

The PCLinuxOS magazine

Volume 156

January, 2020

**HAPPY
NEW
YEAR!**



**R.I.P. Sproggy!
You Will
Be Missed**

In This Issue...

- 3 From The Chief Editor's Desk...**
- 4 Screenshot Showcase**
- 5 R.I.P, Sproggy! You Will Be Missed!**
- 7 Screenshot Showcase**
- 8 ms_meme's Nook: PCLinuxOS Is All There Is**
- 10 PCLinuxOS Recipe Corner: Impossibly Easy Bacon Pie**
- 11 De-Googling Yourself, Part 8**
- 16 Screenshot Showcase**
- 17 Short Topix: Mozilla Removes ,
AVG Firefox Extensions Over Snooping Claims**
- 22 GIMP Tutorial: Photo Filmstrip**
- 24 Screenshot Showcase**
- 25 Tip Top Tips: Move Firefox 71's Tab Bar
BACK To The Bottom**
- 27 PCLinuxOS Family Member Spotlight: rgradle**
- 29 IPTV: Free Streaming Over The 'Net**
- 31 Screenshot Showcase**
- 32 PCLinuxOS Bonus Recipe Corner: Ground Beef Stew
over Garlic Mashed Potatoes**
- 33 Casual Python, Part 12**
- 45 Screenshot Showcase**
- 46 ms_meme's Nook: PCLOS Capers**
- 47 PCLinuxOS Puzzled Partitions**
- 51 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: Sproggy, Timeth, ms_meme, Meemaw

Magazine Layout: Paul Arnote, Meemaw, ms_meme

HTML Layout: YouCanToo

Staff:

ms_meme
Meemaw
Gary L. Ratliff, Sr.
Daniel Meiß-Wilhelm
daishi
Alessandro Ebersol

CgBoy
YouCanToo
Pete Kelly
phorneker
Khadis Thok
Smileeb

Contributors:

Ramchu

The PCLinuxOS Magazine is released under the Creative Commons Attribution-NonCommercial-Share-Alike 3.0

Unported license. Some rights are reserved.

Copyright © 2018.



From The Chief Editor's Desk ...

So many of our PCLinuxOS family members live literally with a few hours of each other, but so few of us have ever met in person. The only PCLinuxOS users that I have ever met are those that I convinced to give PCLinuxOS a try. And I already knew them, personally. I already considered them friends. Outside of my immediate family that lives in my own home, that number is quite low. I try to not be that annoying, evangelistic PCLinuxOS “preacher.” But, if I see a situation where PCLinuxOS can help, I’m quick to make the suggestion that they at least give the Live media a try.

The PCLinuxOS family is unique in its friendliness and helpfulness. Over time, many of us make “friends” with other PCLinuxOS users from all around the globe. I know I consider many PCLinuxOS users friends from all corners of the globe: England, Ireland, Germany, China, Thailand, South Africa, Canada, Mexico, Brazil, France, Greece, Poland, Netherlands, and of course, the U.S. I’m sure there are other countries harboring PCLinuxOS users that I missed, but that list is just off of the top of my head.

Life happens, and people come and go. That’s just normal. But others stay and stay, and those bonds strengthen over the years. And yet, despite those bonds, they never meet, face-to-face. Often times, it’s logistically impractical to ever meet. They may live on another continent, separated by a vast ocean. At other times, it may be political divisions that prevent “friends” from ever meeting.

But when you live only hours away, it’s exceptionally sad to never meet those with whom you share a special bond face-to-face. Years and years have been invested in developing those bonds of friendship, yet a few measly hours prevent ever meeting up face-to-face.



The recent and unexpected death of our beloved PCLinuxOS family member Sproggy has brought this thought front and center in my mind. Sproggy and I shared that special friendship bond. However, logistically, I think we both knew how improbable it was that we’d ever be able to meet up. The vast Atlantic Ocean separated us, with him living in the U.K., and me in the U.S. Still, we made plans for the

day when we would finally be able to meet up. But the distance never deterred us from wishing, hoping, and making those plans. Now, with his passing, those plans will forever remain just unfulfilled wishes and hopes. With all of the time we spent talking and working together, I feel as if I *know* his wife and kids. I know many of the trials, tribulations and troubles he faced. I even knew he was 5’ 10” (178

cm) tall, and weighed around 220 pounds (99.8 Kg). He also “knew” my wife and kids, as well as many of the trials, tribulations and troubles I’ve stared down.

There is a point where someone who is “just another user” becomes more than that. They become friends. They become part of your extended family. They enhance your very being. The sum is greater than the individual parts. Over the years, you both develop that special bond with one another.

There are many around the PCLinuxOS family, past and present, that I’d love to meet up with. I’d love to share tacos and a beer with Tex. I’d love to accompany ms_meme on one of her daily strolls. I’d love to meet up with Meemaw for a nice lunch over some awesome Kansas City BBQ. I’d like to hang out with Crow, so he could show me those beautiful sights in Mexico and some local, authentic Mexican food. *alaveC\$* seems like he’d be a blast to hang out with and throw back a pint. I’d love to visit with Rudge and JRex, and swap recipes. Mr Cranky Pants-YouCanToo would be great to do the same thing with. Joechimp would be fun to visit with or hang out with, no matter the situation. I wonder if Old-Polack’s speech is as colorful as his posts. In fact, there are a lot of you that I would love to meet up with, so my apologies if I didn’t list you. You’re probably on the list, but I don’t have room here to list all of you.

So, if there’s that PCLinuxOS family member that you’ve developed that special bond and friendship with, by all means, make every attempt to meet them. If you live in a big city, look for other PCLinuxOS users in your area. Meet them for a cup of coffee at a coffee shop, or even at a fast food restaurant. You could even make it a monthly or weekly event. But, if only a few hours separate you, there really is no excuse to meet at least once. There are things that simply cannot be conveyed online.

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

GORILLABOX

Preinstalled with PCLinuxOS KDE
New. Fast. Customizable.
Order Yours Today!



Screenshot Showcase



Posted by Chrisripp, on December 25, 2019, running enlightenment.

R.I.P., Sproggy! You Will Be Missed

by Paul Arnote (parnote)

On December 23, 2019, our beloved PCLinuxOS family member, Sproggy, lost his battle with cancer.

The news was delivered by his oldest son, Kyle. The [post](#) read:

I didn't know where to put this. My name is Kyle, and my dad was forum member Sproggy or Kori.

I know he has been a long time user of your software, and he always loved being part of your community. It is with sad news that I am informing you that my dad passed away on 23rd December 2019.

He had been battling cancer for a couple of years. It started in his bowel and ended in his lungs.

Even to the end, he was still very jokey, and wanted me to let you know he always felt at home with you all here.

He left me with instructions that if anyone would like the master copies of his artwork that I can share them with them. Please message and I can send them zipped separately in email.

His PC is going to stay with me, so I will have access to his work.

Thanks,

Kyle

(Follow the previous link to view reaction to the sad news from the PCLinuxOS family.)



Kori Tilcock, a.k.a. Sproggy, a.k.a. Kori Tk

Born on March 7, 1974, Sproggy was only 45 years old when the cancer brought his life to a way-too-early end. He leaves behind his wife of 27 years, Mo, his daughter Neecey, and his two sons, Kyle and Connor. They resided in Bedford, England (the last I knew). His most recent job was as a senior applications engineer for Mastercard, maintaining their databases on the backend of the system.

Remembering Sproggy

When I first joined the PCLinuxOS forum, Sproggy was a moderator. We both hit it off pretty early on. My interactions with him increased a lot when I took

over the editor's role for the magazine. We would chat frequently – usually daily – in the magazine's IRC channel on FreeNode, then called #pclinuxos.mag (it's now #pclosmag).

We would chat about everything and anything. We'd talk about family, politics (particularly anytime there was a General Election coming up in the U.K.), world events, personal trials and tribulations, work, what's for dinner, and sometimes just nonsense. There was hardly a topic we didn't touch on. At that time, the magazine's IRC channel was a hopping place. Joble, Hootiegibbon, CSolis, grnich, ms_meme, AndrezjL, Meemaw, myself and many others frequently hung out there. Sproggy would join in on the conversations with just about everyone.



Sproggy and his wife, Mo

I have fond memories of the “argument” we had on whether milk gravy would/should be considered gravy. He always said no. I would goad him on, telling him he hadn't lived until he had a big platter of biscuits smothered in sausage gravy (and I knew JUST the place to take him, if I ever got the chance). We often talked about the possibility of meeting up,

face to face, should he get the chance to visit the U.S. someday. He and Mo actually did get to make the trip to the U.S. once, but they became ill while here, and we weren't ever able to make the meeting happen. We always talked about throwing back a pint, if and when that happened. Most recently, in early November, he and I made "plans" to head to the shooting range to shoot skeet when he and Mo finally made it over here on another trip.

When I wasn't available for chats, Sproggy would spend a lot of time talking with Meemaw.

Meemaw recalled her conversations with Sproggy with fondness. "I always loved it when Sproggy came into IRC. We discussed a lot of things, but mainly talked about our families or artwork. When he made one of the versions of his Xfce called Phoenix, he actually let me design one of the logo wallpapers that he included. He was always very kind and supportive of my graphics efforts, even though his were so much better. I was chatting with him in November shortly after he started posting his new wallpapers, and shared a couple with me before he posted them. He mentioned that if he couldn't sleep or was in pain, he would go to the computer and do some graphics. I learned so much from him, and will miss him terribly," she said.

Sproggy was a passionate person, and he directed that passion into things that mattered to him. His interests included Linux (particularly PCLinuxOS), the Xfce desktop, and his graphics/artwork. He was or became highly skilled in them all. He created the first iteration of Xfce on PCLinuxOS that I ever tried, and it was his Xfce remaster that made me an avid Xfce fan.

When I first took over the editor's role on the magazine, gseaman (Galen) was doing the monthly layouts for the magazine's HTML edition. When Galen decided to step back from the magazine to refocus on packaging, Sproggy stepped in to do the job. He continued to do the magazine's layout of the

HTML version for the next couple of years, before handing that job over to Rudge.



When a new version of Xfce was forthcoming, Sproggy learned packaging so that he could compile and package up the new version of Xfce. He became quite good at it, too.

Sproggy left his PCLinuxOS family for a while, starting in 2012ish. While he checked in periodically over the ensuing years (he made it a point to tell me that he was a fan of Ika's Xfce community remaster), he returned to his PCLinuxOS family full time in July, 2019, and remained with us up until the end.

He shared his new PCLinuxOS-oriented artwork with his PCLinuxOS family. Working on his artwork was therapeutic for Sproggy, as he'd sometimes wake up in the middle of the night. He wouldn't be able to sleep due to the pain, so he'd sit at his computer and put his creative talents to use.



Through it all, Sproggy never gave us much indication of just how serious his condition was. While I'd like to think that maybe even he didn't know, I know that isn't true. He knew. That's why he came back to the one place that felt like home to him: his PCLinuxOS family.

Perhaps the most telling sign he gave us came in a poignant [post](#) he recently made to the forum. The post was a month and a half before his untimely demise. It read (emphasis mine):

Well, I got asked recently why I have returned to PCLinuxOS after so long away ... the last few years I have been very unwell progressively getting worse ...

Being part of PCLinuxOS community brought me so much joy when I was an active member ... it made me feel like I belonged somewhere for once ... but these last few years I have felt I needed to give something back and share some love with the group for that time.

I have just finished another round of chemo & radio ... I have had 3 tumours removed from my bowel and the tumour in my lung has been shrunk, but it is still there ... I have corrosive esophagitis, duodenitis, blood clots

and nodules littering my lungs, sinus tachycardia, several auto-immune deficiencies caused by the chemo.

I don't know how long life has left so I am here to create art as that's what I enjoy most and share that with you guys ... I hope to see whatever time is left bringing some smiles to people's faces before I am checked out.

Hopefully, this will clear up the curiosity ... peace & love people.

Kori



Without a doubt, PCLinuxOS has lost a valued family member when cancer took Sproggy from us, way ahead of his time. He had a lot more living to do, and a lot more to accomplish. Who knows, in the absence of the cancer, what all he may have gone on to do.

We'll miss you Sproggy! We'll miss your humor, your wit, your insights and all of your contributions yet to come that will remain unfulfilled. We'll miss your fun-loving spirit. Thank you for all that you contributed in the time that you were among us.

All of us here at The PCLinuxOS Magazine extend our most heartfelt condolences to Sproggy's family. He touched many lives here, so we can only imagine the loss you must be feeling.

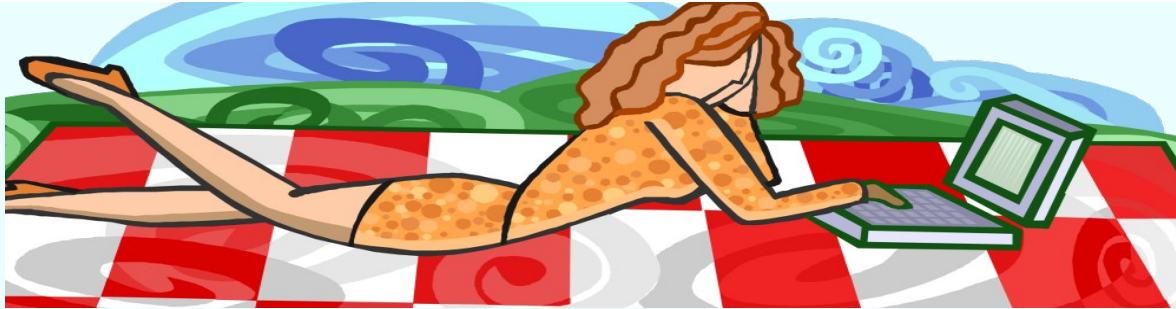
DESTINATION **LINUX**
LINUX IS OUR PASSION

Screenshot Showcase



Posted by present arms, December 2, 2019, running Openbox.

ms_meme's Nook: PCLinuxOS Is All There Is



I remember when I was a very young girl
Our family bought Windows 98
I'll never forget the look on my father's face
As we gathered 'round
And he opened the folders

I stood there in my pajamas
And saw his whole world
Go up in tears
And when he turned it off I said to myself
Is that all there is to Windows

Is that all there is is that all there is
If that's all there is my friend
Then let's stop booting
Let's have a party
And do something naughty
If that's all there is

And when I was twelve years old
My daddy took me to the store
The biggest store on earth
And he said have anything you want
I wanted Windows XP all my friends had it

As I sat there
I had the feeling something was missing
I don't know what
When we left the store I said to myself
Is that all there is to Windows

Is that all there is is that all there is
If that's all there is my friend
Then let's stop booting
Let's have a party
And do something naughty
If that's all there is

And then I fell in love
With the most wonderful boy in the world
We never took long walks by the river
We sat for hours gazing at the virus scanner
We were very much in love

Then one day he went away
And left me with Windows Vista
And when the affair was over I said to myself
Is that all there is to Windows

Is that all there is is that all there is
If that's all there is my friend
I know what
You must be saying to yourself

If that's the way she feels about it
Why doesn't she just end it all
Oh no not me
I'm not ready for that final boot
'Cause I know just as well

As I'm standing here talking to you
Before that final moment comes
I'll be downloading a new OS
And saying to myself

That's not all there is that's not all there is
That's not all there is my friend let's keep on booting
Let's have a party and boot something hearty
PCLinuxOS is all there is



MP3

OGG

PCLinuxOS Recipe Corner



*from the kitchen of
youcantoo*

Impossibly Easy Bacon Pie

INGREDIENTS:

12 slices bacon, crisply cooked, crumbled
1 cup shredded Swiss cheese (4 oz)
1/3 cup chopped onion
3/4 cup Original Bisquick™ *See our Homemade
Bisquick Recipe
1 1/2 cups milk
3 eggs
1/8 teaspoon pepper

DIRECTIONS:

Heat oven to 400 degrees F. Grease 9-inch glass pie plate. Sprinkle bacon, cheese and onion into pie plate.

In medium bowl, beat remaining ingredients until blended; pour into pie plate.

Bake 35 to 40 minutes or until knife inserted in center comes out clean. Let stand 5 minutes before serving.

