted from Google Photos, and vice-versa. The change is designed to help prevent accidental deletion of items across products/services.

The change should help bring more granular control when copying photos from Google Drive into Google Photos. To facilitate this control, Google Photos will include a new feature, called "Upload from Drive." This will allow Google Drive users to upload selected photos and videos from Google Drive into Google Photos, including items from Drive's "Shared with me" items. However, the items are not synced between the two services. Items uploaded in high quality will not count towards your storage quota, but items uploaded in Original Quality will count towards your storage quota.



### South Korea Gov't To Switch To Linux



The South Korea Ministry of the Interior and Safety announced in May 2019 that it would be switching to Linux when its free Windows 7 support ends in January 2020, according to an [article](#) in The Korea Herald.

The change is expected to cost the South Korean government 780 billion won, or \$655 million U.S. dollars. The switch will include the purchase of new PCs to complement the switch.

The ministry stated that it would start off by test-running Linux on its PCs, and if no security issues arise, systems running Linux would be more widely introduced within the government. The ministry wants to test if the computers running Linux could be run on private networked devices without security risks. They also want to insure compatibility with existing websites and software, which have been built to run on Windows.

The ministry's digital service bureau chief Choi Jang-hyuk said the ministry expects cost reductions through the introduction of the open-source OS, and also hopes to avoid building reliance on a single operating system.

You might remember that back in 2013, the city of Munich, Germany completed a switch to Linux from Windows. That switch was initially approved nine years earlier, back in 2004. However, in 2017, the Munich City Council voted to return to Windows 10 in an about face that is expected to cost the city over \$50 million (U.S.). While the switch to Linux was successful, the end user discomfort with Linux was rooted in a failure to provide adequate user training. Today, Munich expects to replace all of its remaining Linux installations with Windows 10 by sometime in 2020.

### Firefox: A Brand Evolution In Progress



*(Excerpted from the Mozilla [blog](#) entry)* The “Firefox” you’ve always known as a browser is stretching to cover a family of products and services united by putting you and your privacy first. Firefox is a browser AND an encrypted service to send huge files. It’s an easy way to protect your passwords on every device AND an early warning if your email has been part of a data breach. Safe, private, eye-opening. That’s just the beginning of the new Firefox family.

Now Firefox has a new look to support its evolving product line. Today we’re introducing the Firefox parent brand — an icon representing the entire

family of products. When you see it, it’s your invitation to join Firefox and gain access to everything we have to offer. That includes the famous Firefox Browser icon for desktop and mobile, and even that icon is getting an update to be rolled out this fall.



This update is about more than logos. The Firefox design system includes everything we need to make product and web experiences today and long into the future.

- \* A new color palette that expands the range of possibilities and makes distinctive gradients possible.
- \* A new shape system derived from the geometry of the product logos that makes beautiful background patterns, spot illustrations, motion graphics and pictograms.
- \* A modern typeface for product marks with a rounded feel that echoes our icons.
- \* An emphasis on accessible color and type standards to make the brand open to everyone. Button colors signal common actions within products and web experiences.

Privacy is woven into every Firefox brand experience. With each release, our products will continue to add features that protect you and alert you to risks. Unlike Big Tech companies that claim to

offer privacy but still use you and your data, with us you know where you stand. Everything Firefox is backed by our Personal Data Promise: Take Less, Keep It Safe, No Secrets.

The brand system is built on four pillars, present in everything we make and do:

**Radical.** It's a radical act to be optimistic about the future of the internet. It's a radical act to serve others before ourselves. We disrupt the status quo because it's the right thing to do.

**Kind.** We want what's best for the internet and for the world. So we lead by example. Build better products. Start conversations, Partner, collaborate, educate and inform. Our empathy extends to everybody.

**Open.** Open-minded. Open-hearted. Open source. An open book. We make transparency and a global perspective integral to our brand, speaking many languages and striving to reflect all vantage points.

**Opinionated.** Our products prove that we are driven by strong convictions. Now we're giving voice to our point of view. While others can speak only to settings, we ground everything in our ethos.

The Firefox brand exploration began more than 18 months ago, and along the way we tapped into many talents. Michael Johnson of Johnson Banks provided early inspiration while working on the Mozilla brand identity. Jon Hicks, the designer behind the original Firefox browser logo, was full of breathtaking design and wise advice. Michael Chu of Ramotion was the driving force behind the new parent brand and system icons.

We worked across internal brand, marketing, and product teams to reach a consistent brand system for our users. Three members of our cross-org team have since moved on to new adventures: Madhava Enros, Yuliya Gorlovetsky, and Vince Joy. Along with Mozilla team members Liza Ruzer, Stephen

Horlander, Natalie Linden, and Sean Martell, they formed the core working team.

Finally, we're grateful to everyone who has commented on this blog with your passionate opinions, critiques, words of encouragement, and unique points of view.

### New Feature Allows You To Determine How Long Google Retains Your Information



We all know that Google sucks up your private data like a vacuum cleaner on steroids on overdrive. Finally, there might be some relief in sight.

Back in May, 2019, Google started rolling out a new feature that automatically deletes some of your personal data it stores on you. You can determine whether Google keeps your data for 18 months or three months. As time marches on, your data that is older than what you have set (three months or 18 months) falls off on the back side. This setting

covers *all* of Google's services, including search, maps, and YouTube.

Up until now, you had to go in and manually delete this data – or turn it off completely. By deleting it, Google doesn't always have enough information about you to make recommendations on what it thinks you might like, or where it thinks you might want to go.

Here's how to make this change to your Google account.

**Step 1:** Go to [myaccount.google.com](https://myaccount.google.com), and log in if you haven't already.

**Step 2:** Choose "Data & Personalization" in the pane on the left.

**Step 3:** Select the arrow next to "Web & App Activity."

**Step 4:** Choose "Manage Activity."

**Step 5:** Select "Choose to delete automatically."

**Step 6:** Select either 18 months or three months.

Given the option of saving 18 months or three months of data, I would opt to save only three months. The less of your private data Google has, the better your privacy. Google, trying to persuade you to retain more data, says this:

*"The activity you keep can improve your experience anywhere you use your Google Account. What you search, read and watch can work together to help you get things done faster, discover new content and pick up where you left off."*

While Google has plans to expand the controls and make them easier to find inside its apps and services, this method is the fastest and easiest way to manage your privacy. Google also said that it

plans to roll out similar controls for how it tracks your location, so be on the lookout for that, as well.

### PCLinuxOS Short Topix Roundup



**Gmail's confidential mode** became the default mode for G-Suite users at the end of June. Users of Gmail personal accounts have had the confidential mode available to them since Gmail's mid-2018 redesign. The confidential mode allows you to set an expiration date, and once reached, the message can no longer be opened. Additionally, the message content cannot be copied, downloaded, forwarded, and the sender can revoke access at any time. To bolster security, the message will only unlock after the recipient types in a verification code sent by SMS text message that's sent to their phone. Users will know that it's active when your message window has a blue header and a warning near the bottom of the compose window that the content will be protected.

**Google confirmed an Android security backdoor** in early June, according to a Forbes [article](#). This exploit, a banking trojan called Triada, was found on brand new Android phones, straight from the box. It was initially discovered in 2016 by Kaspersky Lab researchers. They called it the most advanced mobile trojan malware they had ever seen, rivaling the malware found on Windows. The phones were sold in Russia, Poland, Indonesia, China, the Czech Republic, Mexico, Kazakhstan, and Serbia. While it was initially thought that only a few models were affected, according to an [article](#) on Bleeping Computer, it was found that 42 different models of Android phones were affected.

**Google Cloud suffered an outage** in early June that caused YouTube, Shopify and Snapchat to sputter, according to an ArsTechnica [article](#). Millions across the globe lost access to their Gmail accounts. And, Google engineers had trouble repairing Google Cloud ... because the tools they needed were stored on Google Cloud. DOH! It all started with a couple of innocent misconfigurations during scheduled Google Cloud server maintenance, coupled with a software bug. The problems started around 2:45 p.m. (US Eastern Time), and things didn't return to normal until after 7 p.m. (US Eastern Time) that evening.

## Screenshot Showcase



*Posted by Mr Cranky Pants - YouCanToo, on June 1, 2019, running KDE.*

# GIMP Tutorial: Pressed Text

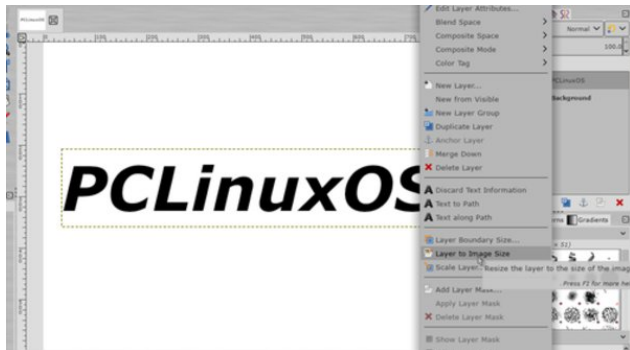
by Meemaw

I saw this tutorial several months ago and thought it was neat. It makes a text that looks sort of like it was stamped into the background. I used the same pattern as the original, so I'm going to also try it with a different pattern to see how it changes.



So, let's go through it. I made my drawing 800 x 600 and my text Verdana Bold Italic, size 120 px. I used the name of everyone's favorite distro, but use whatever text you want.

When you get your text written, change to the move tool and move it to the center of your canvas, then right-click the text layer and choose **Layer to image size**.



Now, choose only your text by right-clicking the text layer and choosing **Alpha to selection**. Now your

letters are outlined. The next step is to fill the letters with a pattern. Choose Bucket fill and change to pattern fill. Choose the pattern named "Pine?", and fill the letters. Now, add a drop shadow using **Filters > Light and Shadow > Drop Shadow**. (I used the legacy drop shadow tool.) Make your drop shadow offset 4 px.

The next step is to fill our letters again. However, we want to leave that wood grain as a border, so go to **Select > Shrink** and shrink it 3 px. Now, choose your Bucket fill again, but change back to FG fill and use a gray color. Your letters should now be gray with a wood grain border.



We're going to add another drop shadow, but it will affect the gray parts of the letters. Choose **Select > Invert** so everything else is chosen, then choose **Filters > Re-Show Drop Shadow**. Change the offset back to 8 px and click OK. This puts the shadow on the inside of the border.

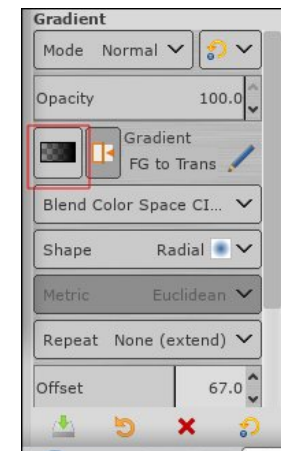


Now, go to **Select > None**, then click on the background layer and fill it with the same pattern.



Now we can merge all our layers by clicking **Image > Flatten image**.

We'll put the first of two gradients in now. Change to the **Gradients** tool, and choose a radial gradient, **FG to transparent**. Reverse the gradient as well, so the transparency is on the "left".



Now, start in the center of your canvas and draw your mouse pointer to the bottom right corner (next page, top left).

The next step is to do a bump map on this creation. Click on **Filters > Light and Shadow > Lighting Effects**. Click on the **Bump Map** tab and click the box that says **Enable Bump Map**. If our project had





multiple layers, we would be able to choose which layer we wanted to use, but we only have one layer, so just click OK.



Let's add another gradient. If you click on your gradient tool again, your settings should still be there. Start in the center again and drag past the corner.

If you've noticed, the drop shadow tool has left some extra at the sides of the project. At this point you can

crop out whatever you don't want. I chose to crop a lot of my drawing, leaving just the text. Since it is your project, you can choose how much or little to crop.



Neat effect, isn't it? I tried it again, using "Pastel stuff" for the letters and "Rain" for the background. It looks totally different. It may not appeal to everyone, but this is just another example of what can be done.



**PCLinuxOS**  
*Users Don't*

Text  
Phone  
Web Surf  
Facebook  
Tweet  
Instagram  
Video  
Take Pictures  
Email  
Chat

***While Driving.***

***Put Down Your  
Phone & Arrive Alive.***

# uBlock Origin + Privacy Badger: Say Goodbye To Ads, Tracking In Firefox

by Paul Arnote (parnote)

A problem with the web that started years ago may have hit a crescendo a few years back, but it remains a prevalent problem: being bombarded with inane, mindless ads that most of us could care less about. Now, couple the advertisement bombardment with incessant tracking of our movements around the web, and you have a new problem. Often times these issues are linked, with many ads linking back to internet trackers.

Most users don't care to look at ads that they have no interest in. Even when the ads are "targeted" to our interests, the ads are still mostly irrelevant to the vast majority of users. And certainly, no informed, self-respecting user wants their every movement tracked and scrutinized.

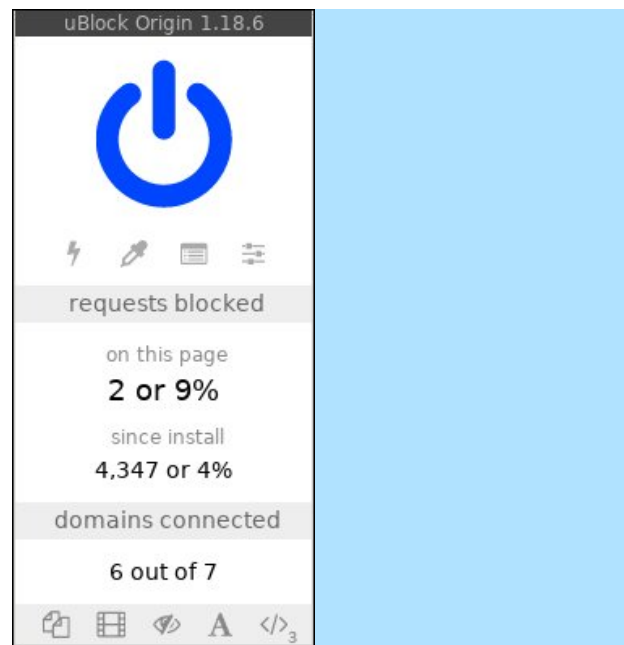
Fortunately, Firefox users can install two extensions that will stop most of this activity dead in its tracks: [uBlock Origin](#) and [Privacy Badger](#) (both links are to the Firefox extension page). While some users may think this is redundant, the two actually complement one another. But for users serious about protecting their privacy, it makes perfect sense.

None of the descriptions of the extensions is intended to be a complete how-to on how to use them. Rather, the descriptions below explain how these two extensions work together to give you (and your privacy) the utmost protection, along with a basic description for those who may not be familiar with the extensions.

## uBlock Origin



uBlock Origin isn't just an ad blocker. On its [GitHub](#) page, it is described as a "wide-spectrum blocker." It blocks ads, trackers and malware sites. uBlock Origin, which originated in 2014, is written by Raymond Hill.