TIP:

Chop 3 green onions to use, instead of the chopped yellow onion.

Impossibly Easy Bacon Pie can be covered and refrigerated up to 24 hours before baking. You may need to bake a bit longer than the recipe directs since you'll be starting with a cold pie. Watch carefully for doneness.



* See our Homemade Bisquick Recipe at <http://recipes.dm-enterprises.net/?p=recipe&recipe=289>.

Donate NOW



gofundme

De-Googling Yourself, Part 8

by Alessandro Ebersol (Agent Smith)



We have reached the end of this series, and let's reflect on all of that's been discussed before.

Don't Be Evil... No more

We saw that Google's motto is no longer, they have changed, now it is Do the right thing, and that can be many things depending on where you look.

But as we want to preserve ourselves from the "eyes" of the big brother, we will have to take some precautions.

Google's Do's and Don'ts

Well, as we've seen before, I'm not going to go over already-trodden terrain, but to list, in a succinct and straightforward way, what to do to lessen Google's intrusion into our private lives.



Google's DNS

Google's famous DNS, 8.8.8.8, should not be used.

Google may "hide" sites it doesn't want to promote. Use OpenDNS instead (208.67.222.222)

Use Only HTTPS (Hyper Text Transfer Protocol Secure)

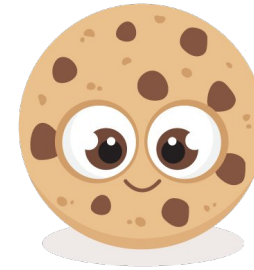
HTTPS (HTTPS) is an implementation of the HTTP protocol over an additional layer of security that uses the SSL/TLS protocol. This additional layer allows data to be transmitted over an encrypted connection and to verify server and client authenticity through digital certificates. The TCP port used by default for the HTTPS protocol is 443.

The HTTPS protocol is used, as a rule, when it is desired to prevent the information transmitted between the client and the server from being viewed by third parties, such as online shopping. The existence in the address bar of a lock (which can be left or right depending on the browser used) demonstrates secure page certification (SSL/TLS). The existence of this certificate indicates the use of the HTTPS protocol and that communication between the browser and the server will take place securely. To verify the identity of the server, double-click the lock to view the certificate.

Over HTTPS connections, third-party MITM attacks are not possible because the connection is encrypted end-to-end. Thus, it is possible to track that you have accessed a particular site, but not what you have accessed on that site.



Watch out for cookies!



A cookie is a simple string of text loaded into users' browsers when they visit a site. Its goal is to enable the site to recognize and remember its users. But cookies make up the majority of online website crawlers.

The cookie was invented in 1994 by Netscape's Lou Montulli and John Giannandrea, and originally served to provide websites with a "memory" so that they could, for example, store items in a shopping cart while the user searched for the goods on the site.

Although the cookie still serves this purpose, it can also monitor users and provide a lot of information about user behavior.

The cookie is widely used for profiling and targeted marketing, and most websites set a large number of both their own and third party cookies.

There are also many different cookies: required cookies, analytics or statistics cookies, marketing cookies, and advertising cookies. Strictly necessary cookies work so that a site operates its most basic functions, so that a visitor can visit it. These rarely, or never have any way of tracking users.

However, analytics or statistics cookies are usually third-party cookies that track and record user behavior to provide information to the site owner. Marketing and advertising cookies are also often third party cookies that serve to make targeted advertising possible. These cookies are site tracking tools for the companies involved: The company itself, the site owner, and companies that use cookies for tracking and profiling (Google and all others in the ad technology industry: Yahoo, Bing, Facebook , etc ...)

Types of Cookies

As we saw above, there are "normal" cookies, but there are still two other types of cookies: Supercookies and Evercookies.

Supercookies contain a unique identifier that allows trackers to link records in their data to track their history and browsing behavior (for example, websites visited including length of stay). An example is Flash cookies (also known as Local Sharing Objects or LSO) which, unlike standard cookies, work independently of the browser and have no expiration date. They are stored locally and can be removed manually. To prevent them from being stored on your device, you must set the Flash plugin in your browser to not load flash objects or only with your consent. To do this, select the "Never Activate" or "Prompt to Activate" options.

Evercookies are extremely persistent and difficult to get rid of when they get into the device. Its purpose is to identify a user even after removing standard cookies, Flash and others. Evercookies use different storage mechanisms to store data in different formats at various locations on your device. If cookie data is removed, it is immediately recovered from an alternate storage location.

Countermeasures

To protect yourself from spy cookies, there are some precautions you can take.

- Use a cookie tracking extension: Cookie AutoDelete (Firefox, Google Chrome), Cookie Master (Palemoon), Crush Those Cookies (Palemoon), Cookies Control Panel (Palemoon), Cookies Exterminator (Palemoon, Seamonkey or Firefox), Edit this cookie (Google Chrome), Vanilla Cookie Manager (Google Chrome).
- Manually delete all browser cookies after exiting. Or check in the settings: Delete all cookies on logout.
- Create a new browser profile (Firefox or Google Chrome) with each new session. This prevents cookies from being persistent as each new session is as if a new user were logged on to the computer.

However, over time, user tracking techniques have evolved to become more elaborate and complex.

Tracking users without cookies

Tracking pixels, also called 1x1 pixel or pixel tags, are transparent images that consist of a single pixel, present (though virtually invisible) on a web page or in an email.

When a user loads the webpage or opens an email, the tracking pixel is also loaded, allowing the sender of the tracking pixel, usually an ad server, to read and record that the webpage is loaded or the email is open and similar user activities.

The purpose is the same as for third party cookies: to get user information for targeted marketing (or whatever purposes)

Information that may be obtained by websites and third parties through tracking pixels includes:

- which OS the user uses
- which browser or email program the user uses
- when the site was visited or the email was read
- user behavior on the site visited
- IP Address and User

Location Tracking pixels are a widely used form of analytical or statistical tracking, but the General Data Protection Regulation (GDPR) considers it illegal if it is not first consented to by the user.

Tracking with Javascript

Tracking scripts are parts of Javascript code that usually implement a tracking pixel on a site and are responsible for creating different request types for external domains, passing the data to them.

The tracking script can access and create requests using the data available on the page, as well as set different cookies that can be used as identifiers. This commonly transmitted data consists of data available in HTML, a URL, a data layer, cookies, or obtained through an event listener or an API call. The tracking script can make different types of requests to pass this information to the tracking pixel.

An example of tracking javascript is "analytics.js" from Google.

Countermeasures

To avoid problems with trackers without cookies, either transparent gifs or scripts, the following extensions can be installed:

- No Script

The NoScript extension provides extra protection for Firefox, Seamonkey, and other browsers (Google Chrome): This free, open source add-on allows JavaScript, Java, Flash, and other plugins to be run only by trusted websites of your choice (eg. your bank's website).

NoScript also provides the most powerful anti-XSS and anti-Clickjacking protection ever available in a browser.

NoScript's unique whitelisting-based preventive script blocking approach avoids exploiting security vulnerabilities (known as Meltdown or Specter and even not yet known!) without loss of functionality.

You can enable JavaScript, Java, and plug-in execution for trusted sites by simply left-clicking on the NoScript status bar icon or by using the contextual menu for easy operation in pop-up windows without a status bar.

• AdBlock Plus

Adblock Plus is an ad filtering extension for Firefox, Chrome, Opera, Microsoft Edge, Internet Explorer, Yandex and Maxthon.

This program was included in the list of Firefox add-ons as of January 17, 2006. Due to its huge usage, around 80,000 downloads per day have been reported, reaching 100 million in total.

Like Mozilla's image blocker, Adblock blocks HTTP requests according to their source addresses and can block IFrames, scripts and Flash. It also uses automatically generated user stylesheets (CSS) to hide, block, elements such as text-based applications on a page as they load.

• ScriptBlock

A smart extension that controls javascript, iframes, and plugins in Google Chrome.

Scriptblock is an extension that provides better control of javascript, iframes and other unwanted content.

It can even mitigate cross site scripting (XSS) and drive-by-downloads attacks.

ScriptBlock works perfectly in conjunction with other extensions like AdBlock, AdBlock Plus or Ghostery.

You can add trusted sites to a whitelist so that they

are not affected by ScriptBlock. Available for Palemoon and Google Chrome.

Fingerprinting

Fingerprinting is an advanced browser identification technique that the user is accessing in order to uniquely and accurately identify the user.

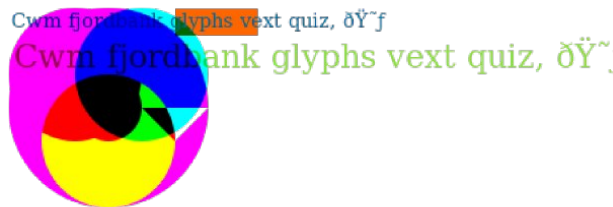
There are several Fingerprinting techniques, and let's illustrate some of them. In fact, the sophistication of Fingerprinting is such that it seems to come out of a spy movie.

Basically, there are 4 types of Fingerprinting:

- Canvas Fingerprinting
- AudioContext Fingerprinting
- WebRTC Local IP Discovery
- Canvas-Font Fingerprinting

Canvas Fingerprinting

HTML Canvas allows the web application to draw real-time graphics, with functions to support drawing shapes, arcs, and text in one Custom canvas element. Differences in font rendering, anti-aliasing, and other device features make devices draw the image differently. This allows the resulting pixels to be used as part of the device fingerprint. The image below is a representative example of the types of screenshots used by fingerprinting scripts.

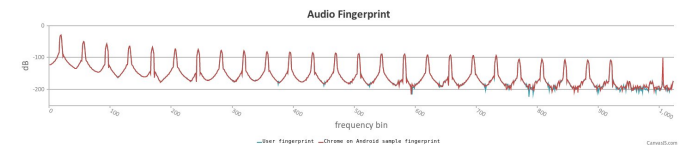


AudioContext Fingerprinting

Fingerprinting techniques are not usually used in isolation, but together. One of the new fingerprinting techniques accesses the AudioContext interface and related interfaces. Trackers try to use the audio API to fingerprint users in various ways.

How it works? An audio signal is generated by an oscillator and the resulting signal is split after processing to create an identifier. This does not require access to the device microphone and depends on differences in the way the generated signal is processed.

In the figure below, the fingerprint audio of my notebook (where I wrote this article) compared to the fingerprint audio of a standard Android reference device.



Also, with this technique, it was possible for the script to profile my hardware admirably:

AudioContext Fingerprints

AudioContext properties:

```
{
  "ac-sampleRate": 48000,
  "ac-state": "suspended",
  "ac-maxChannelCount": 0,
  "ac-numberOfInputs": 1,
  "ac-numberOfOutputs": 0,
  "ac-channelCount": 2,
  "ac-channelCountMode": "explicit",
  "ac-channelInterpretation": "speakers",
  "an-fftSize": 2048,
  "an-frequencyBinCount": 1024,
```

```
"an-minDecibels": -100,
"an-maxDecibels": -30,
"an-smoothingTimeConstant": 0.8,
"an-numberOfInputs": 1,
"an-numberOfOutputs": 1,
"an-channelCount": 1,
"an-channelCountMode": "max",
"an-channelInterpretation": "speakers"
}
```

Fingerprint using DynamicsCompressor:

```
35.73833402246237
```

Fingerprint using DynamicsCompressor (hash of full buffer):

```
5f5ca60e8aa59e3b0f1f5d3064746a42b760ed57
```

Fingerprint using OscillatorNode:

```
-122.10106658935547, -122.17668151855469,
-121.51886749267578, -120.64137268066406,
-119.46350860595703, -118.02296447753906,
-116.34650421142578, -1  14.44546508789062,
-112.31880950927734, -109.9481201171875,
-107.31201171875, -104.37184143066406,
-101.08609008789062, -97.42935943603516,
-93.51634979248047, -90.6439208984375,
-82.70191192626953, -44.70573425292969,
-31.952308654785156, -29.518218994140625,
-36.196712493896484, -56.003013610839844,
-95.44043731689453, -91.57635498046875,
-95.32821655273438, -99.18425750732422,
-102.70282745361328, -105.87161254882812,
-108.73149108886719, -111.33251190185547
...
```

Fingerprint using hybrid of OscillatorNode / DynamicsCompressor method:

```
-129.05262756347656, -119.44970703125,
-108.23188018798828, -101.38246154785156,
-102.750732421875, -114.5762939453125,
-109.89714813232422, -98.22866821289062,
```

```
-95.67564392089844, -103.14139556884766,
-117.47007751464844, -111.52037048339844,
-101.82388305664062, -101.04195404052734,
-89.28307342529297, -93.03608703613281,
-89.74243927001953, -53.4901237487793,
-40.75892639160156, -38.32568359375,
-45.00003433227539, -64.69832611083984,
-95.99256896972656, -88.3079833984375,
-94.27742004394531, -98.67274475097656,
-107.62669372558594, -114.796630859375,
-106.75559997558594, -95.29183959960938
...
```

WebRTC Local IP Discovery

WebRTC is a framework for real-time, point-to-point communication in the browser and accessible via Javascript. To find the best path between peers, each pair collects all available candidate addresses, including addresses from local network interfaces (such as Ethernet or WiFi) and public-side addresses from NAT, and makes them available to the Web application without explicit permission. of user. A fingerprinter script can leverage these addresses to track users.

Canvas-Font Fingerprinting

Javascript and Flash were used to enumerate fonts in the browser and also to profile users.

The HTML Canvas API provides a third method for deducing fonts installed in a specific browser. The screen rendering interface exposes a `measureText` method, which provides the resulting width of the text drawn on the screen. A script can try to draw text using a large number of fonts and then measure the resulting width. If the text width is not equal to the text width using a default font (which would indicate that the browser does not have the tested font), the script may conclude that the browser has that font available.

As an example, here is the data that a test script extracted from my notebook:

Canvas, Javascript, and Flash Font Detection

Canvas font detection:

Andale Mono; Arial; Arial Black; Bitstream Vera Sans Mono; Calibri; Cambria; Cambria Math; Comic Sans Courier; Courier New; Georgia; Helvetica; Impact; Lucida Console; Lucida Sans Unicode; Microsoft Sans Serif; Palatine; Segoe UI; Tahoma; Times; Times New Roman; Trebuchet MS; Verdana; Wingdings;

JS / CSS font detection:

Andale Mono; Arial; Arial Black; Bitstream Vera Sans Mono; Calibri; Cambria; Cambria Math; Comic Sans MS; Courier; New Courier; Georgia; Helvetica; Impact; Lucida Console; Lucida Sans Unicode; Microsoft Sans Serif; Palatine; Segoe UI; Tahoma; Times; Times New Roman; Trebuchet MS; Verdana; Wingdings;

Flash font detection:

AD MONO, Adventure, Andale Mono, Arial, Arial Black, Babelfish, Bitstream Charter, Bitstream Vera Sans, Bitstream Vera Sans Mono, Bitstream Vera Serif, Blue Highway, Condensed Blue Highway, Calibri, Cambria, Cambria Math, Cantarell, Cantarell Extra Bold, Cantarell Light, Cantarell Thin, Christmas School, Comic Sans MS, Courier New, DejaVu Math TeX Gyre, DejaVu Sans Condensed, DejaVu Sans Light, DejaVu Sans Mono, DejaVu Serif, DejaVu Serif Condensed, Deutsch Gothic, Dingbats, DirtyBaker'sDozen, Georgia, Hershey-Gothic-English, Hershey-Gothic-German, Hershey-Plain-Duplex, Hershey-Plain-Duplex-Italic, Hershey-Plain-Triplex, Hershey-Plain-Triplex-Italic, Hershey-Script-Complex, Hershey-Script-Simplex, Impact, Larabiefont, Liberation Mono, Liberation Serif, Lucida Console, Lucida Sans Unicode, Microsoft Sans Serif, New, Nimbus Mono L, Nimbus Roman No9 L, Nimbus Sans L, Nimbus Sans L Condensed, Segoe UI Light, Segoe UI Light, Standard Symbols L, Symbol, Tahoma, Times New Roman, Trebuchet

MS, URW Bookman L, URW Chancery L, URW Gothic L, URW Palladio L, Utopia, Verdana, Webdings, Wingdings

Countermeasures

To try to “fool” fingerprinting, you can install some extensions in your browsers:

Secret Agent (Palemoon, Waterfox, Seamonkey, Icecat) enhances the privacy of your web browsing by rotating the identity of the browser 'User Agent' with all web requests (or each page load or session of each browser).

Randomizing your User Agent makes it a little harder for bad guys, rogue ISPs, spies like Phorm, corrupt governments, and other nasty surveillance / tracking threats to correlate your clicks based on device fingerprinting.

Ghostery is a free, open source privacy and security related browser extension, plus a mobile browser application. Since February 2017, it belongs to the German company Cliqz International GmbH (formerly owned by Evidon, Inc., formerly Ghostery, Inc. and the Better Advertising Project). The code was originally developed by David Cancel and associates.

Ghostery allows its users to easily detect and control JavaScript tags and trackers. JavaScript bugs and beacons are embedded in many web pages, largely invisible to the user, allowing the collection of user browsing habits through HTTP cookies, as well as participating in more sophisticated forms of tracking, such as canvas fingerprinting.

As of 2017, Ghostery is available for Mozilla Firefox, Google Chrome, Internet Explorer, Microsoft Edge, Opera, Safari, iOS, Android, and Firefox for Android (not available for Palemoon, Seamonkey, Icecat).

Use And Abuse Private Browsing

A Private browsing automatically deletes browsing information such as passwords, cookies and history, leaving no trace after you log out.

Private browsing is available on all popular browsers (Firefox, Chrome and all derivatives thereof)

TOR and VPN

Using TOR browser can get you in trouble:

The U.S. Supreme Court quietly approved a rule change that would allow a federal magistrate judge to issue a search and seizure warrant for any target using anonymity software like Tor to browse the internet.

As for Virtual Private Network (VPN) networks, they're as good as the companies behind them. That is, as long as these companies don't hand over your browsing history and data to government agencies, you're fine.

But otherwise...

I think that both the TOR browser and virtual private networks deserve their own articles, where they can be further analyzed.

Final Words

Over the course of eight months, we thoroughly and thoroughly analyzed the ways Google uses to spy on its users.

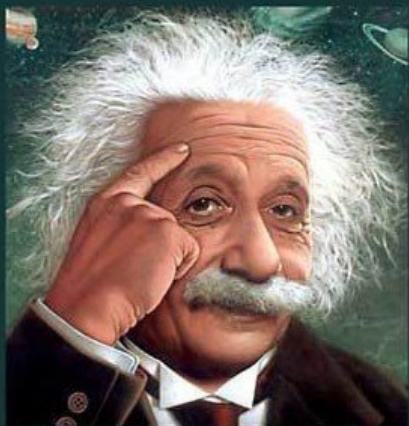
We've looked at the known ways, and many new online tracking technologies, that in these perilous times we live in, where freedom of thought and expression can put the average citizen in hot waters, given the prevailing polarization and political radicalization.

Protecting privacy, freedom of thought and expression has become a necessity these days, but mostly staying out of trouble, controversy and online discussion that can hurt ordinary people (as Richard Stallman was harmed by giving his opinion on a very sensitive subject).

So take a good look at tips for protecting your online privacy, and especially don't get involved in controversy or sensitive issues, whether it's on Twitter, Facebook, YouTube or whatever, as out-of-context statements can cost careers, reputations and even jobs.

A big hug, and see you next month.

DESTINATION LINUX
LINUX IS OUR PASSION



PCLinuxOS
Radically Simple

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 ...

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.



Screenshot Showcase

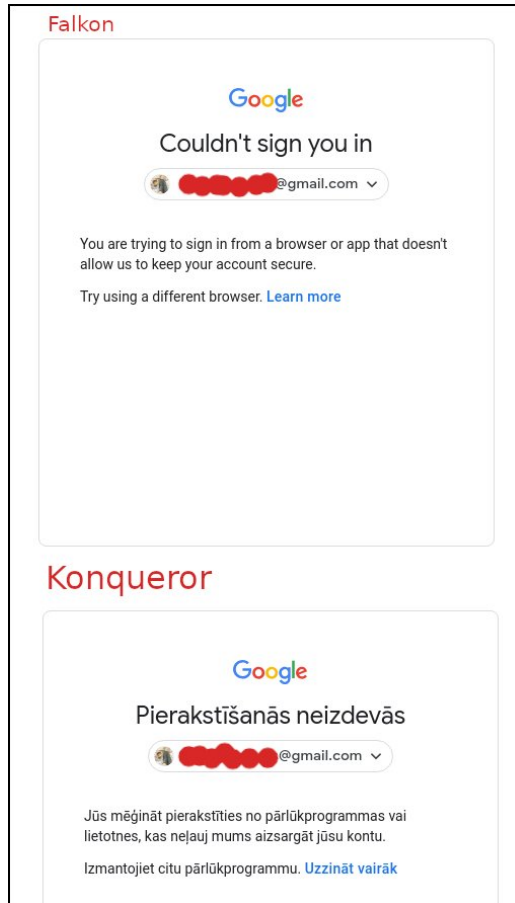


Posted by francesco_bat, on December 11, 2019, running Trinity.

Short Topix: Mozilla Removes Avast, AVG Firefox Extensions Over Snooping Claims

by Paul Arnote (parnote)

Google Now Bans Some Linux Web Browsers



In its “infinite” wisdom, Google has now blocked certain Linux web browsers from accessing its services, according to an [article](#) on BleepingComputer. These Linux web browsers include Falkon, Konqueror and QuteBrowser. All but QuteBrowser are in the PCLinuxOS repository. Google asserts that they “may not be secure” ... as if logging into any Google service is like logging into some kind of fortress of security and privacy.

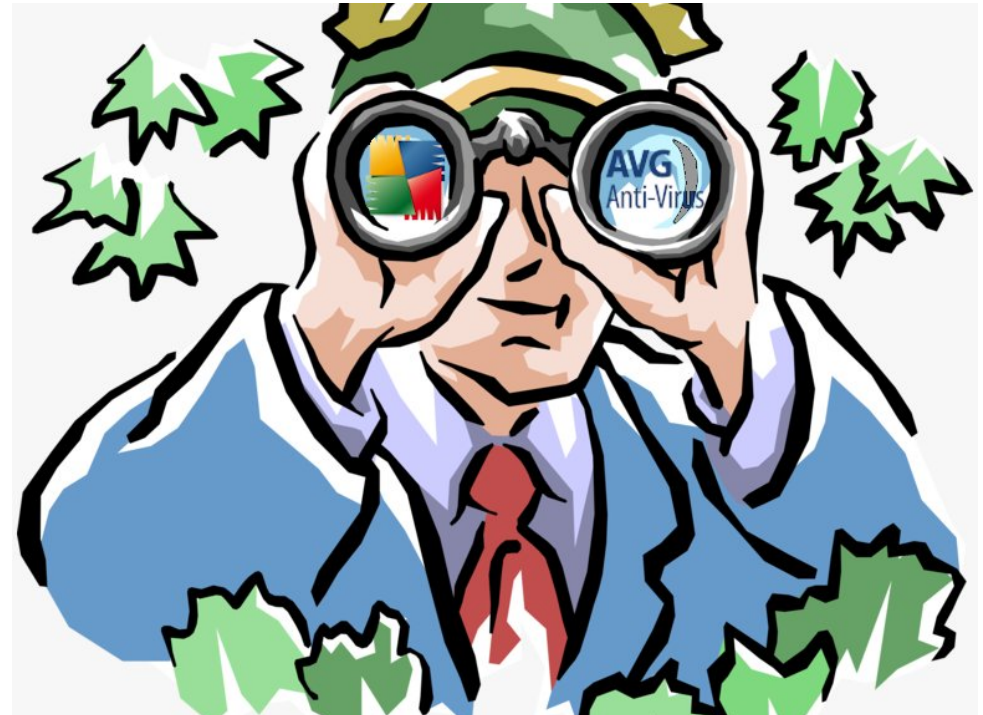
The issue was reported on [Reddit](#) by u/onedoer. Ironically, there are also several replies in the Reddit thread of other users who have had no problem using the aforementioned Linux web browsers.

BleepingComputers, in their independent tests, were able to confirm – on multiple machines – the inability to log into Google services on both Falkon and Konqueror. Clicking on the “Learn More” link reveals a number of

reasons that Google provides for the possible block. They could include lack of support for JavaScript or having JavaScript support turned off, having insecure or unsupported extensions installed, the use of automated testing frameworks, or the browser is embedded in a different application.

There is only speculation about why the web browsers work for some users, but not others. Google has not yet responded to a query from BleepingComputers with an answer of any kind. Imagine that.

Mozilla Removes All Avast, AVG Firefox Extensions Over Snooping Claims



Do you remember the free antivirus package AVG Free by Avast from back in your Windows days? Or, maybe you are aware of their current “antivirus” offerings for Linux. Well, for the past couple of years, Avast has made available four addons for Firefox. They are Avast Online Security, AVG Online Security, Avast SafePrice, and AVG SafePrice.

In late October, Wladimir Palant, the creator of Adblock Plus discovered that these addons were collecting WAY, WAY, WAY more information than was

Short Topix: Mozilla Removes Avast, AVG Firefox Extensions Over Snooping Claims

necessary to perform their duties. Take a look at the information collected, as pointed out from Palant's [blog](#):

The interesting fields were:

uri: The full address of the page you are on.

title: Page title if available.

referrer: Address of the page that you got here from, if any.

windowNum and tabNum: Identifier of the window and tab that the page loaded into.

Initiating_user_action windowEvent: How exactly you got to the page, e.g. by entering the address directly, using a bookmark or clicking a link.

visited: Whether you visited this page before.

locale: Your country code, which seems to be guessed from the browser locale. This will be "US" for US English.

userid: A unique user identifier generated by the extension.

plugin_guid: Seems to be another unique user identifier.

browserType and browserVersion: Type (e.g. Chrome or Firefox) and version number of your browser.

os and osVersion: Your operating system and exact version number (the latter only known to the extension if Avast Antivirus is installed).

All of this information, and more, was transmitted to Avast and AVG servers. According to a gHacks [article](#), data was sent whenever a user opened a page or switched tabs, as well as every single click on a search page.

If the above revelation didn't unnerve you, take a look at this assertion on Palant's blog page:

The data collected here goes far beyond merely exposing the sites that you visit and your search history. Tracking tab and window identifiers as well as your actions allows Avast to create a nearly precise reconstruction of your browsing behavior: how many tabs you have open, what websites do you visit and when, how much time do you spend reading/watching the contents, what do you click there and when do you switch to

another tab. All that is connected to a number of attributes allowing Avast to recognize you reliably, even a unique user identifier.

Palant reported the excessive data collection to Mozilla, and received a polite "thank you" for his notification. Then, in less than 24 hours, Mozilla quietly removed the Avast and AVG addons from the Mozilla addon repository, stopping short of putting them on the blacklist (which would have inactivated/removed every installed copy on users' machines).

In response, Avast provided the following statement:

We have offered our Avast Online Security and SafePrice browser extensions for many years through the Mozilla store. Mozilla has recently updated its store policy and we are liaising with them in order to make the necessary adjustments to our extensions to align with new requirements. The Avast Online Security extension is a security tool that protects users online, including from infected websites and phishing attacks. It is necessary for this service to collect the URL history to deliver its expected functionality. Avast does this without collecting or storing a user's identification.

We have already implemented some of Mozilla's new requirements and will release further updated versions that are fully compliant and transparent per the new requirements. These will be available as usual in the Mozilla store in the near future.

Avast claims, via a BleepingComputer [article](#), that it had less than 12 hours to resolve the issue.

As of the writing of this article (late December), the issues appear to have been resolved, and the Avast and AVG addons are once again available in the Mozilla addon repository. Avast provided the following statement to gHacks:

"Privacy is our top priority and the discussion about what is best practice in dealing with data is an ongoing one in the tech industry. We have never compromised on the security or privacy of personal data. We are listening to our users and acknowledge that we need to be more transparent with our users about what data is necessary for our security products to work, and to give them a choice in whether they wish to share their data further and for what purpose. We made changes to our extensions including limiting the use of data and these changes are explained clearly in our Privacy Policy. Our browser extensions Avast Online Security and AVG Online Security are back on the Chrome Store, and on the Mozilla Store (since 12/17). It's important to us that users understand that we're listening to concerns about transparency and data use, and striving to do better and lead by example in this area."

Through at least the beginning of this "event," the Avast and AVG extensions remained available through the Google Play Store. Avast's response seems to

Short Topix: Mozilla Removes Avast, AVG Firefox Extensions Over Snooping Claims

indicate that at some point, the extensions were also removed from the Google Play Store.

Firefox Launches Firefox Private Network Browsing



(From the Mozilla [blog](#)) In September, we (Mozilla) [introduced the beta release of our Firefox Private Network \(FPN\)](#), an extension which provides a secure, encrypted path to the web to protect your connection and personal information when you use the Firefox browser. Since then, we've received feedback from our beta testers on how they're using FPN, its protections, and we learned about websites that weren't compatible as well as connection issues. This allowed us to quickly identify and fix bugs, and ensure a stable product.

As we continue our beta testing, we are considering various ways to bring additional privacy protections to our users. Today we're announcing an [additional beta test for US-based Firefox account users](#) who didn't get a chance to get in the initial group, and are interested in testing FPN.

In the next phase of our beta, we are [offering a limited-time free service](#) that lets you encrypt your Firefox connections for up to 12 hours a month. With the holidays around the corner, the FPN couldn't come at a more convenient time. We know people are traveling and might have to rely on an unsecured public Wi-Fi network, like the one at the airport, at your local coffee shop, or even at your doctor's office. FPN provides encrypted internet traffic thus giving you peace of mind whenever you're using our browser.

This limited-time free service is currently [available in the US](#) on the Firefox desktop browser and you'll need a [Firefox account](#) to try the service. You can sign up directly from the extension which can be found [here](#).

For those looking to extend their protection beyond the browser, you can now sign up to be one of the first to experience the newest member of the FPN family. This month, [Firefox account holders](#) can request invitations to experience device-level protection with our new full-device VPN (virtual private network). [Join the waitlist](#) and if you're eligible, we'll follow up with a link to access the VPN at an introductory price of \$4.99 per month. Currently the VPN will be available for Windows 10 only, and like the rest of the FPN, it is only available to US-based Firefox account holders. Pricing and platform availability will continue to evolve and we look forward to hearing your feedback.

Firefox Says NO To Fingerprinting

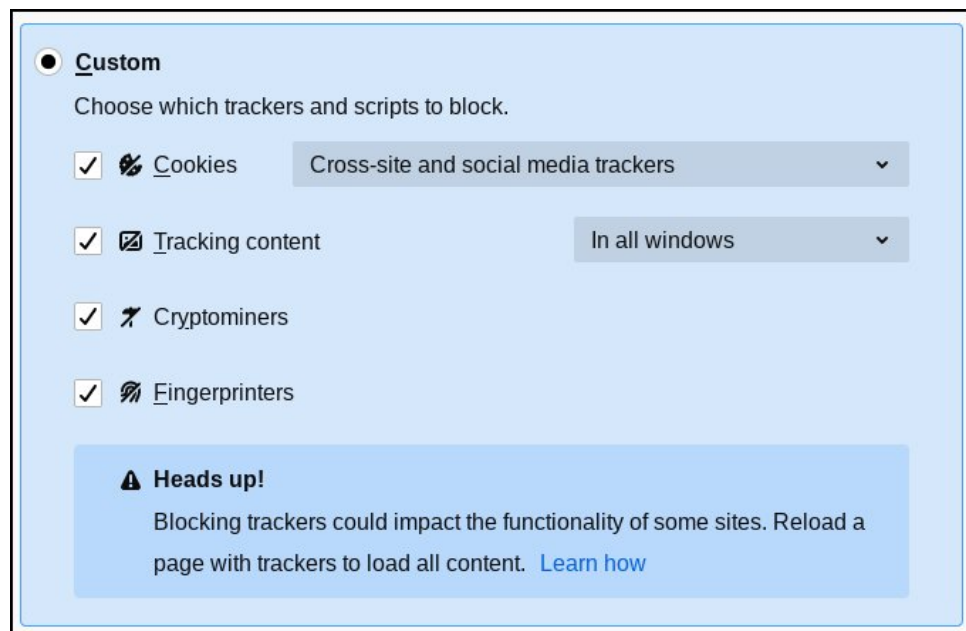


One of the topics at this past summer's Google IO developer conference was how to help preserve user privacy by blocking "fingerprinting." Well, Firefox just did an end run around Google, and has blocked fingerprinting. So exactly what is fingerprinting?