Clicking on the uBlock Origin icon produces a control box like that pictured above. When a blacklisted site tries to track you, uBlock Origin blocks it. When a blacklisted site tries to load ads, uBlock Origin blocks it. When you try to visit a malware site, uBlock Origin blocks it. By itself,

uBlock Origin does an outstanding job of protecting your privacy, as well as keeping you from having to endure an endless, mindless stream of ultra-annoying ads.

The problem is that new trackers are coming out all the time. So, after a bit of time, the list of trackers will become outdated ... at least until the next uBlock Origin update. Users can use the advanced mode, where users can setup dynamic filtering rules. But, this may be a bit much to ask of many users, and involves quite a bit of work and knowledge to do properly.

## Privacy Badger



Privacy Badger, created by the Electronic Frontier Foundation (EFF), uses heuristics to block trackers. Most trackers have a specific subset of behavior attributes, and Privacy Badger learns these as it goes. So, when you first install Privacy Badger, you might not see much difference at first. But the longer you use Privacy Badger, the better it gets at blocking trackers.

Privacy Badger is **not** an ad blocker. However, some ads may end up blocked, since some are tied to trackers. This is merely a positive side effect of using Privacy Badger.

From the Privacy Badger page:

DESTINATION LINUX  
LINUX IS OUR PASSION

The longer you browse with Privacy Badger, the better it gets at protecting you. Is your Badger not blocking anything yet? That's OK! Instead of keeping lists of what to block, Privacy Badger automatically discovers trackers as you browse the Web. So start browsing a little to teach your Badger.

Invisible tracking happens in all sorts of ways; ads are just the visible tip of the iceberg. Privacy Badger sends the [Do Not Track signal](#) to trackers telling them not to track you. If they ignore your wishes, your Badger will learn to block them—whether they are advertisers or trackers of other kinds.

Privacy Badger starts blocking once it sees the same tracker on three different websites. Three strikes and it's

The screenshot shows the Privacy Badger interface with the following elements:

- Logo: A badger head icon.
- Text: "Privacy Badger version 2019.2.19".
- Text: "Privacy Badger detected 9 potential trackers on this page. These sliders let you control how Privacy Badger handles each one. You shouldn't need to adjust the sliders unless something is broken."
- Three status icons: a red 'no' symbol, a broken egg, and a green checkmark.
- Table of trackers with sliders:
 

js.agkn.com	Slider (mostly red)
dpm.demdex.net	Slider (mostly red)
techrepublic-1.disqus.com	Slider (mostly black, some yellow)
ad.doubleclick.net	Slider (mostly red)
cdn-gl.imrworldwide.com	Slider (mostly red)
secure-us.imrworldwide.com	Slider (mostly red)
- Buttons: "Disable Privacy Badger for This Site", "Did Privacy Badger break this site? Let us know!", "Donate to EFF".

## uBlock Origin + Privacy Badger: Say Goodbye To Ads, Tracking In Firefox

out! If you use other privacy tools or ad blockers, Privacy Badger will learn less, but that's OK. It might catch things your other blockers miss.

If you think Privacy Badger is breaking a page (a video doesn't play, for example), you can click the 'Disable' button to turn off Privacy Badger for that site.

Please don't forget to click on 'Did Privacy Badger break this site'. We respect your privacy so we don't send automatic reports.

Your Badger learns locally on your device. Privacy Badger will NEVER share data about your browsing unless you choose to share it.

### So Why Install Both?

You might be asking this very question. Well, the Privacy Badger page says it best:

*If you use other privacy tools or ad blockers, Privacy Badger will learn less, but that's OK. It might catch things your other blockers miss.*

Plus, remember that Privacy Badger is not an ad blocker. You will probably already have one of those installed, anyways. uBlock Origin will block the trackers it knows to block. But, should any trackers get through the first line of defense with uBlock Origin, Privacy Badger will block those as well. As a result, exceptionally few to none of the trackers will be able to follow you around the web.

The one-two punch of uBlock Origin and Privacy Badger will keep you safe from the vast majority of trackers in Firefox. While nothing is perfect, the combination of these two Firefox add-ons is probably as close as you can get.

At this point, it's impossible to predict what's in store for users of Google Chrome. First, Privacy Badger isn't even available on Chrome ... at least that I've been able to see. Second, given Google's plans to

render ad blockers much less effective (see the "Google To Block Modern Ad Blocking On Chrome" article elsewhere in this issue), Chrome users are most likely going to be victims of increased advertising, to be sure. Third, since Google is one of the most prolific trackers on the web, I can easily imagine a situation where Google would not allow a tracker blocker like Privacy Badger in the Google Play Store (from where you install Chrome extensions). It cuts into the currency that fuels Google: vacuuming up your private data to serve you ads.

The advertisement features a cartoon penguin standing on a red base. To the right, the text reads "Linux Training Courses & Classes" in a serif font, and "BeginLinux.com" in a large, bold, red, 3D-style font.

The advertisement features a man in a light blue vest and a dark bow tie, looking frustrated. The text reads: "Does your computer run slow? Are you tired of all the 'Blue Screens of Death' computer crashes? Are viruses, adware, malware & spyware slowing you down? Get your PC back to good health TODAY! Get PCLinuxOS Download your copy today! FREE!" The PCLinuxOS logo is at the bottom right.

# PCLinuxOS Family Member Spotlight: Aragorn

## As told to YouCanToo



### What is your name/username?

Call me Ishmael. No, wait, that has been done already.

Nah, just keep it to Aragorn. I've been using that name for well over a decade already. When the LOTR movies came out, many unrelated people - including my then-girlfriend and my brother's then still small children - all felt that I strongly resemble Viggo Mortensen in his portrayal of Aragorn. So eventually I adopted the name.

I don't look like that anymore now - - even though I do still have the long hair, and it's actually a lot longer now - - but then again, Viggo himself doesn't look like that anymore.

### How old are you?

I'll be 56 in about 5 weeks. If I'm still alive by then. Strange things happen, you know?

(On a more serious note, I have lost several friends recently - - a couple to cancer and one to an embolism -- and I have lost friends many years ago who were younger when they died than I am now. So you never know what's waiting around the corner. I mean: look at Tex. He's still fighting the big "C" too. (fingers crossed)

### Are you married, single?

Single. I was officially engaged to be married three times, and once unofficially. As I keep telling people, there are so many daggers sticking out of my back that people start thinking I'm a porcupine.

### How about Kids, Grandkids (names and ages)?

I don't have any children that I know of, but two of my ex-girlfriends had children, and so I've been a surrogate dad a few times.

### Do you have pets, what is your favorite?

When I was still living with my first fiancée, we had a cat, twice. Unfortunately, back at the time, I was allergic to cats. This allergy seems to have gone away in the meantime, although I am still allergic to dogs.

My brother has a dog, a border collie and a cat, but that cat is almost never to be seen when I'm around, because the dog hates her and chases her away all the time. Personally I'm more of a cat person, though.

### Are you retired, still working and if working, what do you do?

I am officially disabled. I've done all kinds of jobs in the past, from working at the assembly line at a (now defunct) General Motors plant as my first job, all the way to working for Town Hall in administration and working in a school as a programmer. And lots of stuff in between, too. I've also been an assistant-teacher in language courses and computer courses for adults.

And of course, as other male Belgians my age have all come to experience, I've been in the military for 10 months - - the exact duration of the service depended on the year, the type of service and whether you were stationed in Belgium or in

Germany. We still had compulsory military service here in those days. I served from December 1983 until late September 1984. After your service, you were then still considered part of the reserve for a number of years. During that time, I was administratively reassigned to the Marines, but I've never been summoned again, and in the meantime, the Belgian military no longer has any Marines - that's all for the para-commandos and the special forces now.

I am currently "employed" - but not as a paid job - as the administrator at two forums, and that pretty much takes up all of my time now.

### Where do you call home? What is it like? IE: weather, scenery

I live in Belgium, in the northern, Dutch-speaking part called The Flanders. I guess you could say that I live at the countryside, but that's not entirely true, because this particular town here is highly industrialized and commercialized. There used to be a shipyard here where they built large tankers, ferry boats and container carriers. The shipyard itself employed about 1600 people directly, and about 2000 more by way of its logistics chain. However, the place was mismanaged and it went bankrupt.



Flanders, Belgium

The government tried to rescue it but that didn't work out. The shipyard's grounds have in the meantime all been sold and repurposed for housing, commerce and industry.

As for the weather, well, right now it's warm and sunny - I reckon it'll be around 18 °C now, although nights are still close to freezing. It's because of climate change, of course. The climate is very unpredictable around here - it always has been in Belgium, but the last couple of years it can really be freaky at times. We've had a month of February now about 5 years ago or so where it was -20 °C during the day, and now it's T-shirt weather.

If I were to win the lottery - yeah, that'll be the day - then I'd move to the coast. It's where I used to go on vacation with my family when I was young, and I've always wanted to live there.

### **Where did you go to school and what is your education level?**

I doubt whether the name of the schools I went to would mean anything to you. Officially, my education level is high school, but I did go to college - nursing first (for two years), and then I went back to college at the age of 28 in Applied Information Technology.

I quit the nursing because I was tired of schools and the pressure that my parents were putting on me. I had to quit the IT training because of both financial and health considerations - I was rushed to hospital in an ambulance twice just before the exams. But that said, there were several horrible teachers there. They went too fast and they didn't care about properly explaining things - that's what you get when you have a military officer for a math teacher - or you couldn't even understand a single word they were saying because they were mumbling all the time.

I do still perpetually educate myself with all kinds of material available, and in multiple disciplines. I'm autistic and I have an eidetic memory, so I easily retain things I read about the subjects that interest

me. My high school training was in science - physics, chemistry, biology, math and geology - and I'm still interested in physics, mainly general relativity and quantum mechanics.

I'm also still interested in information technology, but specifically desktop workstations, minicomputers, servers and mainframes. I don't care much for the IoT. I do own two smartphones, but I use them as regular phones - texting and calling, and that's it. I'm not a smartphone zombie yet, and I don't intend to become one.

### **What kind of things you like doing? hobbies, travel, fishing, camping?**

I don't like traveling, although I do like driving, and I'm quite a petrolhead. I grew up with that. My dad was a truck driver and he would work as a mechanic in a local garage on Saturdays and when he had days off from his main job. Most of the men in my family and in my parents' circle of friends were all well-versed on automotive technology.

Another thing that I've had in me since birth - it's probably genetic - is that I'm a musician. We had musicians in both the paternal and maternal branches of my family, and my dad was a trumpet player when he was young. I've been playing all kinds of musical instruments since I was a toddler, but my true calling came when I was a teenager. That's when I fell in love with the electric guitar, although my parents weren't quick to allow me to have one - it still took until I was 16 before I had my first guitar, and my parents have never endorsed me in that, because they wanted me to have a traditional job.

I've been in a couple of bands, but nothing professional. I have a very eclectic taste in music, and so I have an equally eclectic number of genres that I incorporate into my playing, all the way from progressive rock and hard rock over to funk and jazz, and maybe even a wee touch of heavy metal. My hay days were the 1990s, though. Because in the early 2000s, I moved to a different apartment -

where I don't have enough room - and then the internet came into my life as well, and health problems started domineering my wellbeing, and so I haven't exactly been playing much anymore in the last 15 years or so.

### **Why and when did you start using Linux?**

In December 1999. I had never been one to accept the hardware manufacturer's choice of operating system, and while my first ever PC came with DOS 5.0 and Windows 3.0 - which I've used for about six months - I wanted to have OS/2 on my computer, and so I've used OS/2 for over five years. Then I needed a new computer, and I really wanted a UNIX system, but that would have been very expensive, and it was hard to come by. My friends were (of course) all using Windows 95, but that was a DOS-based system, and after having used a real 32-bit operating system for over five years, and with the new computer having a Pentium II processor, I wasn't going to settle on anything DOS-based. So I compromised and I got Windows NT 4.0. I used that for two years.

Then, late in 1999, I read an article in a computer magazine in which they were discussing several GNU/Linux distros - SuSE, Mandrake, RedHat, Slackware, Debian, Caldera and TurboLinux. Two weeks later I was at a software shop to buy a Microsoft Encarta for my brother as a Christmas gift, and there on the shelf were several of those distros that the magazine had touched upon. I hesitated, but eventually I picked up the Mandrake box - it was the 6.0 PowerPack - and I took it with me to the cashier.

I ran Mandrake in dual-boot with NT 4.0 for about a month, and then, on the 1st of January 2000, NT 4.0 refused to boot, in spite of the service packs and the official Microsoft Y2K pack I had installed. GNU/Linux booted up fine, and so my choice to stick with that was easily made. I was already seriously impressed by GNU/Linux and the whole Free & Open Source Software philosophy anyway. I've never looked back.

I've used several distros over the years. On my own computers, it has mainly been Mandrake (before it became Mandriva), PCLinuxOS, Mageia and Gentoo. But between 2002 and 2009 or so, I ran an IRC network with a bunch of people, and we ran Mandrake and CentOS on our servers. We also had one machine with Debian, but that one was located in Norway and I wasn't the admin of that box.

### What specific equipment do you currently use with PCLinuxOS?

Right now, a desktop and a laptop.

### Do you feel that your use of Linux influences the reactions you receive from your computer peers or family? If so, how?

I've had a few strange looks from a few people, but other than that, nobody really seems to care, and I've even had people who started using GNU/Linux themselves in the meantime then ask me for assistance. Of course, when you talk about computer matters, then most people still assume that you'd be running Microsoft Windows, although I also know a few people who use a Macintosh.

The thing is that nowadays, most of people's activity is centered around the Internet, and applications like Firefox, Chrome/Chromium, Opera, Thunderbird, et al, are cross-platform. Whether you visit a website by way of Firefox on GNU/Linux or Firefox on Windows, it doesn't make a whole lot of difference. And thanks to LibreOffice et al, we can easily view and even modify and save MS-Office documents.