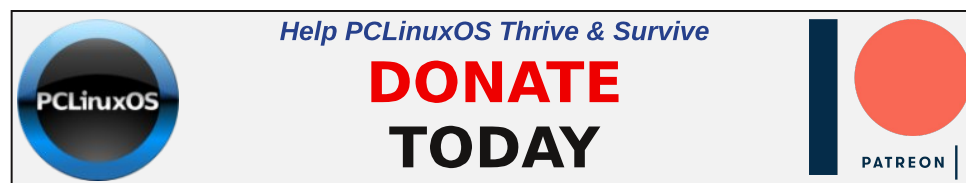
Virtually everyone knows what cookies are when it comes to web browsing. Cookies are easy to block, and many sites give you the option to opt out of them. But fingerprinting is different. It collects basic information from your browser session and your computer, using it to create a unique "fingerprint" of you. Included may be the make and model of your computer, how much RAM you have, processor identification, web browser, screen resolution, your timezone, the language you use, the combination of third-party extensions you have installed,

Short Topix: Mozilla Removes Avast, AVG Firefox Extensions Over Snooping Claims

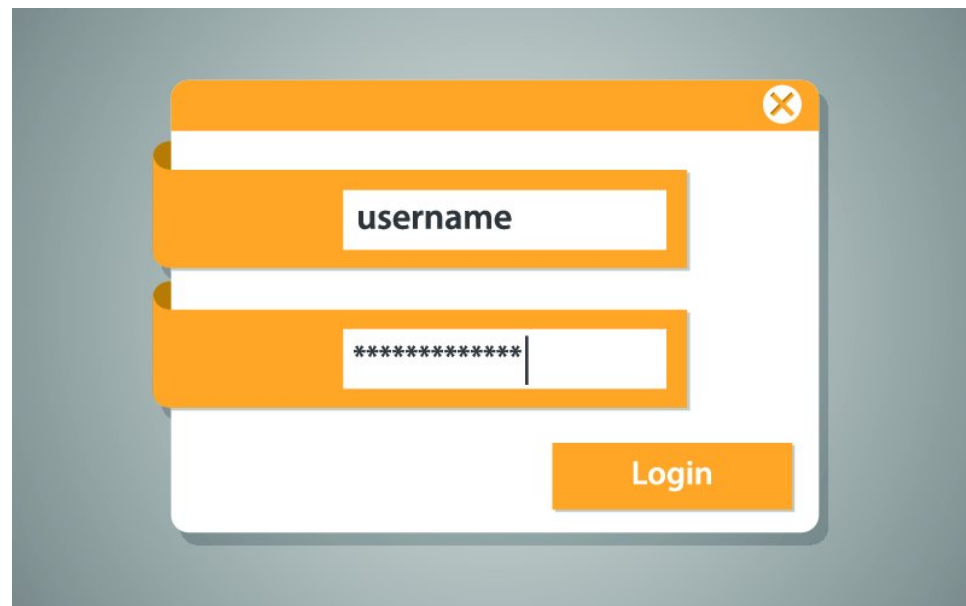
and even the fonts installed on your system. From all of that, it's rather easy to create a very specific and unique profile that can be used to track you across the web. Unfortunately, it happens without most users even realizing that it's happening, and there are mostly no mechanisms to allow a user to opt out of fingerprinting.



Starting with Firefox 72, due out later this month, Firefox will block fingerprint trackers by default. But, you don't have to wait until Firefox 72 to block fingerprint trackers. You can enable it now. In Firefox versions as early as 70, you can turn on protection against fingerprint trackers. Go to Preferences, then Privacy & Security. Under the "Enhanced Tracking Protection" section, select either "Strict" or the "Custom" setting. Under the "Custom" setting, you can choose to allow or disallow certain types of trackers. Strict just blocks them all. Even though turning on all the tracker and script blockers is basically the same as just selecting the Strict setting, I prefer to have the option of turning things off and on as I see fit. Also, the custom setting gives me a little more control of how strict I want to get about blocking trackers and scripts.



Make Sure Your Password Isn't Among The 100 Worst Passwords Of 2019



SplashData has released their list of the top 100 worst passwords of 2019. The battle for the absolute worst password wasn't really a battle or competition – it was the same as the worst password of 2018: 123456. Some variation of the worst password (123456) occupies six of the ten worst passwords in the list. You can view the worst passwords, ranked 50 through 100, [here](#). To view the top 49 worst passwords, go [here](#).

Over and over, users don't seem to be getting the message about using strong passwords to protect their data and privacy. Here at The PCLinuxOS Magazine, we've run numerous articles about passwords, password security, and methods to employ in creating secure passwords.

To summarize, be sure to follow a few simple rules for the creation and use of passwords. First, don't reuse passwords between sites and programs that ask for one. Use ONLY passwords that are unique to each site. Second, don't use birthdays, street addresses, and names of your spouse, children and pets ... basically, anything that is easily associated with you. Third, use a method of generating passwords that is known only to you. This means that you can use a "base" password that is tailored/alterd for each site you need a password for, as an example. Fourth, mix uppercase letters, lowercase letters, numbers and punctuation marks in your password creation process. It makes it much more difficult for hackers to break your password.

Short Topix: Mozilla Removes Avast, AVG Firefox Extensions Over Snooping Claims

It's YOUR data. It's YOUR privacy. No one else is going to bother to protect it for you, so that is your responsibility, and yours alone. The temptation to gain access to your privacy and data is too tempting to allow someone else to be responsible for. Don't make the mistake of underestimating the value of your data and privacy.

PCLinuxOS Roundup



AT THE BEGINNING OF 2019, Google announced their Manifest V3 plans, which ultimately broke several ad blocker extensions, and made them a lot less effective. Now, users are raging against Google for their web API called `getInstalledRelatedApps`. It has been in development since 2015, and has been available to experiment with since Chrome 59 came out in 2017. Basically, the API lets developers determine if their native app is installed on your device. The outrage comes from the potential to violate users' security and privacy, according to an [article](#) on Forbes. The potential for abuse is ripe, and given Google's oversight (or lack thereof) of the Google Play Store, users have every expectation that Google's oversight of the use of this API is most likely going to be sorely lacking.

CYBERSECURITY ANALYSTS AT THE NEWLY FORMED SENTINEL LABS claim to have found some of the first hard links between the crimeware organization TrickBot and Lazarus Group, the cyberwarfare division of the North

Korean military's Reconnaissance General Bureau, according to an [article](#) on TechRepublic.

ACCORDING TO AN [ARTICLE ON DECRYPT](#), hackers have a new way to surreptitiously mine Monero cryptocurrency on your computer. They will drop malware somewhere on your computer, where it remains unknown to the user. The malware itself doesn't contain malicious code, but when activated by a series of commands, it will start using the infected user's computer to mine cryptocurrency. The code lies in an inactive state, remaining undetected, until it is activated by the series of commands. It then sends the mined Monero to a wallet controlled by the hacker.

HOW DO YOU FEEL ABOUT YOUR SMARTPHONE TRACKING YOU everywhere you go, and monitoring everything you do? If you think it isn't a big deal, and you're willing to "live with it" as it currently exists just for the convenience that your smartphone provides, you need to read this introductory opinion [article](#) from the New York Times. It's worse than you think, and they provide ample evidence about just how bad the situation really is. This article will likely scare the bejeezus out of you, as it should. This is the inaugural article of a seven article series that is, to say the least, eye opening.



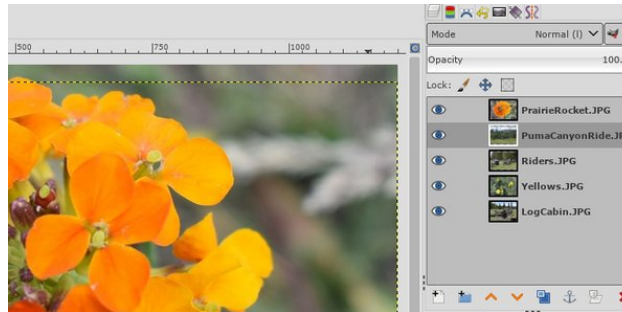
GIMP Tutorial: Photo Filmstrip

by Meemaw

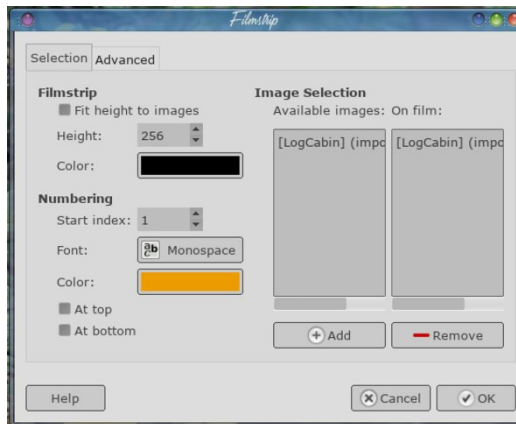
Last month, we saw a Repo Review of the program PhotoFilmStrip, which allows you to create a slideshow from your chosen photos. However, if you want a single image that includes several photos, you can create something like the following in GIMP. I've seen this done by others but we've never covered it in these tutorials, so I thought I'd do one. In this tutorial, we can make a filmstrip out of several photos we load, and use the Curve Bend filter to make it something other than a straight rectangular strip.



I used 5 photos from a recent trip to the mountains. You could use less or more, I'm sure. When you have your photos selected, open GIMP and choose **File > Open As Layers**. Your photos will open in one GIMP project, and you'll be able to see them in the layers window (center, top).

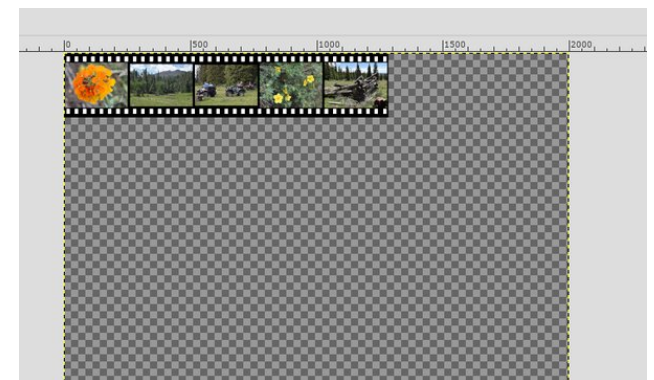
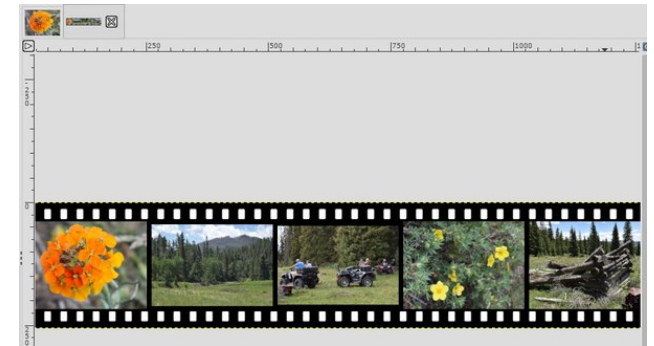


While you are here, you can reorder your photos into the order you want them by moving the layers. When you are satisfied, choose **Filters > Combine > Filmstrip**. In the window, you can keep the defaults and click **OK**. I unchecked the boxes dealing with showing the numbers on the filmstrip.



You'll get a new project with the filmstrip. If you want to, you can delete the first project with the photos opened as layers (right, top).

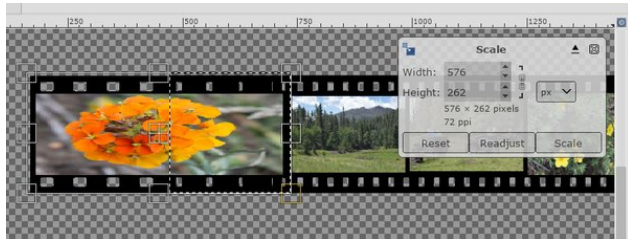
To give yourself room to work, add a transparent layer 2000px by 2000px. Then click **Image > Fit Canvas To Layers** (right).



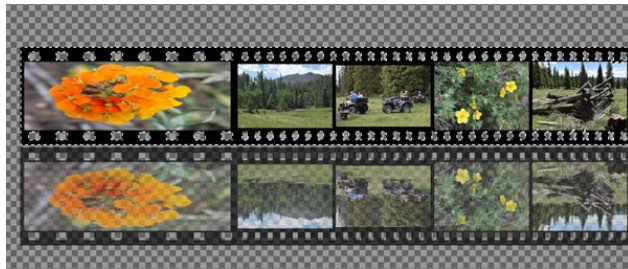
Choosing the layer with the photos, and using the **Move** tool, move the filmstrip to just above and to the right of the center of the canvas.

The openings in the filmstrip are still white and we want them transparent like the background. First, right click on the layer and choose **Add Alpha Channel**. Click on the **Select by Color** tool and then click on one of the white rectangles to select the color white. Now, choose the eraser tool. We're going to start our eraser on one end of the filmstrip and use that tip we learned a few issues ago. Click on the end rectangle, then holding down the **Shift** key, click on the rectangle at the other end to erase them all. Merge your layers.

We need to stretch out the first photo so it will look right when we bend our filmstrip. With the **Rectangle Select** tool, select the first photo, then choose the **Scale** tool, and move the border of your selection to the left so it's just over twice the length that it was, and click **Scale**.

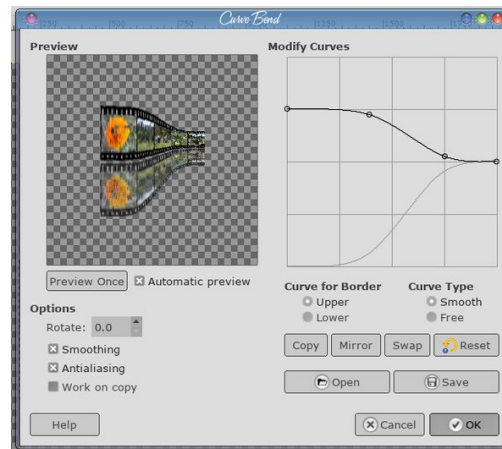
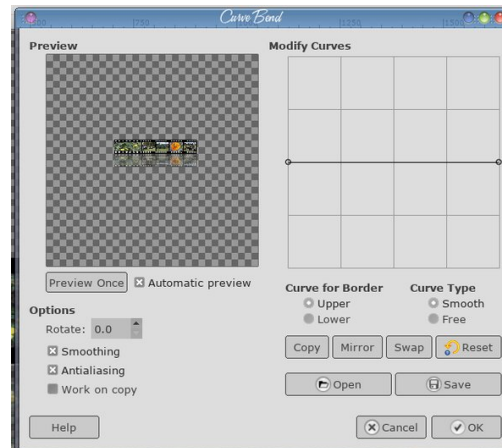


Make a copy of the filmstrip by clicking **<CTRL> + C**, then **<CTRL> + V**. The copy is over the original, of course, but in a floating layer. Flip the copy vertically, then move it beneath the original, and then set the opacity to about 40%.

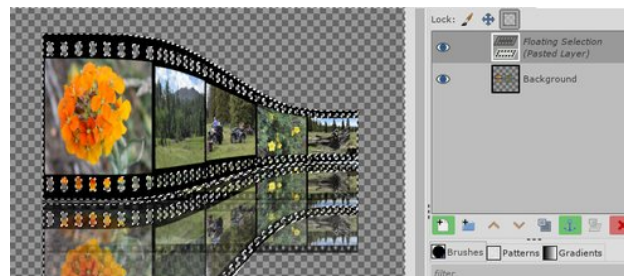


The copy is now on a floating layer, so you should anchor it.

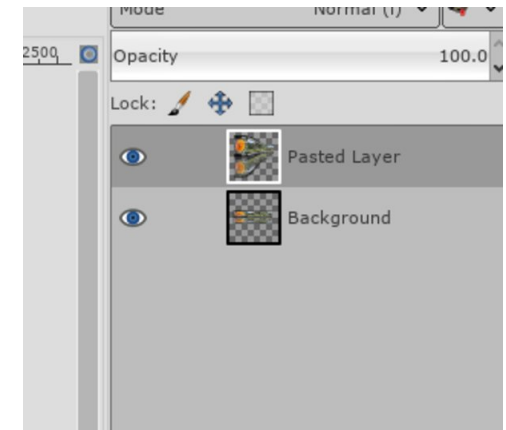
Using the rectangle select again, select both filmstrips, then click **Filters > Distorts > Curve Bend**. You'll get the first window below. In the center you'll see the setting "Curve for Border". Click on **Upper** first, and drag the left end of the line from center to the shape you see in the second window below. Then choose **Lower**, and drag the center line to the lower shape (center).



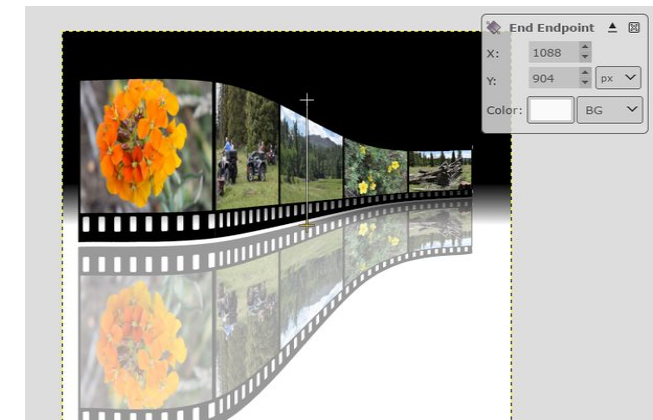
Click on **OK**. This filter takes a few seconds to work depending on your system. Notice that this is also put on a floating layer, and you can see the original strip below the curved strip.



Right click the floating layer and choose "To new layer". You can now delete the layer with the straight filmstrip.

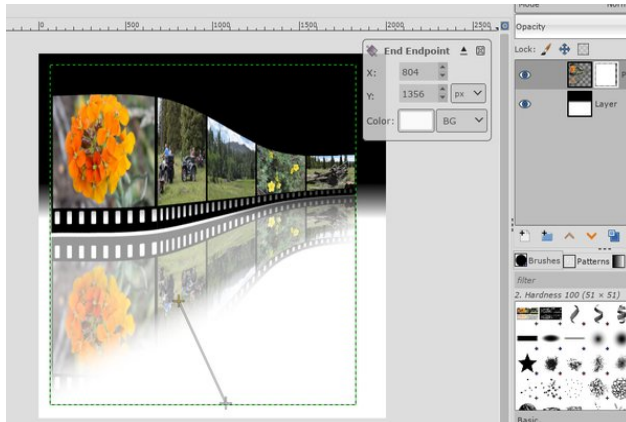


Let's give this a background. Create a new transparent layer and move it underneath your filmstrip. Choose the Gradient tool, FG to BG, and make your foreground black and your background white. Start at the top of the top strip and drag to the top of the bottom strip.



Let's also make the reflection fade out a bit like it should. Choose the filmstrip layer, right click it and click **Add Layer Mask**. In the Add Layer Mask window, make sure White is chosen. In the Layers toolbox, click on the layer mask next to the filmstrip

and using your gradient again, click from the bottom of the window a little diagonally and up into the strip a bit so the mask fades out the bottom part of the strip. How far you go is up to you.



Now you can merge your layers, and crop your creation, and you're finished!



Screenshot Showcase



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

Tip Top Tips: Move Firefox 71's Tab Bar BACK To The Bottom

Editor's Note: Tip Top Tips is a semi-monthly column in The PCLinuxOS Magazine. Periodically, we will feature – and possibly even expand upon – one tip from the PCLinuxOS forum. The magazine will not accept independent tip submissions specifically intended for inclusion in the Tip Top Tips column. Rather, if you have a tip, share it in the PCLinuxOS forum's "Tips & Tricks" section. Your tip just may be selected for publication in The PCLinuxOS Magazine.

It seems like about every six releases or so, Mozilla's Firefox developers change the format (or change something in the format) for the userChrome.css file that allows users to customize their Firefox interface. Then, Firefox users who have tweaked the Firefox interface to be ****just right**** for their use have to start all over again and try to figure out the new changes to get their customizations back. The last change to the userChrome.css occurred with Firefox 65.

Right on cue with Firefox 71, Mozilla's Firefox developers have done it again. Yep. They have altered SOMETHING in the userChrome.css file that renders previous userChrome.css file iterations ineffective and rather useless.

So, this month's **tip** comes from **Ramchu**, where he shows us how to move Firefox 71's tab bar BACK to the bottom-most spot on the Firefox toolbar, where many users prefer it to be, and where many users feel it should be. Color me (Paul Arnote, the magazine's chief editor) a believer of both. At LEAST give us a choice, amongst all the other choices users are presented with to easily make the desired change.

Ramchu's PCLinuxOS Forum post appears in its entirety below.



I just noticed that in Firefox 71 they have once again moved the TAB BAR to the top, and I prefer that it be below the address/URL bar and the bookmarks bar.

To get the tabs, once again, below the address bar and bookmarks bar, click this link, download and save the file:

https://www.userchrome.org/samples/userChrome-tabs_on_bottom_Fx70.css

Rename the saved file to **userChrome.css**.

Open your File Manager, and select the option to "Show Hidden Files."

Navigate to and open the hidden **.mozilla** folder.

Open the **firefox** folder.

Open the **default** folder.

Open the **chrome** folder. If the folder doesn't already exist, then create it. Be sure to name it **chrome**.

Rename the current **userChrome.css** file (if there is one), to something like **userChrome.css.old**.

Now **move** the **userChrome.css** file that you **downloaded** into the chrome folder.

Close and restart Firefox, and your tabs should now be on bottom.

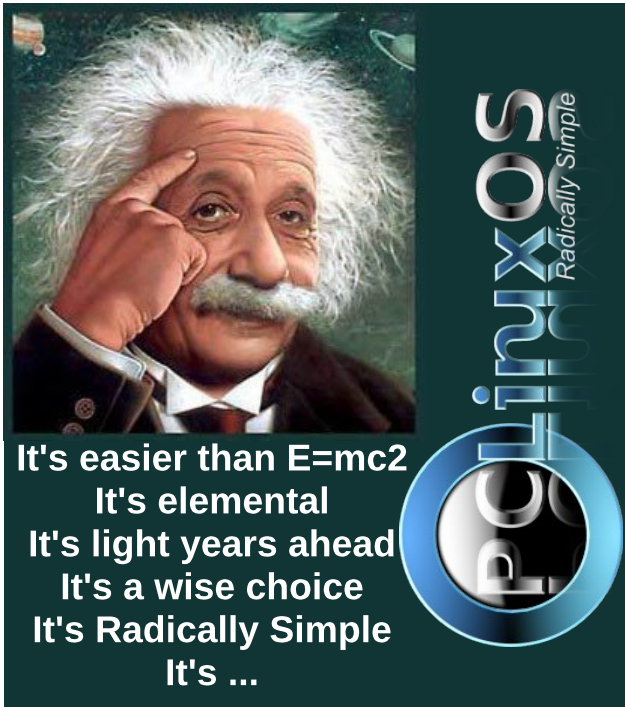
I got all of this from: <https://support.mozilla.org/en-US/kb/forum-response-tabs-below-url-bar>.

Summary

For the life of me, I don't know WHY Mozilla's Firefox developers will NOT put a simple checkbox amongst all the other various user configurable settings that will remove this perennial thorn from the sides of Firefox users who prefer/want/desire their tab bar to be the bottom-most item on the toolbar. Instead, users who prefer their tab bar on the bottom have to do this periodic ritualistic dance to return their tab bar to its rightful place each and every time the Firefox developers whimsically make changes to the userChrome.css file or format. Geez!

One checkbox. BAM! Done! It would be so freaking easy!

C'mon Mozilla!



PCLinuxOS
Radically Simple

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 ...

International Community PCLinuxOS Sites



Netherlands



Turkey



Denmark



Czechoslovakia



Italy



Polskie Centrum
Poland



FINLAND



Brazil

PCLinuxOS Users Don't

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

While Driving.

*Put Down Your
Phone & Arrive Alive.*

PCLinuxOS Family Member Spotlight: rgradle

As told to YouCanToo

What is your name/username?

My name is Rich Gradle, (rgradle on the forum).

Are you married, single?

My wife Lori and I have just celebrated our 36th wedding anniversary. She is from the Pittsburgh area, where we met. I always liked Pittsburgh, as the people are very friendly and some parts of the city are very scenic, especially during the winter. Pittsburgh is very hilly, so not a great place to drive around in the winter, but scenic nonetheless.



How about Kids, Grandkids (names and ages)?

We have one son, a daughter-in-law, 2 grand-dogs, 1 grand-cat, and 1 grand-python. No grand-children yet, but those things sometimes take a while.

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

My wife and I are both retired.

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

We currently call Raleigh, NC, our home. We have lived here for about 28 years. We like this area because we are close to the beach, close to the mountains, and close enough to Pittsburgh to drive up for a visit.



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

I have a Masters Degree in engineering from Texas A&M University. Most of my career was spent in new product development for a company that sells high pressure critical service equipment to power plants.



My work resulted in three patents, with two more in the pipeline.

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

Several years ago I bought a banjo so I could learn how to play bluegrass banjo music. After a number of years practicing, and much to my chagrin, I came to the realization that I have very little musical talent. I am still trying to learn, but with somewhat lowered expectations than when I first started.

Why and when did you start using Linux?

I have been using PCLinuxOS for over 10 years now. I normally build my own computers. I had just built a new one and was contemplating loading a new version of Windows, plus trying to decide on which virus and malware software I would need to get. I had been reading about Linux in general, and noted that Linux machines are generally not bothered by the viruses, etc., and are generally considered to be more secure. I decided I had had enough with all the Windows problems, and proceeded to download 3 Linux distros to try out, PCLinuxOS being one of them. Of the three I downloaded, PCLinuxOS was the only one that worked 100% out of the box – sound worked and wifi worked without having to do a lot of configuration first. Sold!!

What specific equipment do currently use with PCLinuxOS?

PCLinuxOS (KDE) is currently running on my desktop machine (ASUS m/b with AMD A10 processor), and on an HP laptop computer (AMD Phenom II/Mate). Performance on the desktop machine is great, but a bit slow on the laptop. I do some video editing for my church on the desktop machine using Kdenlive, a native Linux application that is very powerful. I also do some graphics development for the church using GIMP. Very powerful, but long learning curve with GIMP. Now I wish I had paid more attention to the GIMP articles that appeared in the PCLinuxOS magazine some time ago. I have a Windows 10 virtual machine on my desktop computer for a few applications that will not run under Linux. My wife, a Windows user from way back, was right at home on her KDE desktop in no time at all. When people try to tell me how complicated Linux is to use, I always bring up my wife's experience as an example of how easy Linux, and especially PCLinuxOS, is to use.

One of the things I always appreciate about PCLinuxOS is that the software is well thought out, meaning that the updates generally work well and without problem. This is really a nod to those to maintain the software in the repository. Thank you, thank you. Also, I always appreciate the help available on the forum. Even when I have made newbie errors, someone is always willing to provide direction to get me on the path forward. Just outstanding.

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

My extended family, including my son, mostly use MAC's, and they have no problem taking any opportunity to give me a ration of grief about not using a MAC and an iPhone. Using Linux just adds a little to the amount of grief I get, and a bit of head shaking. But, in the end, I just grin.

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

While I think that PCLinuxOS is the best Linux distro available today, I am somewhat perplexed that there are so many Linux distros. Seems to me that consolidation down to a very few distros might be helpful in getting applications to port to Linux, and getting hardware vendors to better support Linux.

Finally, thanks to Tex, et. al., for the great job they have been doing with PCLinuxOS.

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.



Like Us On Facebook!
The PCLinuxOS Magazine
PCLinuxOS Fan Club



Chat with PCLinuxOS users
from all over
the World.



**Sign up
TODAY!**

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



Support PCLinuxOS! Get Your Official

PCLinuxOS
Merchandise Today!

PCLinuxOS

IPTV: Free Streaming Over The 'Net

by Paul Arnote (parnote)

Today, people are cutting the cable TV cord like never before. There are several reasons.

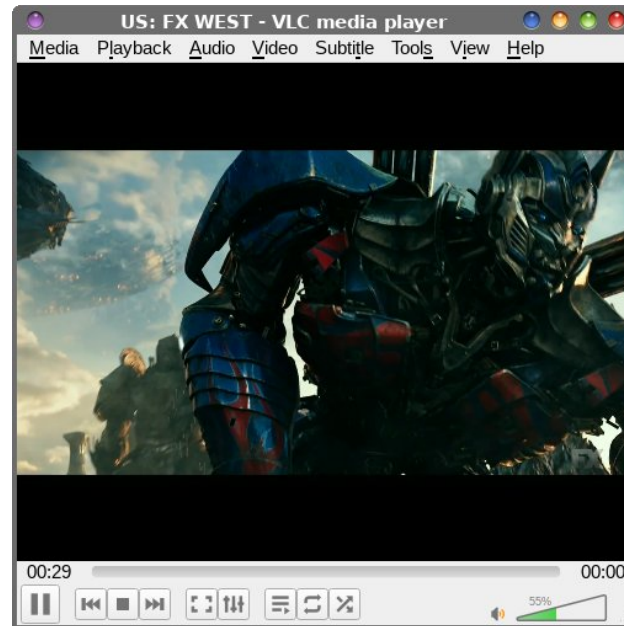
First, as prices rise, people are finding themselves less able to afford such a “luxury” as cable TV. Budgets tighten, and people’s pay doesn’t seem to go up nearly as fast as the cost of the bills. Something has to give. I’m firmly of the belief that one of the reasons for the cable TV prices going up and up and up is the increasing number of people cutting the cord. The upkeep costs on the distribution system remains relatively static, and with fewer people paying that monthly bill, those who remain with the cable TV provider have to bear a bigger brunt of those upkeep costs. Of course, the providers don’t want to see their profits decrease, so the costs of the monthly service continue to spiral upward.

Second, with all the streaming options today ... Netflix, Amazon, Disney+, Hulu, and an ever-growing list of other entertainment providers ... it doesn’t make a lot of sense for many people to be tied to a rigid channel lineup. Most people don’t care about much more than a dozen of the hundred-plus channels in a typical cable TV lineup, and people want to watch what they want to watch on their own schedule.

Third, at least in the U.S., if you live near or in a larger city, there are a lot of additional choices of OTA (over the air) channels that were never available before digital TV became the norm.

Well, welcome to another choice, and a possible fourth reason to cut the cable TV cord: IPTV. It stands for Internet Protocol TV, and you can now

stream your favorite channels online. It will require some diligence on your part to be sure you can continue to receive your favorite channels, but in the end, it just may be worth it.



It could hardly be much easier to implement, too. Download the m3u playlists off of the internet, and play those playlists in VLC or Kodi. I use VLC, rather than installing Kodi, which I have found becomes a full-time job just to maintain. Voila! Free TV!

So where do you find the IPTV playlists? Well, you could Google them. There are literally hundreds of lists out there from different people compiling them. But, at best, it’s a crap shoot. Here’s why: the list of servers changes **daily**. So, what worked yesterday or works today, probably won’t work tomorrow. This is where some diligence on your part will pay off. Some lists have hardly any channels that play.