### What would you like to see happen within PCLinuxOS that would make it a better place. What are your feelings?

Well, there are still a few minor bugs that I would like to see fixed - a few security-related things - but I cannot commit myself to reporting on all of those at the moment because the computer I'm typing this from - and which is the only workstation I can use right now - is dying. The hardware is incredibly unstable. It crashes multiple times per day. It was a refurbished machine - because that was all I could

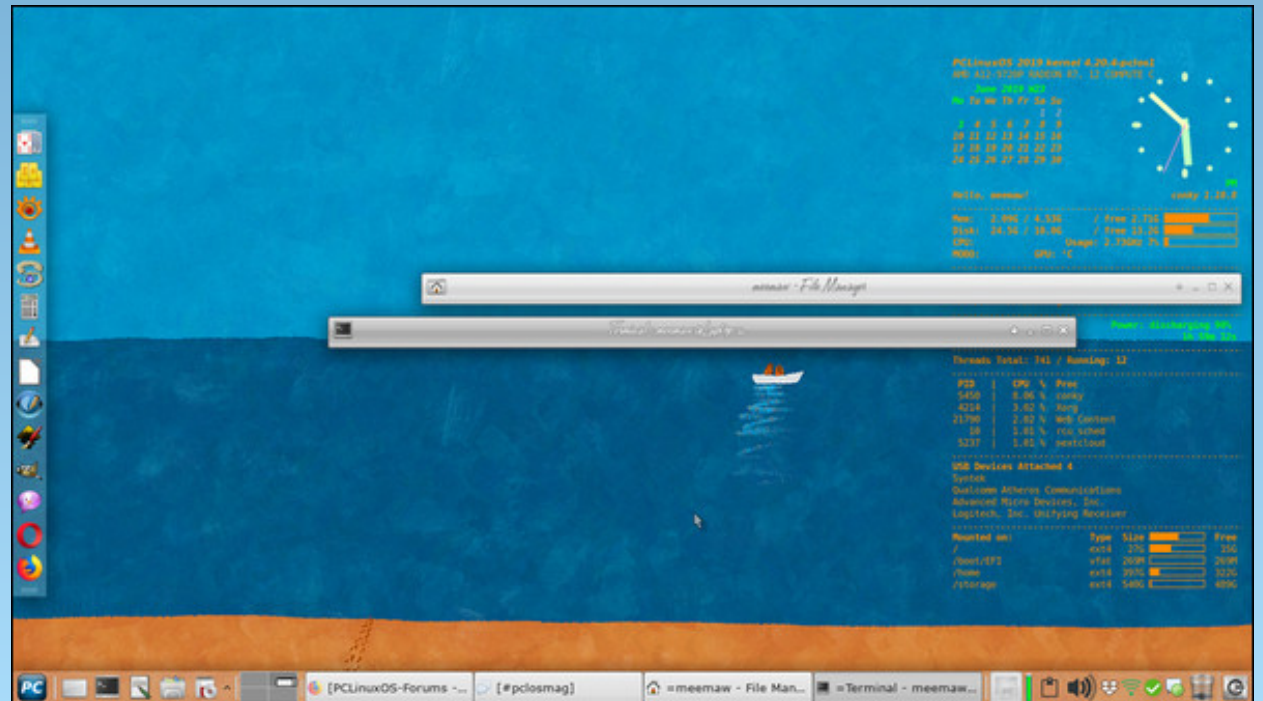
afford when my other computer broke down. (Disabilities fees are not exactly intended to make you rich, you know?)

*PCLinuxOS Family Member Spotlight is an exclusive, monthly column by YouCanToo, featuring PCLinuxOS forum member. This column will allow "the rest of us" to*

*get to know our forum family members better, and will give those featured an opportunity to share their PCLinuxOS story with the rest of the world.*

*If you would like to be featured in PCLinuxOS Family Member Spotlight, please send a private message to youcantoo, parnote or Meemaw in the PCLinuxOS forum expressing your interest.*

## Screenshot Showcase



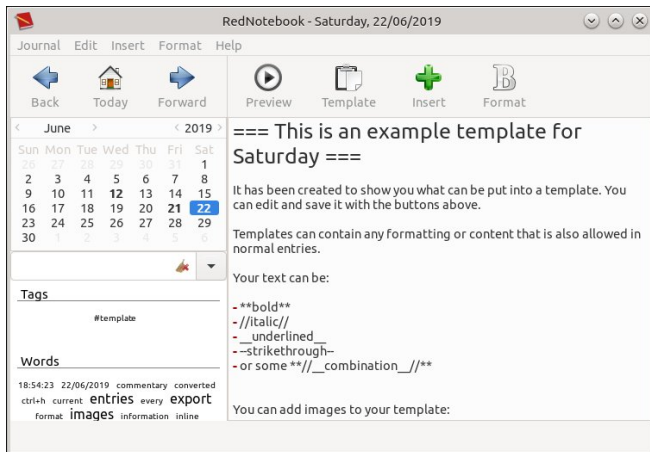
Posted by Meemaw, on June 3, 2019, running Xfce.

# Repo Review: RedNotebook

by CgBoy

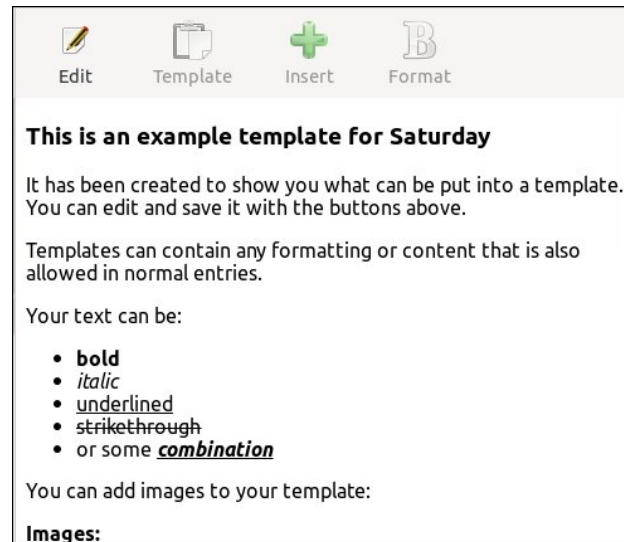
This month I'll be taking a quick look at **RedNotebook**, a simple journal keeping application. Some of its features include customizable templates, automatic saving, tags, formatting with easy to use markup, and more.

RedNotebook has a fairly simple interface. To the left of the main text entry field is a small calendar. Days with journal entries will appear in bold. Below that is a search and the word cloud, which is where journal entry tags and the most commonly used words will appear. If you want to remove a word from the cloud, just right click on it and click Hide. To tag a journal entry, simply add a hashtag to the text.



The markup is reasonably simple, and offers quite a few options for formatting. RedNotebook comes with a variety of templates for you to use, plus you can also make your own custom templates. Clicking on the Insert button allows you to easily add images, files, links, bullet lists, titles, and lines into the journal entry without actually needing to write any markup.

Clicking on Preview will then display a properly formatted version of the text, as shown below.



RedNotebook does have a spell checker, but for some reason I couldn't seem to enable it. You can view some journal statistics such as word count, the number of edited days, etc. Journals can be exported to plain text, HTML, and LaTeX formats. You can choose which specific entries are to be exported. The whole journal can also be backed up into a zip file.

## Summary

I found RedNotebook to work quite well during my testing. Although I usually don't actually keep a journal myself, I know many other people do. If all you're looking for is just a fast and simple application for writing down your daily thoughts, RedNotebook is definitely a good choice.



Support PCLinuxOS! Get Your Official  
**PCLinuxOS**  
Merchandise Today!

# Google To Block Modern Ad Blocking On Chrome

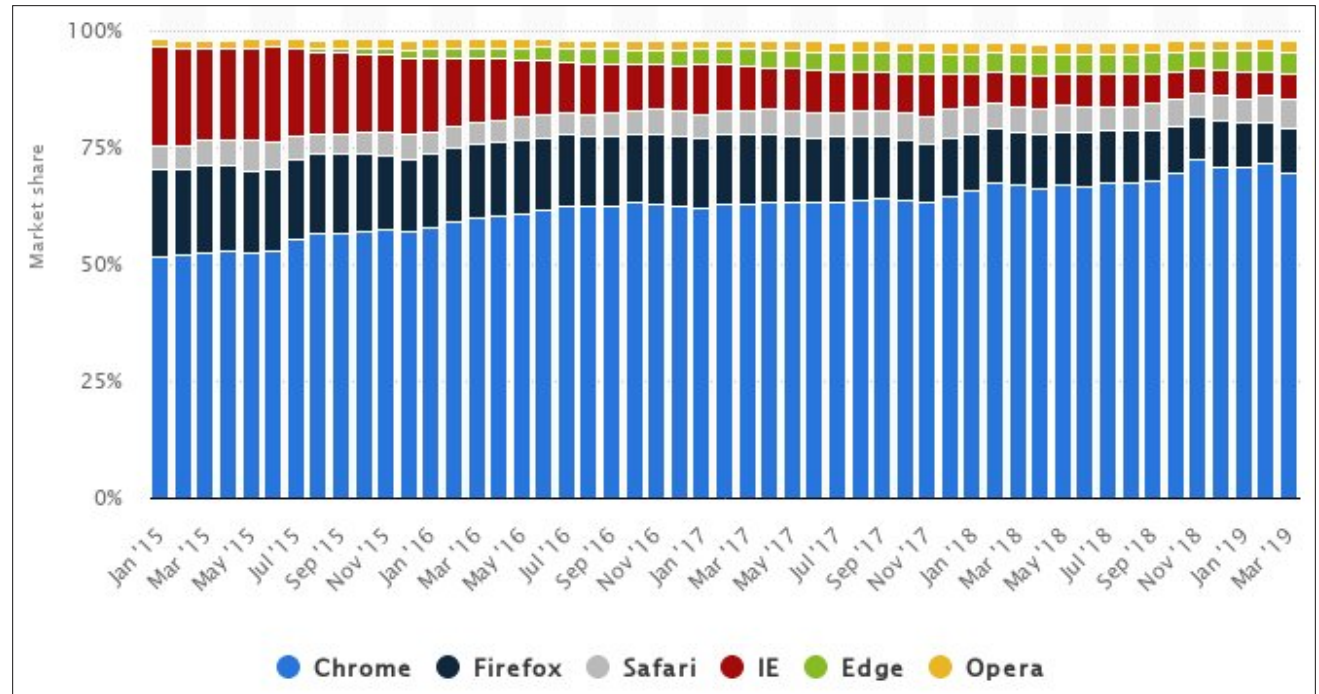
by Paul Arnote (parnote)

You just knew, in your heart of hearts, that it HAD to happen. After all, Google's primary source of revenue is selling advertising. And putting out a browser that makes it relatively easy for users to block ads is just completely counterproductive to a business model that is centered around selling ads. If the ads aren't seen (or cannot be seen), there is no money to be made.



Forget, for a moment, that almost NO ONE clicks through on those ads, which makes it puzzling why advertisers would want to waste their money in such a manner. I know that in all the years I've been on the web, I have clicked through on an ad only a handful of times. Plus, in the past ten-plus years, I have NEVER clicked through on an ad. And, they never influence my buying decisions in a positive way. More than anything, they irritate me enough to vow to never buy that product – ever.

Forget, for a moment, that Google Chrome is the most used web browser in the world (consistently between 65% and 70% market share over the past year), by far. Firefox is a distant second place, with around 10% market share. Google's browser market share is astronomical, compared to all the other web browsers. Now that they are on the top of the heap, so to speak, they want to change the rules. I suspect this move will put a serious dent in those numbers, as users flee the advertising onslaught that this



Source: [Statistica.com](https://www.statista.com)

change will likely bring. A meteoric rise in popularity is usually followed by a meteoric fall from grace.

So, let's look at this proposed change. Currently, modern day ad blockers such as uBlock Origin and AdGuard use Chrome's webRequest API to block ads, often before they can even be downloaded. But, according to an [article](#) at 9TO5 Google, the webRequest API will be deprecated, and replaced with the declarativeNetRequest API. The latter is more akin to the method used by Adblock Plus to block ads. Called **Manifest V3**, you can read all of the proposed changes [here](#) (Google Docs), in a document that spells out all of the proposed changes.

Herein lies the limitation. The declarativeNetRequest API is limited to 30,000 rules. So, ads that don't fit within the "rules" are allowed to come through.

I've used Adblock Plus in the past, and it does an ok job at blocking ads. Just ok, but not great. Since I've switched to using uBlock Origin, I've noticed a LOT fewer ads slipping through, which makes for a much, much nicer, more enjoyable time browsing.

Raymond Hill, the creator of uBlock Origin, came out with a strong [statement](#) against the proposed change in January, 2019.

*In the design document, it is said that the webRequest API will no longer allow to be used in blocking mode:*



## Google To Block Modern Ad Blocking On Chrome

- > In Manifest V3, we will strive to limit the blocking
- > version of webRequest, potentially removing blocking
- > options from most events (making them observational
- > only). Content blockers should instead use
- > declarativeNetRequest (see below). It is unlikely
- > this will account for 100% of use cases (e.g.,
- > onAuthRequired), so we will likely need to retain
- > webRequest functionality in some form.

From the description of the declarativeNetRequest API, I understand that its purpose is to merely enforce Adblock Plus ("ABP")-compatible filtering capabilities. It shares the same basic filtering syntax: double-pipe to anchor to hostname, single pipe to anchor to start or end of URL, caret as a special placeholder, and so on. The described matching algorithm is exactly that of a ABP-like filtering engine.

If this (quite limited) declarativeNetRequest API ends up being the only way content blockers can accomplish their duty, this essentially means that two content blockers I have maintained for years, uBlock Origin ("uBO") and uMatrix, can no longer exist.

Beside causing uBO and uMatrix to no longer be able to exist, it's really concerning that the proposed declarativeNetRequest API will make it impossible to come up with new and novel filtering engine designs, as the declarativeNetRequest API is no more than the implementation of one specific filtering engine, and a rather limited one (the 30,000 limit is not sufficient to enforce the famous EasyList alone).

Key portions of uBlock Origin and all of uMatrix use a different matching algorithm than that of the declarativeNetRequest API. Block/allow rules are enforced according to their \*specificity\*, whereas block/allow rules can override each other with no limit. This cannot be translated into a declarativeNetRequest API (assuming a 30,000 entries limit would not be a crippling limitation in itself).

There are other features (which I understand are appreciated by many users) which can't be implemented with the declarativeNetRequest API, for example, the



blocking of media elements which are larger than a set size, the disabling of JavaScript execution through the injection of CSP directives, the removal of outgoing Cookie headers, etc. -- and all of these can be set to override a less specific setting, i.e. one could choose to globally block large media elements, but allow them on a few specific sites, and still be able to override these rules with ever more specific rules.

Extensions act on behalf of users, they add capabilities to a \*user agent\*, and deprecating the blocking ability of the webRequest API will essentially decrease the level of user agency in Chromium, to the benefit of web sites which obviously would be happy to have the last word in what resources their pages can fetch/execute/render.

With such a limited declarativeNetRequest API and the deprecation of blocking ability of the webRequest API, I am skeptical "user agent" will still be a proper category to classify Chromium.

Google has indicated that the proposed changes are not yet set in stone, and that they are willing to work with developers to insure that their extensions continue to work under the proposed changes. As reported on 9TO5 Google, a Google spokesperson stated:

*These changes are in the design process, as mentioned in the document and the Chromium bug. We want to make sure all fundamental use cases are still possible with these changes and are working with extension developers to make sure their extensions continue to work.*

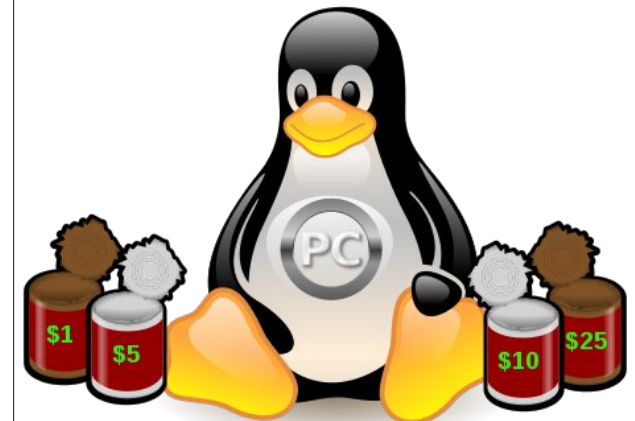
The time may be ripe for Firefox and other browsers to dethrone Google Chrome as the most popular and most used web browser. It may even be easier than anticipated, with Google helping the process along by shooting themselves in their own foot. The browser wars may have just entered a new phase, and the end users will determine the winner.