Others lists have a much greater percentage of playable channels. So, you will have to find a playlist provider that has a good track record, and you’ll have to download a new playlist every day.

Fortunately, I’ve found one such playlist. It is out of Germany, and their playlists have a high percentage of playable channels. It’s at a site called [GratisIPTV](#). The playlists are sorted by country, so you should be able to find the most current playlist for your locale. Since I’m in the U.S., I usually download the U.S. playlists. However, I have downloaded the UK playlists at times, and had good luck viewing them in VLC.

1. Here Copy and Paste United States iptv channels list

```
https://gratisiptv.com/dl/us-gratisiptvm3u-2019-12-23-1.m3u
https://gratisiptv.com/dl/us-gratisiptvm3u-2019-12-23-2.m3u
https://gratisiptv.com/dl/us-gratisiptvm3u-2019-12-23-3.m3u
https://gratisiptv.com/dl/us-gratisiptvm3u-2019-12-23-4.m3u
https://gratisiptv.com/dl/us-gratisiptvm3u-2019-12-23-5.m3u
```

2. Or click on link to download United States iptv channels list

[Download United States IpTV List 1](#)
[Download United States IpTV List 2](#)
[Download United States IpTV List 3](#)
[Download United States IpTV List 4](#)
[Download United States IpTV List 5](#)

I typically visit the site daily to download the most current playlist, and store it in a directory called “iptv” in my Downloads directory. I usually just right click on the link, and select “Save link as...” from the context menu in my web browser (usually Firefox). Do that for each of the playlists listed, if there are multiple playlist files as in the example above. It literally takes me less than a minute to do this every day.

Once you have the playlists downloaded, launch VLC. I find it easiest to simply drag the playlist I want to play from my file manager (Thunar in my case,

IPTV: Free Streaming Over The 'Net

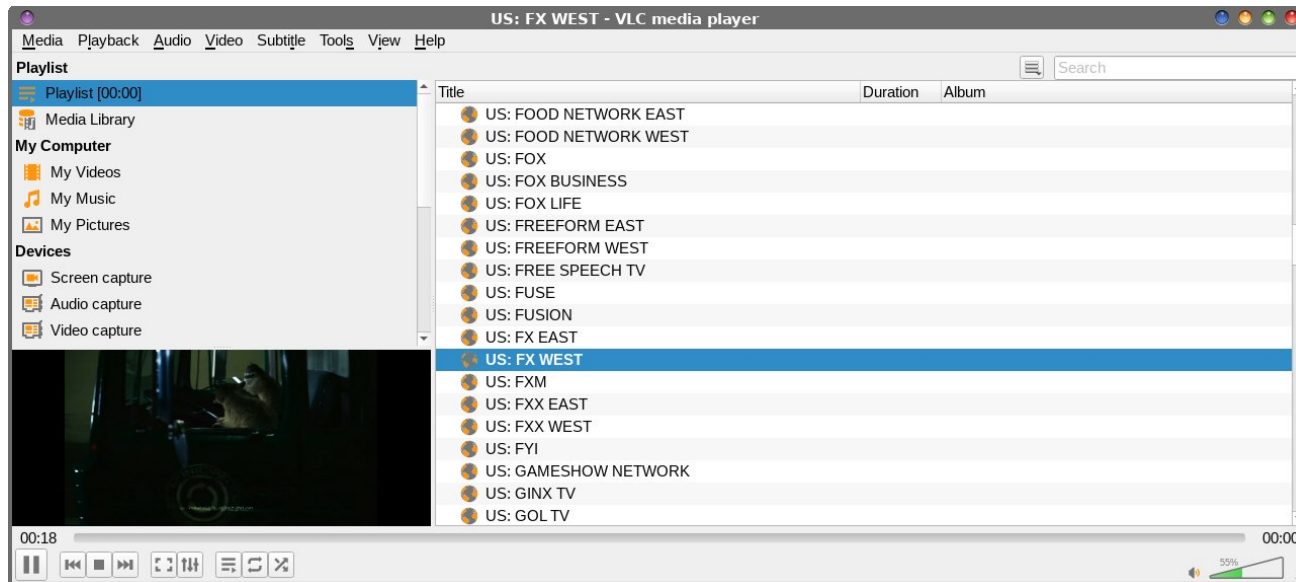
downloaded automatically every day in the background. Or, you could just run the bash script every day. Either way, it would greatly lighten the burden of keeping up with the maintenance tasks associated with using IPTV.

Summary

Of course, as with just about anything else, your mileage may vary. At home, and over my VPN, I can connect to channels all day long without any issues. Meemaw, on the other hand, has had sporadic luck, using the exact same playlists that I'm using. One day nothing would play, then the next, a file played right away. It's unclear if her ISP was blocking access, which is a real possibility. I know at the hospital where I work, the hardware-based firewall precludes/forbids connection to a VPN (that I have found so far), so I cannot use my VPN while connected to the hospital's wifi (as much as I would like to). The firewall at the hospital also prevents me from connecting to IRC. Likewise, very few channels play on my computer at the hospital. I guess having a few channels play is better than having none of them play. But the same channels play just fine at home from the same playlist.

If you are experiencing problems connecting with the playlists, I recommend trying it with a VPN. There are free VPN services out there such as [ProtonVPN](#) (no data limits, and no logs policy ... the latter is very important!), [Windscribe](#) (also no logs policy, with 10 GiB data limit per month), and [Hide.me](#) (no logs policy, with a 2 GiB data limit per month). I'm sure you might be able to find others. Be careful, though. Many of the "free" VPN services may limit your monthly data to only 500 MiB, so they really won't do well for streaming. You'll burn through that in no time at all.

I use a paid VPN service ([Private Internet Access](#), a.k.a. PIA), and it only costs me a few dollars a month (paid as a yearly plan). If you want to get an idea of what's available for free VPN services, check



since I'm running the Xfce desktop), and drop it into the VLC window. Alternatively, you should also be able to go to the "Media" menu, and select "Open file..." from the menu. Then, just select the playlist you want to explore.

When you first drop the playlist into VLC, you will likely see just the name of the playlist in the right pane. Double click it to open up all of the channel choices, which should then look something like what's depicted in the image above. Now, just double click the channel you want to watch, and you should be well on your way to viewing your choice of programming.

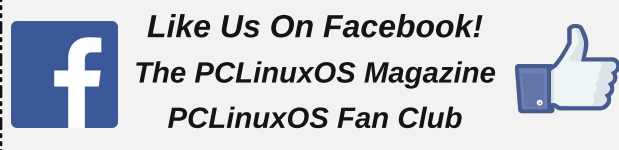
Most of the channels are presented in high definition, and you're likely to be quite amazed at the quality. If you're doing this on a HTPC (home theater PC), it's unlikely that you'll even realize that you're not watching it from cable TV.

Legality

The legality of doing this will vary, depending on the laws of your locale. Thus, I recommend streaming your channels through a VPN ... let's say, to avoid any "legal entanglements."

Possible Enhancements

It should be a relatively easy/trivial task to write a bash script that automatically downloads the playlist(s) using wget or curl. I just haven't done it yet, because I haven't yet had the time to put into it. From what I can tell, the naming scheme used doesn't change much. Only the date of the list really changes. So, the bash script can generate the appropriate filename, and then download the new files. You could also have the script delete any old m3u files in your "iptv" directory. Otherwise, you'll have to manually perform that task. Just imagine that after 30 days you'll have 150 different m3u playlists in your directory, if you download five playlists each day. Also, if you assign the bash script to a cron job, then the new playlists can be



out the TechRadar [article](#) on the subject. Many of the free services also offer paid VPN service. If you want to go with a paid VPN, check out the [article](#) on Tom's Guide that ranks the paid VPN services. ExpressVPN has consistently came out on top in just about every "VPN ranking" article I've seen lately. Even paying for a VPN service is way, way, way cheaper than the monthly cable bill, on the order of something like less than 10% of the cost.

You owe it to yourself to at least try it. You have little to nothing to lose. You may just discover a new way to (at least) find some entertainment value when you're on the go.



JUPITER BROADCASTING

Podcasts For Linux & BSD

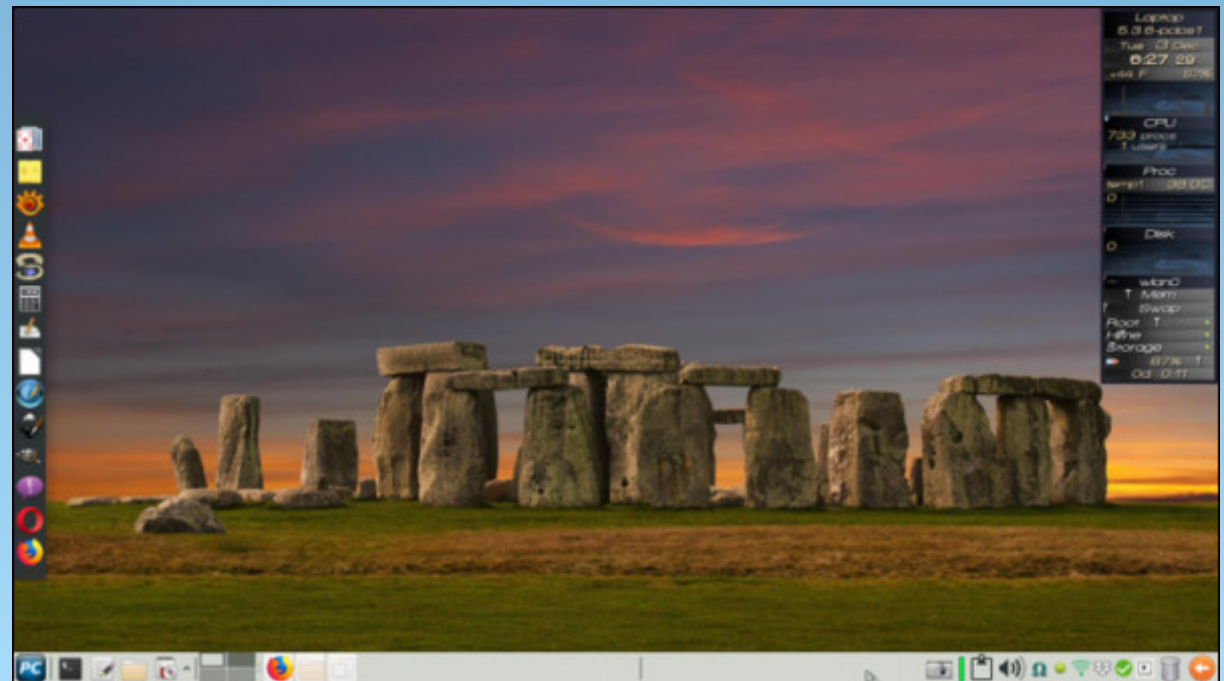
- LINUX ACTION >./NEWS
- LINUX UNPLUGGED
- LINUX >./HEADLINES
- TECH SNAP
- CHOOSE LINUX
- BSD NOW



PCLinuxOS-Cloud secure private simple-to-use

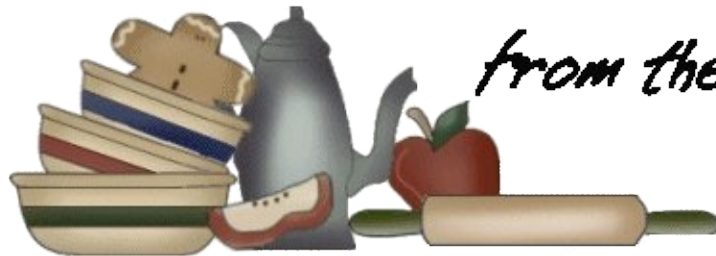
Sign up **TODAY!** pclosusers.com/services-signup.php

Screenshot Showcase



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

PCLinuxOS Recipe Corner Bonus



*from the kitchen of
youcantoo*

Ground Beef Stew over Garlic Mashed Potatoes

INGREDIENTS:

1/2 lb lean (at least 80%) ground beef
1/4 cup chopped onion
1 cup frozen mixed vegetables
1 can (14.5 oz) diced tomatoes, drained
1 jar (12 oz) beef gravy
1/4 teaspoon dried marjoram leaves
1/4 teaspoon pepper
1 1/3 cups water
2 tablespoons butter or margarine
1/2 teaspoon garlic salt
1/3 cup milk
1 cup mashed potatoes (dry)
1 tablespoon chopped fresh parsley

DIRECTIONS:

In 12-inch nonstick skillet, cook beef and onion over medium-high heat, stirring frequently, until beef is thoroughly cooked; drain.

Stir in mixed vegetables, tomatoes, gravy, marjoram and pepper. Heat to boiling. Reduce heat to low; simmer uncovered 8 to 10 minutes, stirring occasionally, until vegetables are tender.

In 2-quart saucepan, heat water, butter and garlic salt to boiling. Remove from heat. Stir in milk, dry potatoes and parsley just until moistened. Let stand about 30 seconds or until liquid is absorbed. Fluff potatoes with fork. Serve beef mixture over potatoes.

TIPS:

Oregano can be used in place of the marjoram; both are mint-family herbs from the Mediterranean mountains.

In place of the potatoes, try this thick, rich stew over biscuits, rice or pasta



Visit.
Contribute.
Build.

The PCLinuxOS
Knowledge Base

It Belongs To YOU!

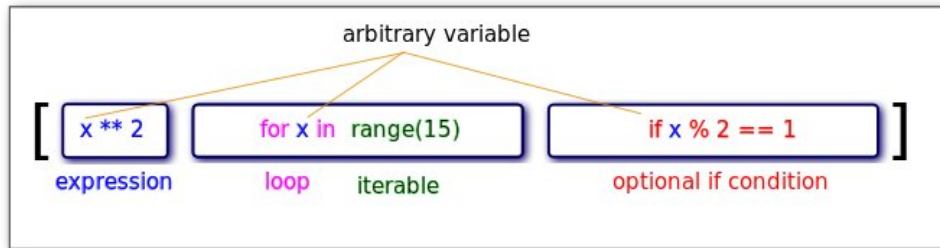


Casual Python, Part 12

by Peter Kelly (critter)

Comprehensions

In Python, a comprehension can be defined for a list, set or dictionary.



A comprehension takes an input sequence, and creates a new sequence by applying an output expression to its elements. The sequence is iterated over using an arbitrary variable, which is also used in the expression. An optional predicate may be added to filter data passed to the output expression. Confused? I'm not surprised, but it really is quite simple. An example may help.

Here is a list comprehension.

```
>>> single_list = ['one', 2, 3, 4]
>>> double_list = [i * 2 for i in single_list]
>>> print(double_list)
['oneone', 4, 6, 8]
```

The expression `i * 2` is applied to every element in the original list creating a new list. The square brackets tell python that this is a 'list' comprehension. In the first iteration, the `i * 2` expression is applied to the string 'one', resulting in a new string 'oneone'. Next the expression is applied to the integer 2 resulting in 4. Each result is stored in the new list `double_list`. Python comprehends applying the expression iteratively to a list. The expression must be suitable to be applied to the element, or an exception is raised. The comprehension may be applied to a slice of the original list.

```
>>> double_list = [i * 2 for i in single_list[1:3]]
>>> print(double_list)
[4, 6]
```

The list to be iterated over can be created on the fly, and the expression can be as complicated as you like.

```
>>> import os, glob
>>> glob.glob('*.py')
['dashboard_ui.py', 'qt_dashboard.py', 'partitions.py',
'wm_funcs.py', 'get_sensors.py']
>>> tmp_list = ['/tmp/' + i for i in glob.glob('*.py')]
>>> print(tmp_list)
['/tmp/dashboard_ui.py', '/tmp/qt_dashboard.py',
'/tmp/partitions.py', '/tmp/wm_funcs.py', '/tmp/get_sensors.py']
```

A list comprehension can be used to filter a list by adding a suitable predicate. For example, to get the squares of only even numbers in a sequence.

```
result = [i ** 2 for i in range(15) if i % 2]
>>> print(result)
[1, 9, 25, 49, 81, 121, 169]
```

or, to filter file by filesize

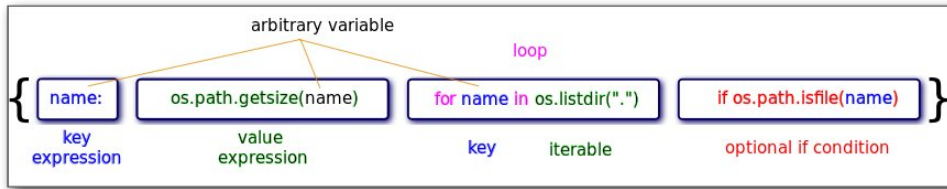
```
>>> med_files = [i for i in glob.glob('*.py') if 1300 <
os.stat(i).st_size < 2000]
>>> print(med_files)
['partitions.py', 'wm_funcs.py']
```

Dictionary comprehensions

Just like list comprehensions, a new data object is created by applying a function to the original objects elements. In this case though, a dictionary object is built and curly braces are used to enclose the comprehension.

```
>>> d = { 1: 'eins', 2: 'zwei', 3: 'drei', 4: 'vier', 5: 'fünf' }
>>> d_upper = {n: d.get(n).upper() for n in d}
>>> print(d_upper)
{1: 'EINS', 2: 'ZWEI', 3: 'DREI', 4: 'VIER', 5: 'FÜNF'}
```

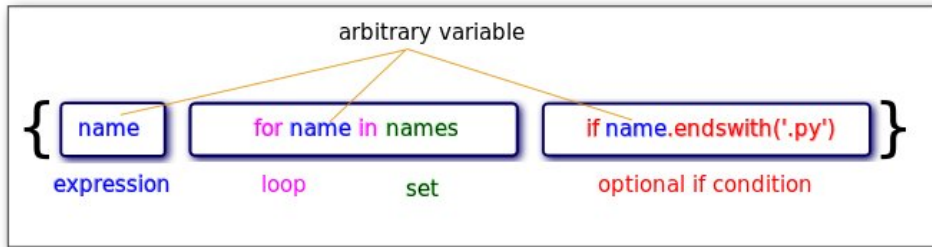
Download the Casual Python Source Code



Get a selection from the dictionary.

```
>>> d_upper = {n: d.get(n).upper() for n in d if n > 2}
>>> print(d_upper)
{3: 'DREI', 4: 'VIER', 5: 'FÜNF'}
```

Set comprehensions



A list may contain duplicates, but a set may not. Set comprehension may be used to remove duplicates. The resultant set may be converted back to a list using the type constructor list().

```
>>> names = ['obama', 'Bush', 'clinton', 'bush', 'REAGAN',
             'carter']
>>> cap_names = { name[0].upper() + name[1:].lower() for name in
                 names}
>>> print(cap_names)
{'Bush', 'Reagan', 'Carter', 'Clinton', 'Obama'}
```

Map, filter, reduce and zip

These are four functions built into Python, which are extremely useful.

Map

The map() builtin applies a function to each element of an iterable in turn. Rather than loop over the list of items, we can use map. Often the function used is a lambda. The map function returns a map object so we should convert it to something we can use. In the example the list() function is used.

```
>>> from math import pi
>>> diameters = [1, 2, 3, 4, 5]
>>> areas = list(map(lambda x: x * pi, diameters))
>>> print(areas)
[3.141592653589793, 6.283185307179586, 9.42477796076938,
12.566370614359172, 15.707963267948966]
```

It is also possible to pass the elements to a list of functions.

```
>>> def squared(x):
...     return x ** 2
...
>>> def cubed(x):
...     return x ** 3
...
>>> funcs = [squared, cubed]
>>> values = [3, 7, 64]
>>> for x in values:
...     powers = list(map(lambda f: f(x), funcs))
...     print(powers)
...
[9, 27]
[49, 343]
[4096, 262144]
```

Filter

The filter builtin returns a value each time a function returns True. The object returned is an iterator, which here is passed to the list function to collect valid items. The target is a list of various object types, from which we filter out only the integers.

```
>>> mixed_list = [13, 'songbird', 7, 5, 3.14159, (72, 34)]
>>> ints = list(filter(lambda x: isinstance(x, int), mixed_list))
>>> print(ints)
[13, 7, 5]
```

The last item in the list, although it contains integers, is a tuple and is therefore not included in the results.

Download the Casual Python Source Code

Reduce

In Python 3, reduce has been moved to the functools module and needs to be imported. Strictly speaking, this means that it is no longer a builtin function, but it has always been grouped with map, filter and zip, so is included here. The reduce function will perform some function on a list of items and return the result.

```
>>> from functools import reduce
>>> product = reduce( (lambda x, y: x * y), [2.34, 1.87, 3.92] )
>>> print(product)
17.153136
```

This applies the function $x * y$ cumulatively to the sequence [2.34, 1.87, 3.92] using arguments x, y the result is a single value: $17.153136 = 2.34 * 1.87 * 3.92$

Zip

Zip is a built-in function, and should not to be confused with the more common compression utility. Zip is a generator that returns an iterable. It takes two collections and merges them in pairs, until the shortest collection is exhausted.

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6, 7]
>>> z = zip(a,b)
>>> print(z)
<zip object at 0x7f62f6a33820>
>>> z = list(zip(a,b))
>>> print(z)
[(1, 4), (2, 5), (3, 6)]
```

When used in a for loop, parallel iterations are possible.

```
>>> for (x,y) in zip(l1,l2):
...     print(x, y, '--', x * y)
...
2.345 5.678 -- 13.314910000000001
3.456 6.789 -- 23.462784
4.567 7.89 -- 36.03363
```

Zip can be useful in constructing a dictionary from a collection of keys and a collection of values.

```
>>> partners = dict(zip(
...     ['Fred', 'Yogi', 'Tom'],
...     ['Wilma', 'Booboo', 'Jerry']))
>>> print(partners)
{'Fred': 'Wilma', 'Yogi': 'Booboo', 'Tom': 'Jerry'}
```

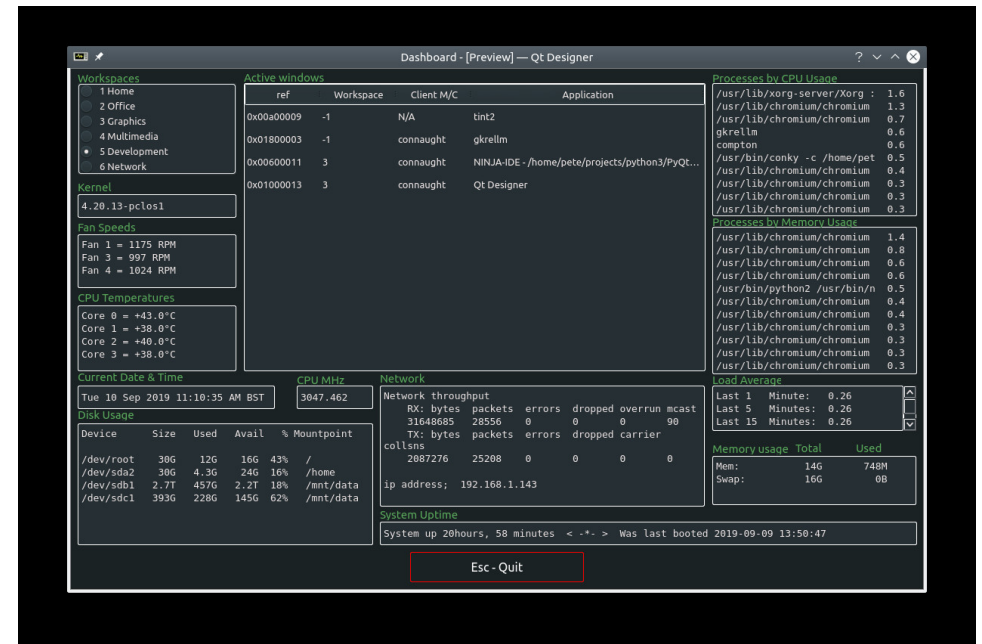
Or, using zip in a dictionary comprehension.

```
>>> partners2 = {k: v for (k, v) in zip(
...     ['Fred', 'Yogi', 'Tom'],
...     ['Wilma', 'Booboo', 'Jerry'])}
>>> print(partners2)
{'Fred': 'Wilma', 'Yogi': 'Booboo', 'Tom': 'Jerry'}
```

Dashboard

My demonstration application this time I have called dashboard. It is a single window containing the results of several system monitoring routines for a Linux system. Each set of results is continuously updated. This version has 14 sets of results, but you may have as many or as few as you find useful.

To implement this application, I have used two of my own standard modules that I can call on any time I need that functionality. In fact, I don't use all of the methods contained in the modules. I shall explain each of the modules and the structure of the main application, but not in as much detail as usual. By now you should be able to understand the code included here. You will be able to use as many or as few of the routines as you like, and construct a system monitor that you feel is useful. For this to work, you must ensure that all of the system commands that are called, such as wmic, are actually installed on your system.



As written, the application uses a dark theme, which is my preference, but if you prefer something else, just add or change the stylesheets in QtDesigner. As the application can be a little resource intensive collecting all of the data, I have included a little shell script to toggle its execution on and off. The full code for that, the application and the two modules, should be available for [download](#) from the magazine website.

In the above screenshot, there is one Quit pushButton, each of the single line boxes are labels, the large Active windows display is a Qtable widget, the workspaces are represented by six QradioButtons, one for each of my named workspaces. Only one radiobutton can be activated at a time, and this will show the current workspace. The green text items are labels, and the rest of the display widgets are QplainTextEdit widgets. Formatting of the display widgets is done by adding a stylesheet to each:

```
background-color: #263034;
color:#EDE4E4;
border: 1px solid white;
border-radius: 2px;
```

The rest of the design is arbitrary. Just use your imagination and style things as you like them.

This is the structure of the code for the main application.

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  import sys
6  import os
7  import subprocess
8  import socket
9  import time
10 from PyQt5.QtCore import *
11 from PyQt5.QtGui import *
12 from PyQt5.QtWidgets import *
13
14 from wm_funcs import getCurrentWorkspace, getManagedWindows
15 from get_sensors import getTemps, getFanSpeeds
16
17 import dashboard ui
18
19
20 class Dashboard(QMainWindow, dashboard ui.Ui_MainWindow):
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
```

Lines 5-9 are the imports required from python's standard library.

Lines 10-12 import the parts of the PyQt framework that will be used.

Lines 14-15 import some of the functions from my own modules.

Line 17 imports the file that resulted from building the user interface in QtDesigner.

Next, we define a class named Dashboard, and finally we create and execute the application.

The next stage of expanding the code is to look at what is contained in the Dashboard class.

```
20 class Dashboard(QMainWindow, dashboard ui.Ui_MainWindow):
21
22     def __init__(self):
23         super(Dashboard, self).__init__()
24         self.setupUi(self)
25         self.getKernel()
26         self.ip = self.getIpAddress()
27         self.btn_Quit.clicked.connect(self.exitApplication)
28         QTimer.singleShot(0, self.start loop)
29
30     def keyPressEvent(self, e):
31
32
33
34     def closeEvent(self, event):
35
36
37     def start loop(self):
38
39
40     def getDate(self):
41
42
43     def getUptime(self):
44
45
46     def getCpuFreq(self):
47
48
49     def getCpuProcs(self):
50
51
52     def getMemProcs(self):
53
54
55     def getMem(self):
56
57
58     def getDiskSpace(self):
59
60
61     def getNet(self):
62
63
64     def getKernel(self):
65
66
67     def getWorkspaces(self):
68
69
70     def getWindows(self):
71
72
73     def getFanRPMs(self):
74
75
76     def getCoreTemps(self):
77
78
79     def getLoadAverage(self):
80
81
82     def getIpAddress(self):
83
84
85     def exitApplication(self):
86
```

This class contains 20 methods, each one providing the details to be filled in to the text provided by the various widgets on the form, with the only exception being the `__init__` method that sets up everything for both the application and the display.

```
def __init__(self):      # initialize the window
    super(self.__class__, self).__init__()
    self.setupUi(self)
    self.getKernel()
    # called here as will not change this session
    self.ip = self.getIpAddress()
    # also unlikely to change
    self.btn_Quit.clicked.connect(self.exitApplication)
    QTimer.singleShot(0, self.start_loop) # set update timer
```

The only strangeness here is that we call two of the methods immediately to be displayed once throughout the run of the application, `getKernel` and `getIpAddress`. The resources reported by these methods are unlikely to change during the life of the application, so they can be called now and relied upon to remain unchanged. Other than this, the application is set up, the Quit button checked to see if it has been pressed, and the application therefore exited. Finally the timer is started via the `start_loop()` method to check for changes reported by the other methods.

The `keyPressEvent` and `closeEvent` methods are, by now, fairly standard methods in our code.

```
def keyPressEvent(self, e):      # escape key press
    if e.key() == Qt.Key_Escape:
        self.exitApplication()

def closeEvent(self, event):      # close button click
    self.exitApplication()
```

The `start_loop` method is called every time the timer 'fires'. What this does is to wait until the time 'fires', call a sleep period of 1 second so that things can stabilise, and then process each of the data-collecting routines in turn, updating the contents of the display and to check, via the `qApp.processEvents()` method, if the `exitApplication` method should be executed.

```
def start_loop(self):
    # loop through methods when timer fires
    while True:
        time.sleep(1)          # seconds between updates
        self.getUptime()
        self.getDate()
        self.getCpuFreq()
        self.getCpuProcs()
        self.getMemProcs()
```

```
        self.getMem()
        self.getNet()
    self.getWorkspaces()
    self.getWindows()
    self.getFanRPMs()
    self.getCoreTemps()
    self.getDiskSpace()
    self.getLoadAverage()
    qApp.processEvents()
```

One by one, these methods are, each with a brief explanation:

`getDate()`

Retrieves the system date and puts it into the `label_now` `setText` method. Python 3 uses utf-8 encoding by default, and so needs to be decoded to be understandable in your locale, gets the relevant text, and then formats it before passing it to be displayed.

```
def getDate(self):
    now = subprocess.check_output("date").decode('utf-8')
    time_details = now.split()
    now_date = (time_details[0:4])
    now_time = (time_details[4:])
    self.label_now.setText(now)
```

`getUptime()`

The `-p` option to the uptime command gives a nicely formatted (pretty) time string, and the `-s` option (since) shows when the system was last booted. Both outputs are then formatted before being passed to the `label_uptime.setText()` method.

```
def getUptime(self):
    up = subprocess.check_output("uptime -p",
    shell=True).decode('utf-8')
    bt = subprocess.check_output("uptime -s",
    shell=True).decode('utf-8')
    up_text = 'System ' + up.rstrip() + ' <--> Was
    last booted ' + bt
    self.label_uptime.setText(up_text)
```

The next five methods should, by now, be self explanatory.

`GetCpuFreq()`

Gets the current cpu frequency.

```
def getCpuFreq(self):
    f = subprocess.check_output("lscpu | grep 'CPU MHz'",
                                shell=True).decode('utf-8')
    freq = f.split()[2]
    self.label_freq.setText(freq)
```

getCpuProcs()

Gets a list of processes sorted by descending cpu usage.

```
def getCpuProcs(self): # first 10 only to fit in display
    cp = subprocess.check_output(
        "ps -eo cmd,%cpu --sort=-%cpu | sed '/CMD/d' |
        head -n10",
        shell=True).decode('utf-8')
    self.plainTextEdit_proc_cpu.setPlainText(cp)
```

getMemProcs()

Gets a list of processes sorted by descending memory usage.

```
def getMemProcs(self): # first 10 only to fit in display
    cp = subprocess.check_output(
        "ps -eo cmd,%mem --sort=-%mem | sed '/CMD/d' | head -n10",
        shell=True).decode('utf-8')
    self.plainTextEdit_proc_mem.setPlainText(cp)
```

getMem()

Gets the current memory usage.

```
def getMem(self):
    mem = subprocess.check_output('free -m | sed -n "2,3p"',
                                   shell=True).decode('utf-8')
    self.plainTextEdit_mem.setPlainText(mem)
```

If you can't follow the commands, try executing them progressively in a terminal so that you can see what each part is doing. For example:

```
ps -eo cmd,%cpu --sort=-%cpu
ps -eo cmd,%cpu --sort=-%cpu | sed '/CMD/d'
ps -eo cmd,%cpu --sort=-%cpu | sed '/CMD/d' | head -n10
```

This command outputs processes currently known to the kernel.