## Donate To PCLinuxOS

**Community Supported.  
No Billionaires/Millionaires.  
No Corporate Backing Or Funding.**

Click [here](#) to make a one-time donation through Google Checkout.

Or, click one of the amounts down below to make a monthly, recurring donation.



# The Ruby Programming Language: More Ruby Concepts for Data Handling

by phorneker

## Recap on Variables

Here is what we have learned so far on variables.

- Global variables, i.e. variables declared accessible to the entire Ruby program are prefixed with a `$`.
- Variables declared within a **class** declaration are prefixed with a `@` and are accessible to **all instances of that class, as well as descendants of that class**.
- Instance variables are prefixed with a `@@` and are accessible **only to that declared instance of the class and descendants of that instance**.
- Local variables are **not** prefixed with anything and are accessible **only to code contained within the block the variable declaration resides, as well as blocks of code within the block of code for which the variable has been declared**.
- **It is a good programming practice to initialize all variables declared within the Ruby program, though Ruby does not require that variables be initialized. *Uninitialized variables within a Ruby program may produce unpredictable results when the program is run, and could cause the program to crash.***

For naming local variables, it is conventional to begin the name with a lowercase letter or an underscore (`_`). Global, class and instance naming conventions are the same, with the difference being what is prefixed before the name, i.e. a `$`, `@`, or `@@` prepended to the variable name.

## Constants are variables, too.

Conventional wisdom tells us that constants never change in value. Ruby treats constants and variables the same as many functions within Ruby process constants and variables the same.

Normally, constants are declared as **global** in terms of accessibility throughout the Ruby program as their values do not change. You can declare constants within a class or module (the latter to be discussed later), **but you cannot declare constants within a method**.

By convention, constants are named in **all capital letters** (not unlike many other programming languages), and begin with a capital letter or an underscore (`_`). Unlike most other variable types, **constants must be declared with an initial (and only) value**. (Otherwise they would not be constants, right? Not to mention that Ruby would produce an error message if you accessed a constant that has not been initialized.)

Some examples of constants are:

```
PI = 3.141592653589
OPERATING_SYSTEM = "PCLinuxOS"
```

**Sorry, this name is taken.**

As with any programming language, there are some keywords that cannot be used for variable names since they have special meaning from within programs. Ruby is no exception. These keywords are better known as **reserved words**.

<code>true</code>	Represents <b>true</b> in a conditional statement, such as if/then, or in control statement such as <b>for</b> , <b>while</b> , or <b>unless</b>
<code>false</code>	Represents <b>false</b> in a conditional statement, such as if/then, or in control statement such as <b>for</b> , <b>while</b> , or <b>unless</b>
<code>nil</code>	Technically, this keyword represents <b>undefined</b> , but is treated the same as <b>false</b> in a conditional statement. <b>This keyword is the returned value for functions or statements that do not return a value.</b>
<code>__FILE__</code>	The filename for the current source file.
<code>__LINE__</code>	The line number being executed by the Ruby interpreter.

## Literal Representation of Numbers and Strings

Integers in Ruby are managed with two built in classes, namely **Fixnum** and **Bignum**. The difference between these classes is a single range of integers.

# The Ruby Programming Language: More Ruby Concepts for Data Handling

**Fixnum** is a class that manages integers with a range of **-1073741824** to **+1073741823**, which is

**-(2<sup>30</sup>)** to **(2<sup>30</sup>)-1**.

For most applications written in Ruby, **Fixnum** is the class that manages integers. Integers outside this range are managed by **Bignum**, and its range is dependent on how large an integer the processor can handle. As PCLinuxOS runs on 64-bit machines, integers managed by **Bignum** range from

**-(2<sup>64</sup>)** to **-(2<sup>30</sup>)+1** and from **(2<sup>30</sup>)** to **(2<sup>64</sup>)-1**

which should be all integers outside the range handled by **Fixnum**.

Obviously, **Bignum** and **Fixnum** support integers represented as decimal numbers. Ruby also supports numbers represented as:

- Octal (base 8), represented as a four digit number, e.g. **0777** (the leading zero here tells Ruby this is an octal integer as opposed to a decimal integer)
- Hexidecimal (base 16), e.g. **0xff** (or 255 in decimal)
- Binary, e.g. **0b1001** (or 9 in decimal)
- Character to ASCII code translations, e.g. **?@** (or 64 in decimal)

The last example is a reuse of the **?** operator. This time the question mark tells Ruby to convert the following character to its ASCII code. This use of the question mark also works with **\n**, **\t**, and **\r** (the non-printable characters), as well as **\'**, and **\\**. (We'll get to those later.)

Also, **Bignum** and **Fixnum** support the use of the underscore in place of the comma, normally used to make digits larger than three digits human readable. The comma would produce an error in Ruby if it were used when we write large numbers such as **-1,073,741,824**. However, **-1\_073\_741\_824** is an acceptable number to Ruby. Hence, Ruby would read this number as **-1073741824**.

The class **Float**, as we can guess, manages floating point numbers. Floating point numbers can be expressed as a regular floating point, or in scientific notation.

As one could presume, floating point numbers **require exactly one decimal point** somewhere in the number *except at the beginning of the number or at the end of the number*. Values that are less than one but greater than zero require a

leading zero before the decimal point (**0.**). Values that could be expressed as whole numbers (same as integers) require a trailing **.0** appended to the value.

All floating points are represented as decimals. No octal, binary, hexadecimal, or ASCII conversions are allowed for floating point numbers.

Floating point numbers **can** be expressed in **scientific notation**. Decimal points are **not required** for scientific notation, but the letter **E** is required. It does not matter whether that **E** is upper case or lower case. Examples of this are:

- **12,500 = 12.5E2 (or 12.5e2) = 12.5 times 10<sup>2</sup>**
- **1E100 = 10<sup>100</sup>** (which is the real definition of the mathematical term **google**)
- **.00125 = 1.25E-3 (or 1.25e-3) or 1.25 times 10<sup>-3</sup>**

Also, the constant **PI** explained earlier happens to be a floating point number.

Some characters need to be expressed with a different notation as they can cause errors within **puts** or **print** statements. In particular, the single quote and backslash have special meaning within strings and do not allow for substitution, e.g. using **#{}**  for variable output within double quotes.

To print a single quote, you need to use **\'** instead of just **'**. Likewise, to print a backslash, you need **two** backslashes (**\\**) instead of one (**\**).

The use of the backslash in Ruby is the same as it is for C, C++ and Java. The following table are character combinations and what they output in Ruby.

Character combination(s)	What is output or input
<b>\n</b> , <b>\0x0a</b> , <b>\012</b>	Newline
<b>\r</b> , <b>\0x0d</b> , <b>\015</b>	Return (usually combined with Newline)
<b>\f</b> , <b>\0x0c</b> , <b>\014</b>	Formfeed (or Eject Page)
<b>\b</b> , <b>\0x08</b> , <b>\010</b>	Backspace
<b>\a</b> , <b>\0x07</b> , <b>\007</b>	Bell (from old Teletype machines)
<b>\e</b> , <b>\0x1b</b> , <b>\033</b>	Escape (for ANSI Terminal and printer control languages other than PostScript)
<b>\s</b> , <b>\0x20</b> , <b>\040</b>	Space (non-breaking)
<b>\ooo</b>	ASCII character in <b>octal</b> notation
<b>\0xnn</b>	ASCII character in <b>hexadecimal</b> notation
<b>\cx</b> , <b>\C-x</b>	Control character where <b>x</b> is the character
<b>\M-x</b>	Meta character (usually Alt on PC keyboards)
<b>\M-\C-x</b>	Control-Alt-x

# The Ruby Programming Language: More Ruby Concepts for Data Handling

This table applies to **getc** and **gets** as well as Ruby programs should be able to handle input from *all* keys on the keyboard. The last three entries are best suited for the **getc** function as **getc** returns the character entered from the keyboard as soon as it is pressed.

## The Keys to Making Hash (but no Corned Beef)

We already discussed what arrays are. Ruby provides a different kind of array that does not use **indices**. **Hashes** are arrays that are not indexed, but are arrays of keyword and data pairs where the **keyword** is used to access this type of array.

Both keywords and data can be of most any data type. What is different about hashes is how they are implemented in Ruby.

```
<hash name> = <type name> = { <keyword> => <data>, <keyword> => <data>, <keyword> => <data>, ... <keyword> => <data> }
```

When data is accessed from a **hash**, each element in the “array” consists of *two variables*, namely the keyword first, then the data value. Let us look at an example:

```
color_hash = colors = { "black" => 0x00, "blue" => 0x01, "green" => 0x02, "cyan" => 0x03, "red" => 0x04, "magenta" => 0x05, "yellow" => 0x06, "white" => 0x07 }
color_hash.each do |key, value|
  print key, " has a value of ", value, "\n"
end
```

The method **each** is actually a **for** statement where there are *two values for each iteration* retrieved from the hash. The **print** statement in this example will execute seven times with a pair of values printed out for each element in the hash.

```
black has a value of 0
blue has a value of 1
green has a value of 2
cyan has a value of 3
red has a value of 4
magenta has a value of 5
yellow has a value of 6
white has a value of 7
```

When making pairs of values for hash, you could separate the keyword from the value with a comma, but I prefer to use the **=>** combination of characters to pair the keyword to the value. It makes the resulting code easier to understand when we know what keyword goes with what value.

As with arrays, we can create empty hashes like this: **<hash name> = Hash.new**

## What We Can Do With Hashes

The **==** operator we use to compare data types such as arrays, integers, strings, and floating point numbers can also be used to compare **hashes**. For **==** to return **true**, each keyword and data pair in both arrays **must be an exact match, right down to the individual bytes**. Otherwise **==** will return **false**. (*Note: This is also true for arrays and strings.*)

Arrays tend to be linear in nature, i.e. elements are appended to or deleted from either the end or the beginning of the list. Hashes, on the other hand, are **not ordered in any way** as there is no index to use to access the array. Instead, you search the list by keyword.

Within hashes, Ruby keeps track of what keywords are available and their association with data.

In languages like Pascal, arrays are explicitly defined as follows:

```
<array type> = ARRAY [<start>..<end>] OF <data type>
```

Where start and end are the starting and ending indices for the defined array. BASIC is even worse when it comes to arrays. First, arrays are defined as either numeric or string type depending on the array name being defined.

```
DIM A$(100)
```

defines an array of 100 strings with index values ranging from 1 to 100.

```
DIM X(32)
```

defines an array of 32 numbers (usually floating point depending on the BASIC dialect used) with index values ranging from 1 to 32.

In Ruby, arrays usually start with an index of zero (0), and end at **<array name>.size-1**. As we have seen, even *this* can be varied.

In terms of how hashes are implemented, Ruby implements a linked list to store hashes. The Ruby programmer does not need to worry about this detail, but if you have taken a computer science course on data structures, you will know what a **linked list** is.

As a data structure, a **linked list** is implemented as follows (in pseudo code):

# The Ruby Programming Language: More Ruby Concepts for Data Handling

```
HASH_ITEM = RECORD
  KEYWORD, DATA: <any data type>;
  PREVIOUS_ITEM: <pointer to HASH_ITEM object>;
  NEXT_ITEM: <pointer to HASH_ITEM object>;
END
```

Then for the first item in the linked list, the **keyword** and **data** are assigned values and the pointers to **PREVIOUS\_ITEM** and **NEXT\_ITEM** are set to **null** indicating the ends of the list.

A separate pointer linking to the current position in the list is also created. We shall call this **CURRENT\_ITEM**.

```
CURRENT_ITEM = NEW(HASH_ITEM);
```

As each item is added to the linked list, a new item object is created, assigned keyword and value pairs, and is then linked to the list.

```
DEFINE FUNCTION ADD_HASH(NEW_KEYWORD, NEW_DATA)
  BEGIN

  // Fill in the data structure with the keyword and data pair.

  NEW_ITEM = NEW(HASH_ITEM);
  NEW_ITEM.KEYWORD = NEW_KEYWORD;
  NEW_ITEM.DATA = NEW_DATA;

  // Link PREVIOUS_ITEM to the current item in the linked list.

  NEW_ITEM.PREVIOUS_ITEM = CURRENT_ITEM;
  NEW_ITEM.NEXT_ITEM = NULL;

  // Finally, update CURRENT_ITEM with the new item object.

  CURRENT_ITEM = NEW_ITEM;
  END
```

This pseudo code is loosely based on UCSD Pascal, but you can see how Ruby handles hashes as linked lists.

We have already seen one way to create hashes, namely with the **new** method.

```
<hash name> = Hash.new
```

This way of creating hashes produces an empty hash **with absolutely no data contained in the hash**. If anything were to be searched in this array, any search function **will return nil** for a value. We can create hashes that have a default value other than **nil**. There are two ways of doing this. With the example above,

we could either modify the statement, or add another statement setting the default.

Suppose we want the default value to be zero (0) instead of **nil**. The two ways of making this happen are as follows:

```
<hash name> = Hash.new(0)
```

which assigns the default value when the hash is created, or

```
<hash name> = Hash.new
<hash name>.default = 0
```

which does the same, except that the latter allows for statements to be executed *before* the hash default value is set.

To search a value within a hash, simply create a statement separating the name of the hash and the keyword to search with a dot. For the colors example,

```
color_hash.black
```

would return the number zero (**0**). If we were to reassign that color to, say, 255 (hexidecimal **0xff**), we would write that as:

```
color_hash.black = 0xff
```

In other words,

```
<hash name>.<keyword>
```

searches the hash for **keyword**. If **keyword** is not found, the default value will be returned (set to **nil** if no default value is set), and

```
<hash name>.<keyword> = <data value>
```

assigns **data value** and **keyword** to the hash.

But, what if you want to find out what default value is set to the hash? The following statement will get you the answer:

```
<hash name>.defaultkey
```

...and if this returns **nil**, a default value was not set for this hash.

To empty out the hash and start over, we simply write this Ruby code:

```
<hash name>.clear
```

To delete a keyword/data pair from the hash, we would write the following Ruby code:

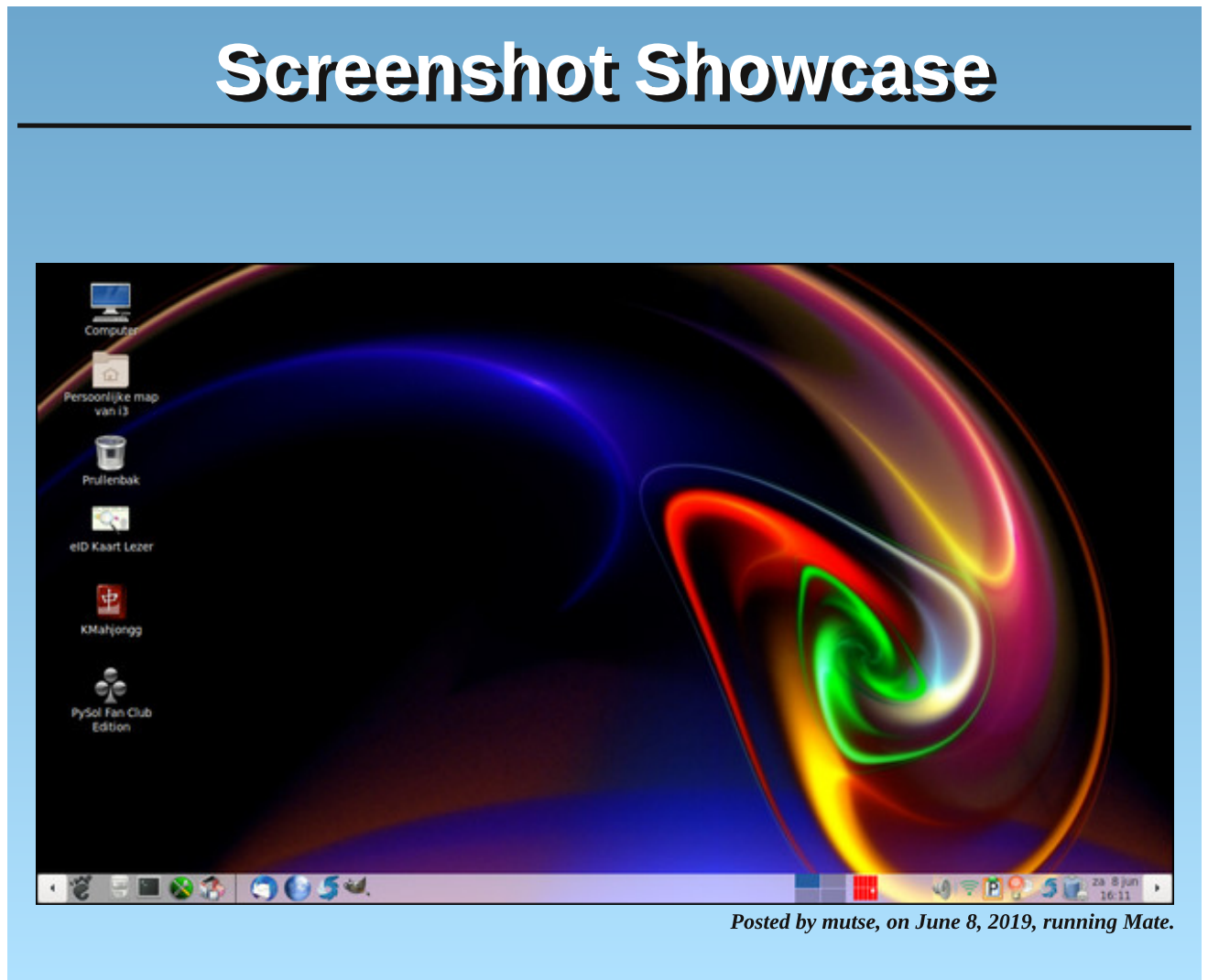
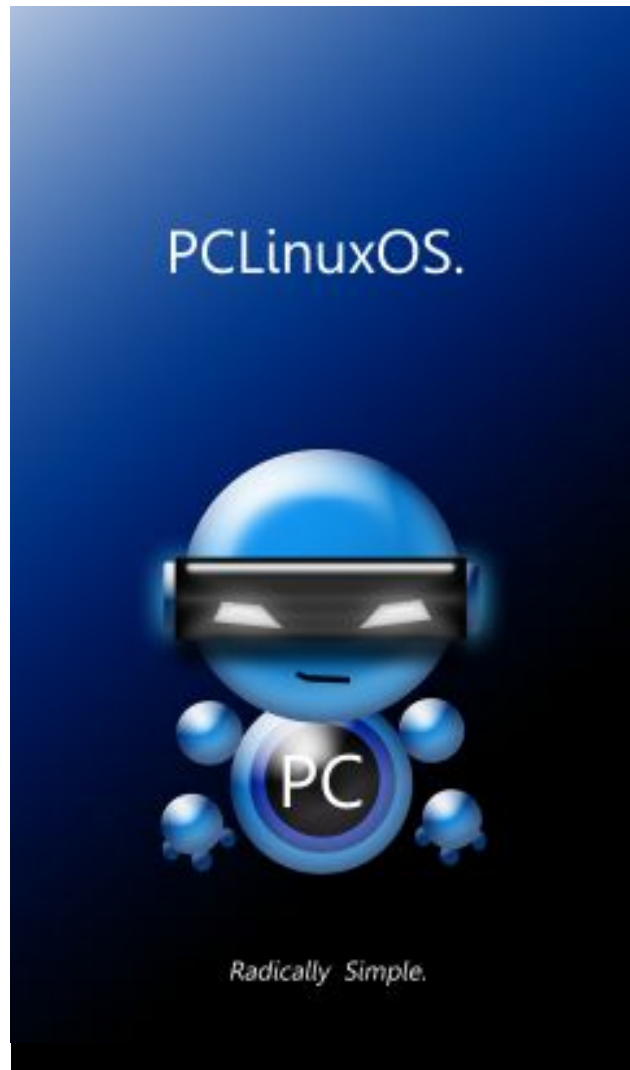
```
<hash name>.deletekey |<keyword>|
```



To be continued...

There are far more methods associated with hashes, and I have yet to cover modules. At this time, I am examining some projects developed in other programming languages. One project that comes to mind is **robotfindskitten**, a zen simulation program originally developed as a text mode application. Another project that comes to mind is the **eliza** program from the 1970s as an early form of artificial intelligence (originally written in LISP) and early interaction between humans and computers. The latter has been ported to many programming languages, but Ruby so far is not one of them.

## Screenshot Showcase



Posted by mutse, on June 8, 2019, running Mate.

# PCLinuxOS Bonus Recipe Corner



*from the kitchen of  
youcantoo*

## Teriyaki Chicken Meatballs

### Ingredients:

#### Meatballs

- 1 lb ground chicken
- 1/4 cup panko crispy bread crumbs
- 1 large egg yolk
- 2 stalks green onions, finely chopped, plus more for garnish
- 1 teaspoon sugar
- 2 teaspoons rice vinegar
- 2 teaspoons soy sauce
- 2 cloves garlic, finely chopped
- 1 teaspoon finely chopped ginger root
- Black pepper, to taste
- Crushed red pepper, to taste

#### Sauce

- 1 cup water
- 1/4 cup soy sauce
- 3 tablespoons packed brown sugar
- 2 tablespoons honey
- 2 teaspoons sesame oil
- 2 teaspoons rice vinegar
- 2 cloves garlic, finely chopped
- 1 teaspoon finely chopped ginger root
- 2 teaspoons cornstarch
- Serve With Toasted sesame seed, if desired
- Soba noodles, rice noodles, cucumber noodles, or rice

### Directions:

1. Heat oven to 400F. Line rimmed cookie sheet or casserole dish with cooking parchment paper.
2. In large bowl, mix Meatball ingredients. Scoop out 2-tablespoon-size balls of the mixture, and place on cookie sheet 1 to 2 inches apart.



3. Bake 20 to 25 minutes or until cooked through and slightly browned on top.

4. In 4-quart saucepan, beat all of the Sauce ingredients except for 1/4 cup of the water and the cornstarch, using whisk. Heat to a simmer over medium heat. In small bowl, beat together remaining 1/4 cup water and the cornstarch with a whisk. When sauce reaches a simmer, drizzle in the cornstarch mixture while beating. Beat until the sauce thickens.

5. If you plan to eat them right away, place cooked meatballs in sauce, and turn them to coat. Sprinkle with green onions and toasted sesame seed, and serve with noodles or rice. If you're wanting to freeze and enjoy later, store meatballs and sauce in separate containers in the freezer up to three months. Thaw in the refrigerator the night before, and reheat everything on the stove before serving.

### Expert Tips

These also make great party appetizers! Stick a toothpick in each one for easy snacking, and serve the sauce on the side, or drizzle it over the top.

Try enjoying these on little rolls for meatball sliders!



# Casual Python, Part 6

by Peter Kelly (critter)

## Strings and lists

In part three of this series, I listed the most important core object types that python uses. Of these, perhaps strings and lists are used more often than the others (numbers are in a world of their own). They have a lot in common in that they can be chopped, sliced and re-combined to make new strings or list objects. They also differ in that strings are immutable while lists can be changed 'in place'. Perhaps it's this difference that makes them play nice together, but then, all of the core types play nice. Strings and lists, though, are real buddies and thus deserve a little more attention.

### Strings

One thing I haven't yet mentioned is string formatting. This works a bit like a mail merge program, where a standard form letter is prepared with places to be later filled with data relative to the recipient. You know the sort of thing:

```
"Hi John, great news – we are now stocking..."
"Hi Sarah, great news – we are now stocking..." You just know they really care.
```

Well, python strings can do this, and have several ways to do it. The current, preferred way is the format **method**.

First, we define the string with braces for place holders where the inserted text will go. Then we apply the format method, passing the data to be inserted as comma separated arguments – a tuple.

```
print("Python {} are one of the core
{}".format('strings', 'objects'))
==> 'Python strings are one of the core objects'
```

Of course, we can do much more than this. The values, by default, are inserted into the braces in the order that they appear in the argument list, but we can override that by supplying numbers.

```
print("Python {0} are one of the core {1}. This makes {0} very
useful".format('strings', 'objects'))
```

```
==> 'Python strings are one of the core objects. This makes
strings very useful'
```

If you don't supply enough arguments to fill insertion points, then you will get an error.

What you supply as arguments can be any printable object.

```
old_price = 20
percentage = 5
increase = old_price * percentage / 100
new_price = old_price + increase
Print("{} + {}% = {}".format(old_price, percentage, new_price))
==> $20 + 5% = $21.0
```

We can make this more realistic by adding a formatting string inside the braces. A formatting string starts with a colon, followed by some formatting specifiers. In this example, we have:

```
{:<8.2f}
< left justify
8.2 8 spaces wide, 2 decimal places
f floating point
```

```
Print("{} + {}% = ${:8.2f}".format(old_price, percentage,
new_price))
==> $20 + 5% = $21.00
```

There are actually three spaces after the final price to make up the 8 spaces total we specified. This enables nicely formatted, consistent output from variables.

```
Percentage = 3.637
increase = old_price * percentage / 100
new_price = old_price + increase
Print("{} + {}% = ${:8.2f}rounded.".format(old_price, percentage,
new_price))
==> $20 + 3.637% = $20.73 rounded.
```

If using numbered placeholders, then the number is placed before the colon.

```
Print("{}{0} + {1}% = ${2:8.2f}rounded.".format(old_price,
percentage, new_price))
```



The general format of the method is:

```
{placeholder:[[fill]align][sign][#][0][width][,][.precision][type]
}
```

placeholder      this may be a number or name to reference the placeholder embedded in the string

:

marks the beginning of the format specification

[fill]align      align can be one of:  
                   < - left align  
                   > - right align  
                   = - padding is placed between the sign and the digits '+000123.456' (numeric only)  
                   ^ - centered  
                   fill is the padding character, '0' is the

default sign  
                   + - a sign is used for both positive and negative numbers  
                   - - a sign is used for negative numbers only (default)  
                   space - leading space on positives, '-' sign on negatives

0                   0 - preceding the width forces sign-aware zero padding

width               int - force the minimum field width

,

, - use comma thousands separator

.precision          .int - number of decimal places for a number, maximum width for a string

type                s - string, the default

integers:

b - binary  
 c - converts to unicode character before printing  
 d - decimal integer, the default if passed an integer  
 o - octal format  
 x - lower case hex format  
 X - upper case hex format  
 n - as d but forces locale appropriate separator characters

floating point:

e - exponent, default precision is 6  
 E - upper case exponent  
 f - fixed point, default precision is 6  
 F - converts 'nan' and 'inf' to upper case  
 g - general format  
 G - general format, uses upper case as appropriate  
 n - locale aware g  
 % - percentage

### Formatted String Literals

Since Python 3.6, formatted string literals are available and are prefixed with the letter 'f'.

The formatting is as described for the string format method.

```
side_a = 3
side_b = 4
hypotenuse = (side_a ** 2 + side_b ** 2) ** 0.5
```

```
f"A right angled triangle having square sides of length 3 & 4 has a hypotenuse of length: {hypotenuse:4.1f}"
```

```
A right angled triangle having square sides of length 3 & 4 has a hypotenuse of length: 5.0
```

### The old way

There is another way you may see, which is based on the c language printf routine. I would recommend not using this method unless you are particularly proficient with printf, but for completeness it looks like this:

```
pi = 3.1459265359
micron = 0.000001
minus_int = -2345
plus_int = 5678
distro = 'PCLinuxOS'
```

```
# arguments are passed in a tuple
print('Output is: %5.2f, %.3e, %6d, %6d, %5s' %(pi, micron, minus_int, plus_int, distro))
```

```
Output is: 3.15, 1.000e-06, -2345, 5678, PCLinuxOS
```

Lists

Deep copy vs. shallow copy.

The elements of a list may be of any type, even other lists. Although this is convenient, it can cause problems when making a copy of a list, as lists are mutable.

A copy can be made of a list by using the empty slice operator [:]. However, if the list contains other lists as elements, then a reference is made to the inner list in the new copy. This is known as a 'shallow copy.' Changing this element in either list results in a change in both lists. The id builtin returns the memory address of the object.

```
a = [1, [2, 3, 4], 5, 6]
b = a

id(a)          ==> 140475638438512
id(b)          ==> 140475638438512
# both variables bound to the same object

b = a[:]

# shallow copy

id(b)          ==> 140475651858304
# b now references a new object

id(a[1])       ==> 140475651857296
id(b[1])       ==> 140475651857296
# but the inner list is the original element

b[1][1] = 42

# changing an element of the inner list changes both inner lists
a          ==> [1, [2, 42, 4], 5, 6]

b[0] = 19
b          ==> [19, [2, 42, 4], 5, 6]
a          ==> [1, [2, 42, 4], 5, 6]
# the other elements are not references
```

**To work around this we can import and use the deepcopy function of the copy module.**

```
import copy
```

```
a = [1, [2, 3, 4], 5, 6]
c = copy.deepcopy(a)
# deepcopy a to c
c          ==> [1, [2, 3, 4], 5, 6]

c[1][1] = 42
# alter the inner list
c          ==> [1, [2, 42, 4], 5, 6]
a          ==> [1, [2, 3, 4], 5, 6]
# the original is unchanged
```

Unpacking

Sequence types, such as lists, may be 'unpacked' at their point of use.

```
a_list = ['cat', 'dog', 'rabbit']
c, d, r = a_list
print(c, r, d)          ==> cat rabbit dog
```

The number of object reference names on the left hand side must match the number of elements in the list, or a 'ValueError' will be raised. When the number of elements is unknown or variable, a different syntax can be used. Python supports the use of 'starred expressions'.

One, and only one, of the object reference names may be prefixed with an asterisk. The list is then unpacked, placing one element into each of the unstarred names with the starred name receiving all of the other values as a list. There must be sufficient elements in the list to supply at least one element to each of the unstarred expressions. If no items remain, then the starred expression will return an empty list [].

```
a_list = ['cat', 'dog', 'rabbit']
print(*a_list)          ==> cat dog rabbit
# unpack all elements

b_list = range(1, 4, 2)  # provides 1, 3
a, *b, c = b_list
print('a =', a, '\nb =', b, '\nc =', c)
==> a = 1
      b = []
      c = 3
```

```
b_list = range(1, 3, 2)  # provides only 1
a, *b, c = b_list
Traceback (most recent call last):
```

File "<stdin>", line 1, in <module>

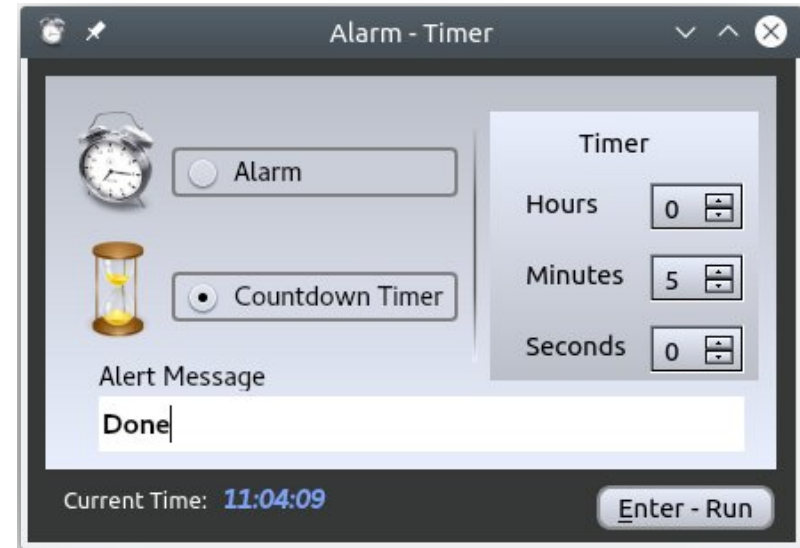
```
ValueError: not enough values to unpack (expected at least 2, got 1)
```

```
b_list = range(2, 15, 3)      # provides 2,5,8,11,14
a, *b, c = b_list
print('a =', a, '\nb =', b, '\nc =', c)
    ==> a = 2
        b = [5, 8, 11]
        c = 14
```

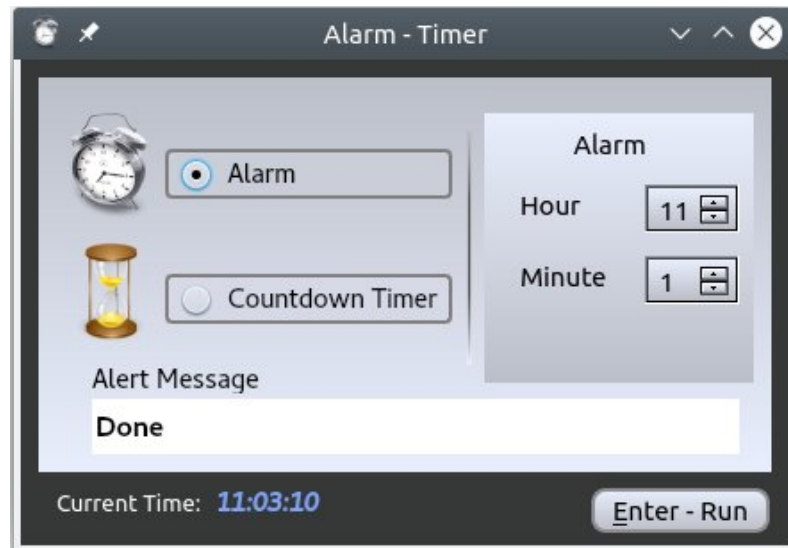
### Alarm - timer

There are lots of timer and alert applications available for the Linux desktop, some better than others, but what can be better than one you build yourself? Your own design can look exactly as **you** want it to look, and will do exactly what **you** want, how **you** like it done.

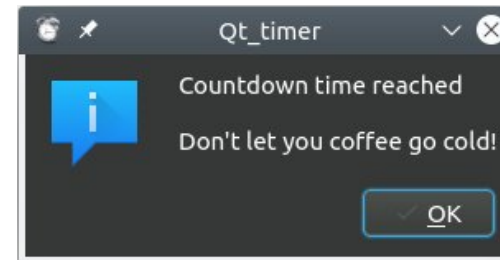
This one has an alarm, a countdown timer, custom dialogs to alert you, plays a sound on time completion and displays an icon in the system tray with a custom right-click menu. What more could you want?



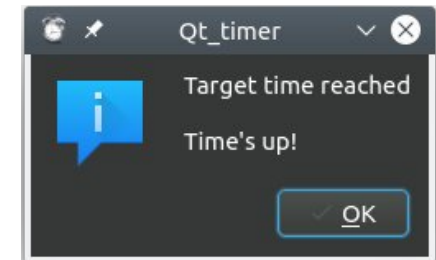
The timer module



The alarm module



Timer alert



Alarm alert



About box

There is a lot more in this application than we are used to, but most of it is simple or we have seen before. Here it is, with the methods closed up as usual.

```
qt_timer.py x
1  #!/usr/bin/env python
2
3  import sys
4  import time
5  import datetime
6  import subprocess
7  from PyQt5.QtCore import *
8  from PyQt5.QtGui import *
9  from PyQt5.QtWidgets import *
10
11  import timer_ui
12
13
14  class Timer(QMainWindow, timer_ui.Ui_Form):
15      def __init__(self):
66      def start_loop(self):
71      def exitEvent(self):
75      def aboutEvent(self):
82      def keyPressEvent(self, e):
87      def alarmInit(self):
93      def countdownInit(self):
100      def setAlertMsg(self):
102      def get_alarm_hr(self):
104      def get_alarm_min(self):
106      def get_cdn_hr(self):
108      def get_cdn_min(self):
110      def get_cdn_sec(self):
112      def getTime(self):
114      def chck_alarm_hr(self):
119      def chck_alarm_min(self):
124      def doIt(self):
145      def doAlarm(self):
165      def doCountdown(self):
187      def exitApplication(self): # Exit point
191      def trayIconActivated(self, reason):
199      def closeEvent(self, event):
208
209
210  if __name__ == '__main__':
```

Create a new directory, and add suitable icons for the timer and the alarm.

Create an executable file named `update_res.sh` with contents:

```
#!/usr/bin/env bash

pyuic5 timer_ui.ui > timer_ui.py
```

When you have used designer to create the user interface, execute the `update_res.sh` file to create the file `timer_ui.py`, which we will import. Do this every time you make changes in designer.

Add a sound file to your folder. I used a system sound:

```
/usr/share/sounds/freedesktop/stereo/complete.oga
```

I copied this to my application folder, as system sounds can change or even be removed

First up we have to create the user interface in designer.

There are some new widgets on this form, including a `stackedWidget`. For this, I am going to start from scratch. Open designer and create a new `Widget`, set `Width` and `Height` to 432 and 272, set the stylesheet to `'background-color: #353836;\ncolor: white'`. Change the window title and window icon as you prefer.

Add a frame, set `x`, `y` to 10, 10 and `Width`, `Height` to 411, 221.

Set the stylesheet to:

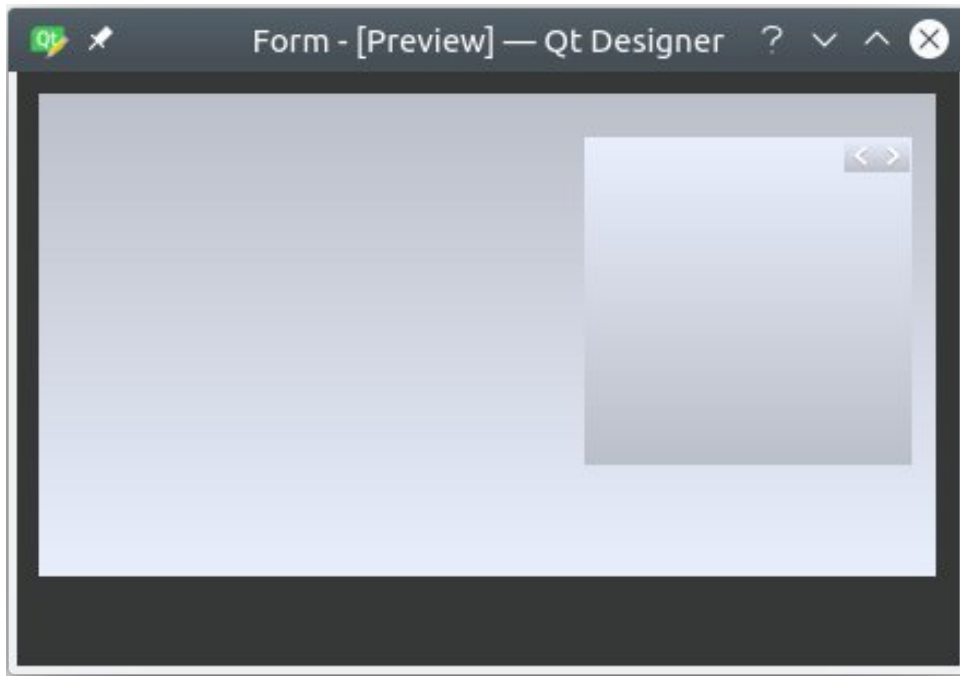
```
background-color: qlineargradient(x1: 0, y1: 0, x2: 0, y2: 1,
stop: 0 #bbbfc8, stop: 1 #e9eefd)
```

Add a stacked widget, set `x`, `y` to 250, 20 and `Width`, `Height` to 150, 150.

Set the stylesheet to the reverse of the previous gradient:

```
background-color: qlineargradient(x1: 0, y1: 0, x2: 0, y2: 1,
stop: 0 #e9eefd, stop: 1 #bbbfc8);
```

This allows us to more easily see where to place the other widgets (next page, top left).



Add two radiobuttons to the frame and two spinboxes to the stacked widget. If you start typing the name of the widget you want in the filter box at the top of the widget box, you will more easily find the widgets you need. This also works in the property editor. If you now click on the right arrow button in the top of the stacked widget, you will be given a new, empty page. Add three spinboxes to this page. These arrow buttons will not be visible in the final application.

Add a QLineEdit to the bottom of the form and then add a pushButton to the bottom right of the main widget. Add labels until everything looks like the first two images above. The stylesheet for all of the labels on the frame and the stacked widget should be:

```
background: transparent; color: black
```

Add this stylesheet to the two radioButton:

```
color:black;
background: transparent;
background-color: #CACED9;
border: 2px solid grey;
border-radius: 4px;
padding: 4px;
```

Then set the topmost radiobutton checked property to checked, and set the text property of both radio buttons appropriately.

And this stylesheet to all five spinboxes.

```
color: black;
```

The only thing that really matters in all of this is the object names of the widgets with which we are going to interact. These should be:

btn\_start for the pushbutton

label\_time for the blue label at the bottom – stylesheet I used was color:#80A3F7

rb\_alarm for the Alarm radio button

rb\_cdn for the Countdown radio button

msg\_edit for the QLineEdit

sb\_alarm\_hour

sb\_alarm\_min

sb\_cdn\_hrs

sb\_cdn\_mins

sb\_cdn\_secs for the five spin boxes

The minimum and maximum values of the spinboxes should be set to:

0, 23 for the hour spinboxes and 0, 59 for the other three.

I also added a line, just my personal preference:

```
orientation vertical
```

```
x, y 230, 30
```

```
Width 3
```

```
Height 130
```

```
stylesheet color: black
```

Test things out with control – r, and try changing pages on the stacked widget.

Save it as timer\_ui.ui. Execute update\_res.sh to generate the python code for this interface.

**The code**

```
#!/usr/bin/env python

import sys
import time
import datetime
import subprocess
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *

import timer_ui
```

The time and datetime imports are new. This is how we get and check the time. We now import timer\_ui for our user interface.

The class definition is like previous projects

```
class Timer(QMainWindow, timer_ui.Ui_Form):
```

This is followed by the `__init__` method definition.

```
def __init__(self):
    super(self.__class__, self).__init__()
    self.setupUi(self)

    self.exitOnClose = False

    my_exit = QAction("Exit", self)
    my_exit.triggered.connect(self.exitEvent)
    my_about = QAction("About", self)
    my_about.triggered.connect(self.aboutEvent)
    menu = QMenu(self)
    menu.addAction(my_exit)
    menu.addAction(my_about)

    icon = QIcon()
    icon.addPixmap(QPixmap("timer.png"))
    self.tray_icon = QSystemTrayIcon()
    self.tray_icon.setIcon(QIcon(icon))
    self.tray_icon.setContextMenu(menu)
    self.tray_icon.setToolTip(
        'Display an alert at a time or after a delay')

    self.tray_icon.activated.connect(self.trayIconActivated)
    self.tray_icon.show()
    self.btn_start.clicked.connect(self.doIt)
```

```
self.hr_now = datetime.datetime.now().hour
self.min_now = datetime.datetime.now().minute
self.stackedWidget.setCurrentIndex(0)
self.sb_alarm_hour.setValue(self.hr_now)
self.sb_alarm_min.setValue(self.min_now)
self.cdn_hr, self.cdn_min, self.cdn_sec = '0', '0', '0'
self.alert_msg = 'Done'
self.msg_edit.setText(self.alert_msg)
```

```
self.rb_alarm.toggled.connect(self.alarmInit)
self.rb_cdn.toggled.connect(self.countdownInit)
self.sb_alarm_hour.valueChanged.connect(self.get_alarm_hr)
self.sb_alarm_min.valueChanged.connect(self.get_alarm_min)
self.sb_cdn_hrs.valueChanged.connect(self.get_cdn_hr)
self.sb_cdn_mins.valueChanged.connect(self.get_cdn_min)
self.sb_cdn_secs.valueChanged.connect(self.get_cdn_sec)
self.msg_edit.textChanged.connect(self.setAlertMsg)
QTimer.singleShot(0, self.start_loop)
```

This looks like a lot of code, but there are a lot of widgets to initialize. The first 20 or so lines are almost a copy of the scratchpad code. We initialize the form, override the close event so we can use the right click menu to shut down, then we implement the right click menu and set up the icons.

The tray icon is then connected to some code, and the `btn_start` sets the selected timer in motion.

Next, the datetime module is brought into play. The `datetime.datetime.now()` method returns a tuple of values representing the current time in the order year, month, day, hour, minute, seconds, and microseconds. All of these are integers, with the last one representing the decimal part of seconds with no trailing zeros. So it can be up to six digits long, as Linux provides a microsecond resolution. These values can be represented by their names, as I have done here, using only the hour and minute values for the alarm. These are assigned to the variables `hr_now` and `min_now`.

The pages of the `stackedWidget` are referenced by zero based integers. We default to the first page, the alarm settings, so use index 0.

We then set the alarm spin boxes to the current hour and minute to provide a start point.

The variables for the countdown hours, minutes and seconds are initialised to zero using the unpacking method I explained earlier. Here I am using a tuple of three zeros to fill the variables.

The default alert message is set to 'Done' and this is inserted into the QLineEdit widget, msg\_edit.

The radio buttons each have their own init methods to be used when activated.

The spinbuttons are set to the values we previously initialised.

If the text in the QLineEdit changes then this is made the new alert message.

The last line in this method uses a QTimer single shot object to kick things off. This is a timer that fires once, has a timeout value and a target. The first argument of zero means that this will never timeout, or rather will live for the life of the application unless otherwise interrupted. The second argument is the routine that it starts. Here it starts the start\_loop method.

```
def start_loop(self):
    while True:
        time.sleep(0.1) # seconds between updates
        self.getTime()
        QApplication.processEvents()
```

Although this is the main loop, it is very simple. While true: will loop forever or until the application is terminated. It goes to sleep for 100 milliseconds, calls the getTime method, processes all application events then the loop starts again.

The **exitEvent**, **AboutEvent** and **keyPressEvent** methods are taken directly from the scratchpad application although the about event uses the much simpler method of implementing a popup box.

```
def exitEvent(self):
    self.exitOnClose = True
    self.close()
    sys.exit()

def aboutEvent(self):
    msgText = (" An alarm & countdown timer \n" +
              " tray resident application\n" +
              " Programmed in Python3\n" +
              " and PyQt5\n")
    dialog_msg = ' Qt5 Timer\n\n' + msgText
    QMessageBox.information(form, 'About', dialog_msg)
def keyPressEvent(self, e):
    if e.key() == Qt.Key_Escape:
        self.exitApplication()
    if e.key() == Qt.Key_Return:
        self.doIt()
```

The **alarmInit** and **countdownInit** methods do what they say.

```
def alarmInit(self):
    self.hr_now = datetime.datetime.now().hour
    self.min_now = datetime.datetime.now().minute
    self.stackedWidget.setCurrentIndex(0)
    self.sb_alarm_hour.setValue(self.hr_now)
    self.sb_alarm_min.setValue(self.min_now)

def countdownInit(self):
    self.stackedWidget.setCurrentIndex(1)
    self.sb_cdn_hrs.setValue(0)
    self.sb_cdn_mins.setValue(0)
    self.sb_cdn_secs.setValue(0)
    self.sb_alarm_hour.setValue(0)
    self.sb_alarm_min.setValue(0)
```

They set things up for the alarm or countdown choices to operate, according to the parameters that were set in the interface.

The **setAlertMsg** method was called from `__init__` to reset the message that will be displayed when the alarm is triggered.

```
def setAlertMsg(self):
    self.alert_msg = self.msg_edit.text()
```

The **get\_alarm...** and **get\_cdn...** methods get and set the current values from the spinbuttons when the value changes.

```
def get_alarm_hr(self):
    self.alarm_hr = str(self.sb_alarm_hour.value())

def get_alarm_min(self):
    self.alarm_min = str(self.sb_alarm_min.value())

def get_cdn_hr(self):
    self.cdn_hr = str(self.sb_cdn_hrs.value())

def get_cdn_min(self):
    self.cdn_min = str(self.sb_cdn_mins.value())

def get_cdn_sec(self):
    self.cdn_sec = str(self.sb_cdn_secs.value())
```

The **getTime** method calls the imported `datetime.datetime.now` method, which returns the current date and time in two parts, the date and the time.

```
def getTime(self):
    self.label_time.setText(
        str(datetime.datetime.now()).split()[1][:8])
```

This is converted to a string, and the string split method used to give us a list of two strings like this:

```
['2019-01-18', '11:20:36.795389']
```

We are interested in only the time part [1] and only the hours minutes and whole seconds [:8].

This string is put into label\_time to update the current time display (in blue on my version) every second.

The **check\_alarm\_hr** and **check\_alarm\_min** routines check if the alarm times have yet been reached

```
def chck_alarm_hr(self):
    self.hr_now = int(datetime.datetime.now().hour)
    set_hrs = self.sb_alarm_hour.value()
    if not self.sb_alarm_hour == self.hr_now:
        self.sb_alarm_hour = self.hr_now

def chck_alarm_min(self):
    self.min_now = datetime.datetime.now().minute
    set_mins = self.sb_alarm_min.value()
    if not self.sb_alarm_min == self.min_now:
        self.sb_alarm_min = self.min_now
```

The **doIt** method gets the current system time status, compares it to the values we set and, if a match is found, calls the appropriate method.

```
def doIt(self):
    currentTime = datetime.datetime.now()
    self.current_hour = currentTime.hour
    self.current_minute = currentTime.minute
    self.current_second = currentTime.second
    self.cdn_set = (self.sb_cdn_hrs.value() +
                   self.sb_cdn_mins.value() +
                   self.sb_cdn_secs.value())

    self.alarm_set = (self.sb_alarm_hour.value() +
                     self.sb_alarm_min.value())

    if self.alarm_set:
        self.doAlarm()
```

```
if self.cdn_set:
    self.doCountdown()
```

The **doAlarm** method checks the current time minus the set time in seconds. If the result is non-zero, then sleep for 1 second, reduce the time to wait by one second, and keep the form hidden. When zero is reached, a sound is played and a dialog popped up. As the dialog is modal, the application waits until it has been closed, and then re-shows the form. If you want to include the sound option, then you will need to have the sox utility installed from the repository (if not already installed). Sox includes the play utility. The -q option suppresses text progress output from the play command.

```
def doAlarm(self):
    if self.alarm_set:
        self.real_secs = (int(self.current_hour) * 3600 +
                          int(self.current_minute) * 60)
        self.secs = (self.sb_alarm_hour.value() * 3600 +
                    self.sb_alarm_min.value() * 60 -
                    self.current_second)
        self.timer_secs = self.secs - self.real_secs
        while self.timer_secs:
            time.sleep(1)
            self.timer_secs -= 1
            form.hide()

        subprocess.Popen('play complete.oga -q', shell=True)
        dialog_msg = 'Target time reached\n\n' + self.alert_msg

        QMessageBox.information(form, 'Qt_timer', dialog_msg)
        form.show()
```

The **doCountdown** method works in exactly the same way as doAlarm, using the spinButton values for comparison.

```
def doCountdown(self):
    if self.cdn_set:
        self.real_secs = (int(self.current_hour) * 3600 +
                          int(self.current_minute) * 60 +
                          self.current_second)
        self.timer_secs = (self.sb_cdn_hrs.value() * 3600 +
                            self.sb_cdn_mins.value() * 60 +
                            self.sb_cdn_secs.value())
        #self.timer_secs = self.secs
        while self.timer_secs:
            time.sleep(1)
            self.timer_secs -= 1
            form.hide()

        # mixer.music.play()
        subprocess.Popen('play complete.oga -q', shell=True)
        dialog_msg = 'Countdown time reached\n\n' + self.alert_msg
```



```

QMessageBox.information(form, 'Qt_timer', dialog_msg)
self.countdownInit()
self.getTime()
form.show()

```

The **exitApplication** method simply closes the form, then exits the application.

```

def exitApplication(self): # Exit point
    self.close()
    sys.exit()

```

The **trayIconActivated** method is again taken directly from the scratchpad application

```

def trayIconActivated(self, reason):
    if reason == QSystemTrayIcon.Context:
        self.tray_icon.contextMenu().show()
    elif reason == QSystemTrayIcon.Trigger:
        self.show()
        self.raise_()
        self.activateWindow()

```

The **closeEvent** method processes what happens when the window close button is clicked.

```

def closeEvent(self, event):
    if self.exitOnClose:
        self.tray_icon.hide()
        del self.tray_icon
        event.accept()
    Else:
        self.hide()
        event.setAccepted(True)
        event.ignore()

```

The way the application is terminated is slightly different from the scratchpad application, but the result is the same. There are always more than one way to accomplish a task.

The final part of the code has some additional code. A routine to prevent more than one instance from running, and another to centralize the application on screen. These are standard routines that can be dropped into an application as required. This kind of code is often known as 'boilerplate code'.

```

if __name__ == '__main__':
    app = QApplication(sys.argv)
    form = Timer()

```

```

# Single instance check
Try:
    import socket
    s = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM)
    s.bind('\0postconnect_gateway_notify_lock_timer')
except socket.error as e:
    error_code = e.args[0]
    error_string = e.args[1]
    print("Error {}, {}. Exiting".format(error_code,
error_string))
    QMessageBox.warning(form,
        "Timer",
        "Process already running, Exiting!")
    sys.exit(0)

# centralize
qr = form.frameGeometry()
cp = QDesktopWidget().availableGeometry().center()
qr.moveCenter(cp)
form.move(qr.topLeft())

form.show()
app.exec_()

```

This is quite a complex little application. Although each little bit of the code is fairly simple, there are a lot of little bits. This can make it difficult to follow.

Try removing some of the code, perhaps all the code for the alarm, and then try to follow the countdown code. You could also add some print statements to see what values are currently referenced by variables. This is a common debugging practice.

```

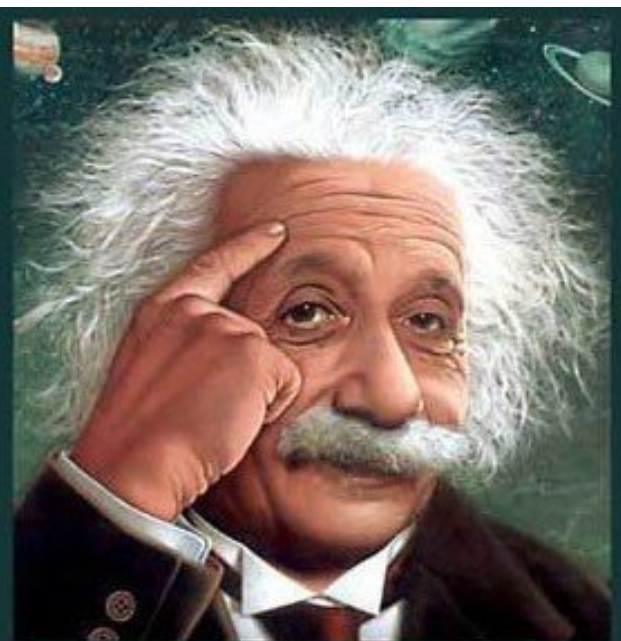
def get_alarm_hr(self):
    self.alarm_hr = str(self.sb_alarm_hour.value())
    print("Countdown seconds is set to :", self.cdn_sec)

```

I do this all the time. Note that if you try to copy and paste the above line, you may get the decorative 66 – 99 quote marks. You will have to change these in the editor. When you are happy with the code, remove the print statements, or put a # mark at the start of each to disable them.



The banner features a colorful speech bubble icon on the left, the text "PCLOS-Talk Instant Messaging Server" in blue and black, and a red speech bubble icon on the right. Below the text, it says "Sign up TODAY! http://pclostalk.pclosusers.com".



It's easier than  $E=mc^2$   
It's elemental  
It's light years ahead  
It's a wise choice  
It's Radically Simple  
It's ...

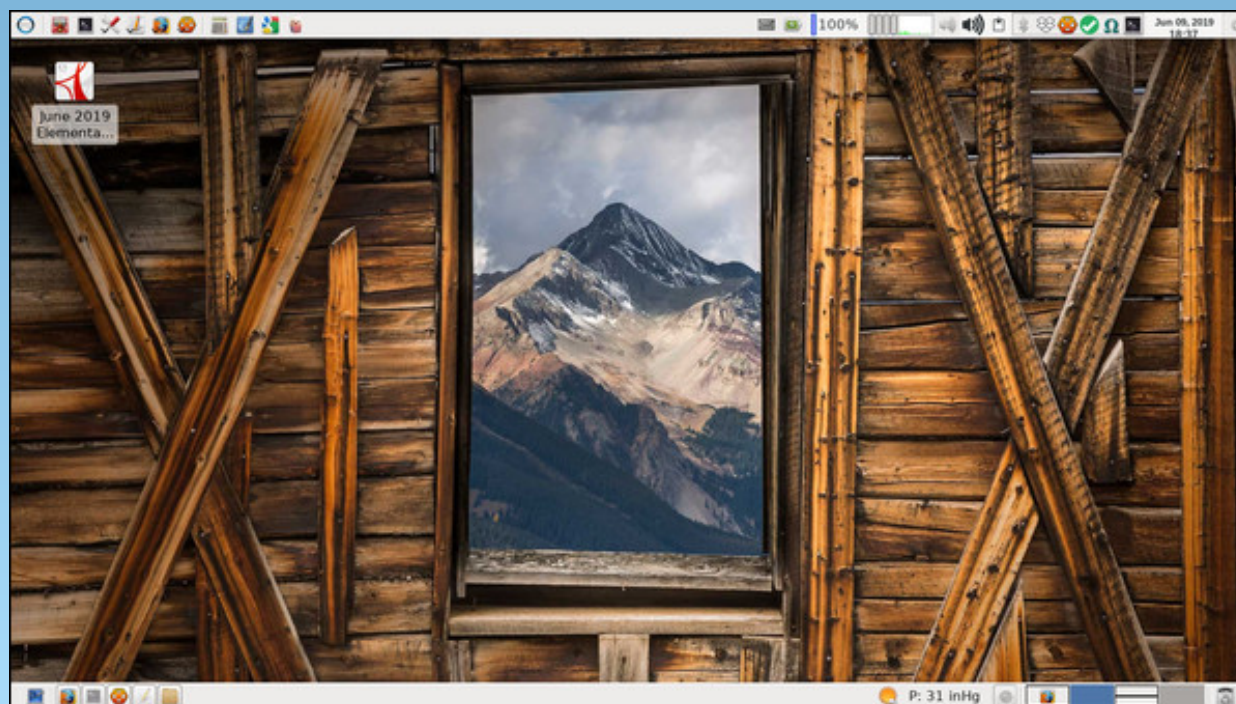


# TorrentFreak

The place where **breaking news**,  
BitTorrent and copyright collide



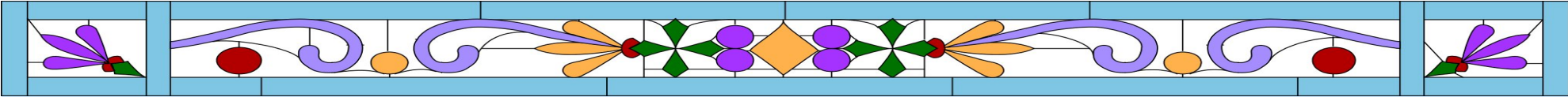
## Screenshot Showcase



Posted by *parnote*, on June 9, 2019, running Xfce.



# *ms\_meme's Nook: Who Can Say*



Who can say why Texstar tried  
His skill applied only time  
And who can say why he felt the need  
Where his path will lead only time

Who can say why Tex gives and gives  
Why his OS lives only time  
And who can say what tomorrow will hold  
What will unfold only time

Who can say why an OS grows  
Why this OS we chose only time  
And who can say why we come to meet  
Our friends to greet only time  
Who can say only time

Who can say when we will part  
Will love remain in our heart  
And who can say what is meant to be  
Will you remember me  
Who can say only time

**MP3**

**OGG**

# PCLinuxOS Puzzled Partitions

			5	2		9	4	
		6						
		4	1			7		3
	9							
	2	5			6	1		
8								2
3				1				
6			2	3			9	8
			9		8			

**SUDOKU RULES:** There is only one valid solution to each Sudoku puzzle. The only way the puzzle can be considered solved correctly is when all 81 boxes contain numbers and the other Sudoku rules have been followed.

When you start a game of Sudoku, some blocks will be prefilled for you. You cannot change these numbers in the course of the game.

Each column must contain all of the numbers 1 through 9 and no two numbers in the same column of a Sudoku puzzle can be the same. Each row must contain all of the numbers 1 through 9 and no two numbers in the same row of a Sudoku puzzle can be the same.

Each block must contain all of the numbers 1 through 9 and no two numbers in the same block of a Sudoku puzzle can be the same.



## SCRAPPLER RULES:

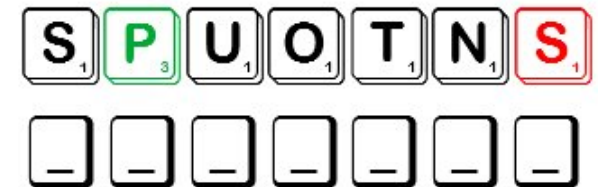
1. Follow the rules of Scrabble®. You can view them [here](#). You have seven (7) letter tiles with which to make as long of a word as you possibly can. Words are based on the English language. Non-English language words are NOT allowed.
2. Red letters are scored double points. Green letters are scored triple points.
3. Add up the score of all the letters that you used. Unused letters are not scored. For red or green letters, apply the multiplier when tallying up your score. Next, apply any additional scoring multipliers, such as double or triple word score.
4. An additional 50 points is added for using all seven (7) of your tiles in a set to make your word. You will not necessarily be able to use all seven (7) of the letters in your set to form a "legal" word.
5. In case you are having difficulty seeing the point value on the letter tiles, here is a list of how they are scored:
  - 0 points: 2 blank tiles
  - 1 point: E, A, I, O, N, R, T, L, S, U
  - 2 points: D, G
  - 3 points: B, C, M, P
  - 4 points: F, H, V, W, Y
  - 5 points: K
  - 8 points: J, X
  - 10 points: Q, Z
6. Optionally, a time limit of 60 minutes should apply to the game, averaging to 12 minutes per letter tile set.
7. Have fun! It's only a game!



Double Word



Triple Word



Possible score 275, average score 193.

Download Puzzle Solutions Here



# PCLinuxOS Word Find: July 2019

## Astronomy Terms

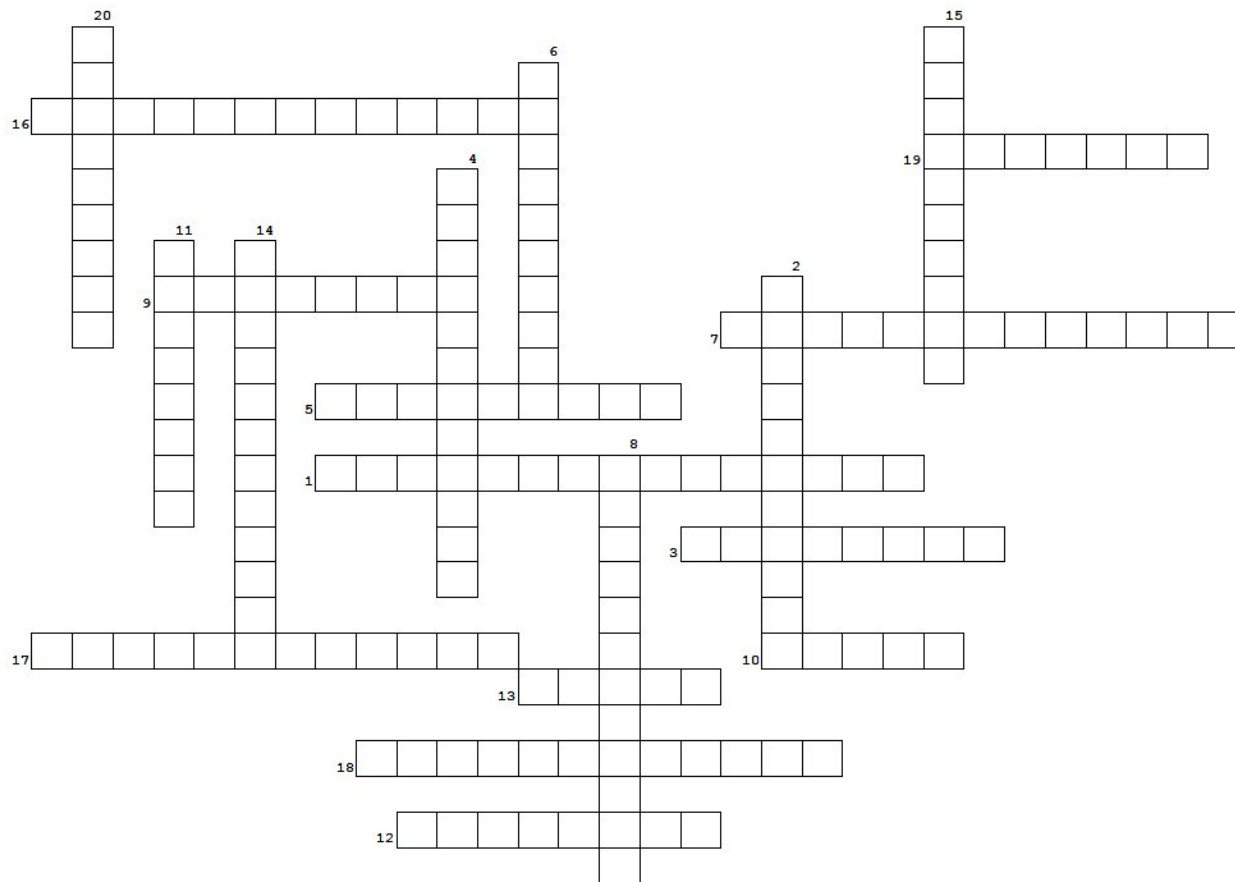
I G T F S Q S M G G S M G F H C I F N O S A H E X X A J X R  
 T N O M M D Z I P S K Q A G O J Z I D X S U N U S G S I L A  
 A I A H Z U A E V A N M I L E W E I N O R T S A P A S E K B  
 W X V H G N V E R W S R W J T H F N I T J R A E U V E O Z J  
 Y A T F T G S O B Y S O H T C D D B Y I H C O S L L N D E S  
 W W E Q S L V C A S D E G S O G V P K B Y B F P S V S Q N R  
 J Y R H H O V B W K Y T N M D H Z T K R P G Q I A Q S R I Z  
 B B R V H B J S R I C E L E J V F H Q O E I N L R O E X T H  
 Q P E X K U D I P S G M L Y B P G T T L R X M C O G L A H G  
 T G S F G L D F Z E C P W I L O R X K A N T N E S K T R K N  
 C Q T K O A Y O G E C I O M A L R K G C O L O L K P H O W U  
 I M R C N R I S F L C T S O D B O P A I V S I A I P G I T S  
 C L I B E C P H Z F F R R L Q D H Z H T A E L T D B I Q S A  
 B E A W T L L Y C E D N O U U Z D K U P T N E O M O E D M R  
 X I L F M U L S E B M F L F M N U W E I H K H T A W W O R B  
 I T H U Z S J G O M X G N H S Y A Y D L E M I W M P M M Z M  
 Y L I G W T V G P L A N E T O I D R T L E B R E P I U K B U  
 L E N G X E Z Z B Q S Y A P J C L U H E O Z E E R B Z R O N  
 M B T L R R B O U R M T T B V Q H O W P N T P Z T C X G D E  
 S N E O X E D L D Y Z Y I Y Y T T T I U G O M L J G D B E P  
 W E R T L I T L A I X A M C J C E Q H R T N I F F E J C I R  
 N L S A I I Q E O U A O W S E N Q R N D O X P L X N L Y G V  
 L L T T W P J P M R U C C Y A B K A B Q K C Z K E I M K X Z  
 F A E E G I R E P O B D X L H A Y J E O W Z M E P H R S A V  
 N N L N X R E T G U L C P I R V X K N P R Z T T A V P O G Y  
 B A L X S X N Q F J N O U V T G Z S E K L W I H X P S A F C  
 M V A V R K D H F U X Q B M C V V B D Q X C G A A C W P Q W  
 K G R Y J Y P K V E S U N S P O T B O Z R Z V G N I N A W Z  
 E P O C S O R T C E P S J R A T S T N A I G D E R D W U L C  
 Q E F N T Q V O T Y E L L O W D W A R F V P X V I Y A K E W

apastron	aphelion
axial tilt	Bailey's beads
bolometer	cislunar
Coriolis force	Deneb
ecliptic	elliptical orbit
exoplanet	egenschein
globular cluster	hypernova
interstellar	Van Allen belt
Kuiper belt	meteor
nadir	penumbra
perigee	perihelion
planetoid	probe
pulsar	red giant star
solstice	spectroscope
spectrum	sunspot
terrestrial	total eclipse
waning	waxing
weightlessness	yellow dwarf
zenith	zodiac

[Download Puzzle Solutions Here](#)




# Astronomy Terms Crossword



1. a large compact spherical star cluster, typically of old stars in the outer regions of a galaxy.
2. a patch of very faint nebulous light sometimes seen in the night sky opposite the position of the sun.
3. the point in the orbit of a planet at which it is furthest from the sun.
4. a zone of energetic charged particles, most of which originate from the solar wind, that are captured by and held around a planet by that planet's magnetic field.
5. a very energetic supernova thought to result from an extreme core-collapse scenario.
6. the point in the orbit of a planet, asteroid, or comet at which it is closest to the sun.
7. the time in the spring when the sun crosses the equator, and when night and day are of equal length.
8. a dying star in the last stages of stellar evolution.
9. the shadow cast by the earth or moon over an area experiencing a partial eclipse.
10. the point of the celestial sphere that is directly opposite the zenith and vertically downward from the observer.
11. the point at which the stars of a binary system are farthest apart
12. a planet composed mainly of elements heavier than hydrogen and helium, such as oxygen, carbon, nitrogen, and sulfur.
13. a first-magnitude star in the constellation of Cygnus
14. occurring or situated between stars.
15. a region of the solar system beyond the orbit of Neptune, believed to contain many comets, asteroids, and other small bodies made of ice.
16. an apparent force that as a result of the earth's rotation deflects moving objects (such as projectiles or air currents) to the right in the northern hemisphere and to the left in the southern.
17. an apparatus for producing and recording spectra for examination.
18. spots of light that appear to encircle the moon, resembling a string of luminous beads, visible immediately before and after a total eclipse.
19. the point in the orbit of the moon or a satellite at which it is nearest to the earth.
20. a planet which orbits a star outside the solar system.

[Download Puzzle Solutions Here](#)

# Mixed-Up-Meme Scrambler



The \_\_\_\_\_  
is up and running.

Clever  
**WHERDS**    \_\_\_\_\_  \_\_\_\_\_

Grin  
**SKRIM**    \_\_\_\_\_

Single  
**WEDUN**    \_\_\_\_\_  \_\_\_\_\_

Feel  
**CHOUT**      \_\_\_\_\_

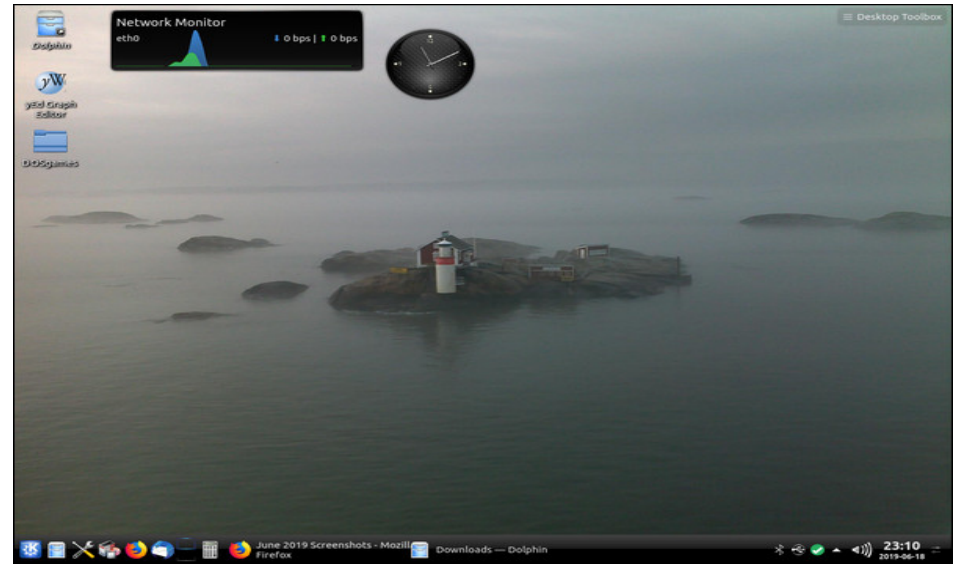
Use the clues to unmix the letters to make a new word. Remix the letters in the red boxes to solve the puzzle.

[Download Puzzle Solutions Here](#)

# More Screenshot Showcase



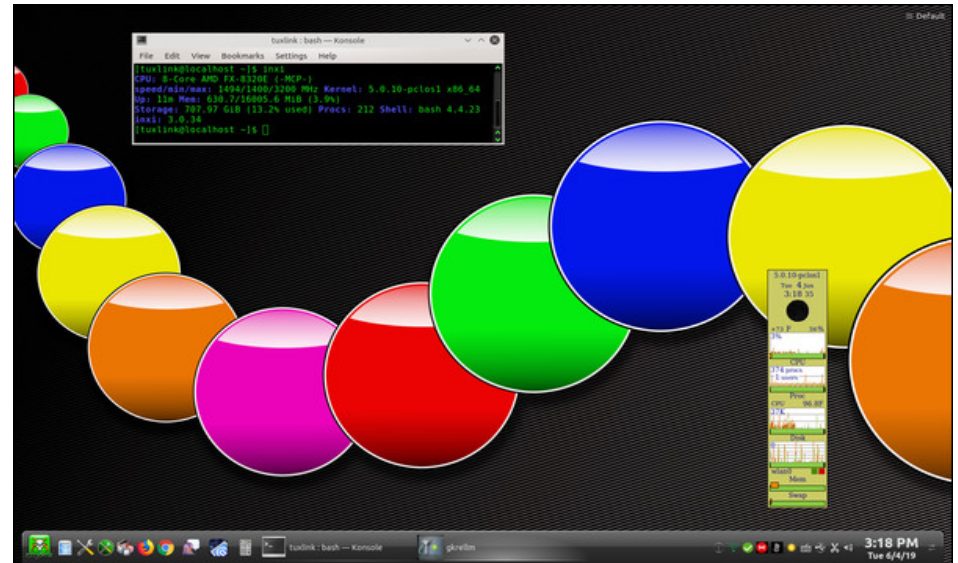
Posted by pujujiop, on June 11, 2019, running Xfce.



Posted by rogerwilco, on June 18, 2019, running KDE.



Posted by Texstar, on June 6, 2019, running icwm.



Posted by tuxlink, on June 4, 2019, running KDE.