-e = all processes
 -o cmd, %cpu = output the command that started the process
 followed by its % cpu usage
 --sort=-%cpu = sort by descending (-%) cpu usage
 sed '/CMD/d' = delete any line containing the text 'CMD'
 this removes the header line
 head -n10 = show only the first 10 lines of output.

getDiskSpace()

Gets the current state of partition space usage.

```
def getDiskSpace(self):
    header = ('Device   Size   Used   Avail   %   Mountpoint\n\n')
    df_str = ('df -h ' + '--
              output=source,size,used,avail,pcent,target' +
              '| grep /dev/ | sed "s/      /      /" | sed -n -e
              "/^\/dev/p"')
    # the sed statement replaces 5 spaces with one
    ds = subprocess.check_output(df_str,
                                   shell=True).decode('utf-8')
    ds_text = header + ds
    self.plainTextEdit_disk_space.setPlainText(ds_text)
```

getNet()

This method simply calls the system ip command, and formats the output suitable for display. The one thing to note here is that the name of the network interface, here eth0, may be entirely different, particularly if you have a wireless connection. For example, on my laptop which runs Arch Linux, I have enp6s0 and wlp3s0 for the ethernet and wireless connections. Simply substitute the correct name for your setup.

```
def getNet(self):
    self.plainTextEdit_net.clear()
    throughput = subprocess.check_output(
        'ip -s link show eth0 | sed -n "3,6p"',
        shell=True).decode('utf-8')
    throughput_text = 'Network throughput\n' + throughput
    self.plainTextEdit_net.setPlainText(throughput_text)
    ip_str = 'ip addresss: ' + self.ip
    self.plainTextEdit_net.appendPlainText(ip_str)
```

getKernel()

This one should be fairly obvious.


```
def getKernel(self):
    k = subprocess.check_output("uname -r",
                                shell=True).decode('utf-8')
    self.label_kernel.setText(k)
```

The next four methods call functions from my own two modules.

getWorkspaces()

The output from calling `getCurrentWorkspace` is a two item tuple containing two strings: the current workspace number and the name. This is stored in the variable `cw`. A list is created of the six radiobutton objects (a list can contain any kind of object). The item in the list referenced by the returned workspace number (converted to an integer) has its `setChecked()` method set to 'True'. This will automatically reset all the other `RadioButtons` to 'False'.

```
def getWorkspaces(self):
    # only interested in current ws number - cw[0]
    cw = getCurrentWorkspace()
    wspace = [self.rb_ws1,
              self.rb_ws2,
              self.rb_ws3,
              self.rb_ws4,
              self.rb_ws5,
              self.rb_ws6, ]
    wspace[int(cw[0])].setChecked(True)
```

getWindows()

As this is the first time we have used a `QtableWidget`, it requires a little explanation.

The `getManagedWindows()` function returns a list of strings containing information about each of the windows currently managed by your window manager. The `wmctrl` command is a complex command that I don't intend to explain too much here. We set the row counter for the table widget to 0, clear the table contents, and then loop through the list of returned strings.

```
If not window:
    continue
```

breaks out of the loop when the list is exhausted.

Each string is split into a list of window items.

The first column, `c0`, is filled with the first item. This is the window managers reference to the window. The next item, which will go into the next column, `c1`, is

the workspace number. A value of -1 means the the window has the sticky attribute set – it will appear on every window. We test for -1, and if found, replace it with the string 'sticky'. This is not supported by all window managers and, as here, KDE gives us a very large number. I'll leave you to decide whether this is important enough for you to recode this part. The next item goes into `c2` and `c3` gets all the rest joined up. We make a list containing `c0`, `c1`, `c2` & `c3`. Looping through the list, we fill each column, and then increase the row count and go around again for the next list of items.

```
def getWindows(self):
    managed_windows = getManagedWindows()
    row = 0
    self.window_table.clearContents()
    for window in managed_windows:
        if not window:
            continue
        window_items = window.split()
        c0 = window_items[0]
        c1 = window_items[1]
        if c1 == '-1':
            c1 = 'sticky'
        c2 = window_items[2]
        c3 = " ".join(window_items[3:])
        col_list = [c0, c1, c2, c3]
        for column in range(4):
            self.window_table.setItem(row,
                                      column, QTableWidgetItem(col_list
                                                                    [column]))
        row += 1
```

getFanRPMs()

This simply calls the `getFanSpeeds()` function from my `get_sensors` module, and appends each value to `plainTextEdit` widget.

```
def getFanRPMs(self):
    self.plainTextEdit_fan_speeds.clear()
    fs = getFanSpeeds()
    for fan in fs:
        self.plainTextEdit_fan_speeds.appendPlainText(fan)
```

getCoreTemps

This method, like the previous one, calls a function from the imported module and populates the target widget.

```
def getCoreTemps(self):
    self.plainTextEdit_core_temps.clear()
    ctmp = getTemps()
    for core in ctmp:
```

```
self.plainTextEdit_core_temps.appendPlainText(core)
```

getLoadAverage()

This method runs the uptime command, chops and reformats the output, and then fills the plainTextEdit widget with the results.

```
def getLoadAverage(self):
    self.plainTextEdit_load_average.clear()
    up = subprocess.check_output("uptime", shell=True).decode('utf-8')
    averages = up.split('average:')[1].split(',')
    la1 = 'Last 1 Minute: ' + averages[0]
    la5 = 'Last 5 Minutes: ' + averages[1]
    la15 = 'Last 15 Minutes: ' + averages[2][:4]
    load_average = la1 + '\n' + la5 + '\n' + la15
    self.plainTextEdit_load_average.setPlainText(load_average)
```

getIpAddress()

I don't really understand how this code works, but it does, so I use it.

```
def getIpAddress(self):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    s.connect(("8.8.8.8", 80))
    return s.getsockname()[0]
```

exitApplication()

This is the standard method copied directly from our very first template application.

```
def exitApplication(self):
    '''quit the application'''
    self.close()
    sys.exit()
```

When we execute the code in this file, the following is executed and, hopefully, displays our application.

```
if __name__ == '__main__':
    app = QApplication(sys.argv)
    form = Dashboard()
    form.show()
```

```
app.exec_()
sys.exit(0)
```

The modules

This is the first of the two home brewed modules – wm_funcs.py. The code should be easy enough to follow. I have made it into a runnable script that executes each of the functions and outputs the results. This ensures that each function is operating satisfactorily.

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  import subprocess
6
7
8  def getCurrentWorkspace():
9      '''retrieves a list of available workspaces and identifies the
10         current one returns a tuple of strings'''
11
12     # get workspace list
13     wk_spaces = subprocess.check_output('wmctrl -d',
14                                         shell=True).decode('utf-8')
15
16     # get current workspace
17     for line in wk_spaces.split('\n'):
18         if line:
19             status = line.split()[1]
20             ws_num = line.split()[0]
21             ws_name = line.split()[8]
22             if status == '*':
23                 return ws_num, ws_name
24     ws_current = ws_num + ' (' + ws_name + ')'
25
26 def getWmInfo():
27     '''get information about the window manager and the environment
28        returns a string'''
29
30     wm_info = subprocess.check_output('wmctrl -m',
31                                       shell=True).decode('utf-8')
32     return wm_info
33
34
35 def getManagedWindows():
36     '''get a list of windows managed by the window manager
37        returns a list of strings'''
38
39     wm_windows = subprocess.check_output('wmctrl -l',
40                                         shell=True).decode('utf-8')
41     return(wm_windows.split('\n'))
42
43
44 if __name__ == '__main__':
45     print(getCurrentWorkspace(), '\n')
46     print(getWmInfo(), '\n')
47     print(getManagedWindows())
48
```

The get_sensors.py module:

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  import os
6  import subprocess
7
8  def getTemps():
9      ''' returns a list of strings containing
10         core information for each CPU core '''
11
12     temps = subprocess.check_output(
13         'sensors coretemp-isa-0000',
14         shell=True).decode('utf-8')
15     lines = temps.split('\n')[3:7]
16     core num = 0
17     core info = []
18     for core in lines:
19         core info.append('Core ' +
20             str(core num) +
21             ' = ' + core.split()[2])
22         core num += 1
23     return core info
24
25  def getFanSpeeds():
26      ''' returns a list of strings containing
27         fan information for all running fans '''
28
29     fan speeds = subprocess.check_output(
30         'sensors it8728-isa-0290',
31         shell=True).decode('utf-8')
32     lines = fan speeds.split('\n')[11:15]
33     fan num = 0
34     f info = []
35     for fan in lines:
36         rpm = fan.split()[1]
37         if rpm == '0': # ignore non-running fans
38             fan num += 1
39         else:
40             fan num += 1
41             f info.append('Fan ' +
42                 str(fan num) +
43                 ' = ' + rpm + ' RPM')
44     return(f info)
45
46
47  if __name__ == '__main__':
48      print(getTemps(), '\n')
49      print(getFanSpeeds())
50

```

You may need to run the sensors command in a terminal to discover the actual names of the sensors on your system, and change the code in lines 13 and 30 appropriately.

Dashboard toggle

This shell script may be used to start/stop the dashboard utility. If started by this script, then running the script again will stop it. If started by other means, it may or may not catch it. This script would ideally be executed using one of the launcher applications we have looked at. Change the path to the application as appropriate for your setup.

```

#!/bin/bash

# is dashboard already running?
G_STATUS=$(ps aux | grep [q]t_dashboard.py$)
# get the pid if so
G_PID=$(echo $G_STATUS | cut -d" " -f2)
# if there is a pid then is running
if [[ $G_PID != "" ]]
then
    # so kill it
    kill $G_PID
else
    # otherwise start it
    ~/mag/python/apps/qt5_dashboard/qt_dashboard.py &
fi
# done!
Exit 0

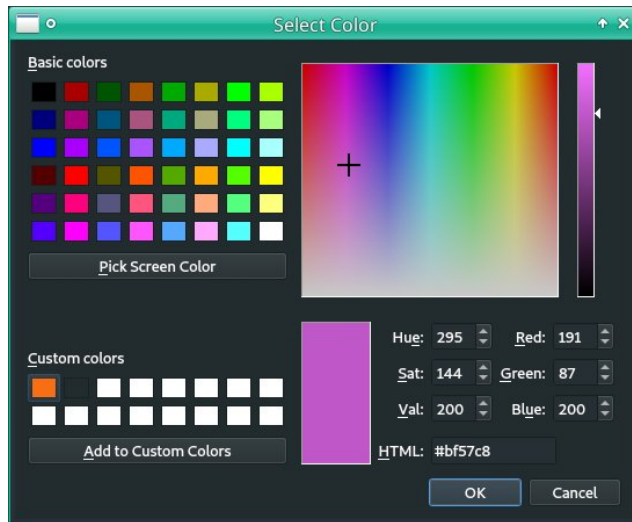
```

Moving on

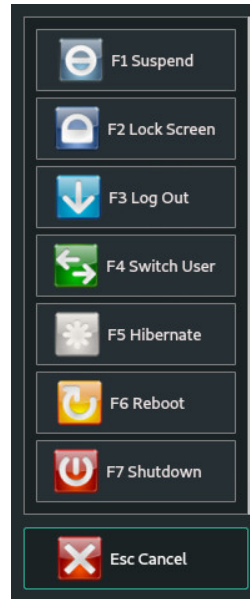
That takes us about as far as we can go and still call it 'casual python'. Anything beyond this is heading towards a more advanced level. If you have followed along with this so far, then you will be in a good position to take that next step. If you don't want to take it further, then you should now have sufficient knowledge to begin writing your own applications.

Although these articles have concentrated on providing graphical applications, there is no need to restrict the use of Python to this. As we have seen, Python has a host of modules that provide methods to interact with the system, and the data retrieved in this way can be filtered and reformatted in many ways to be displayed in a command line terminal.

What you probably need now is practice in order to become more fluent. To give you a few ideas, here are some screenshots of applications I have created. Some are very simple, some offer no features that have not already been covered, and others require a little research in order to implement the features.



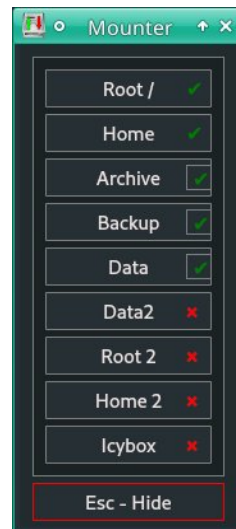
Color picker



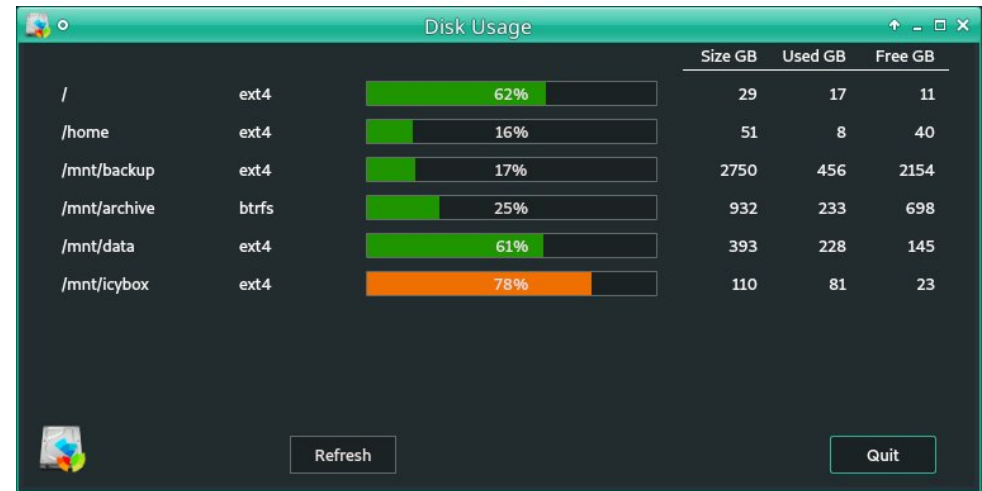
Shutdown Utility



Simple calculator



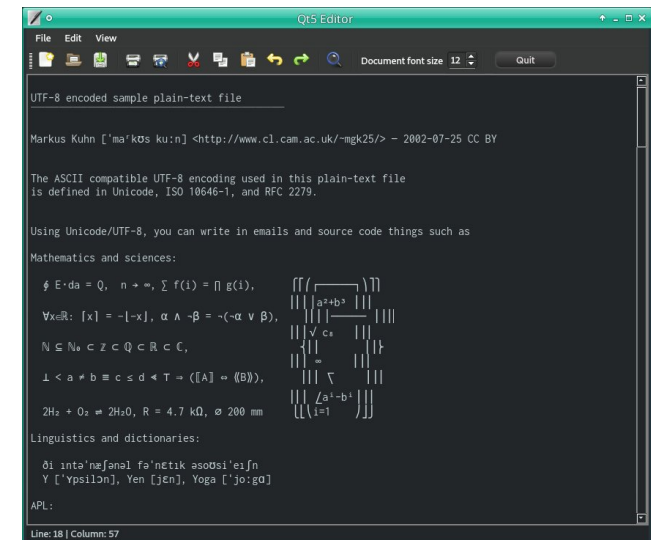
Partition Mounter



Disk usage monitor



Workplace Switcher



Simple Text Editor



Volume Control



Python 3 Git Mercurial Sphinx Vim Zshell

17. Regular expressions

Python uses the extended set of regular expression metacharacters generally known as 'Perl Compatible Regular Expressions or PCRE' but the syntax for using them is different and they are not available unless you import the built-in `re` module, or at least parts of it. Regular expressions are built up into strings and to avoid having to escape certain characters. Raw strings should be used by preceding them with 'r' (r'regexp' or r'regexp')

17.1. Metacharacters

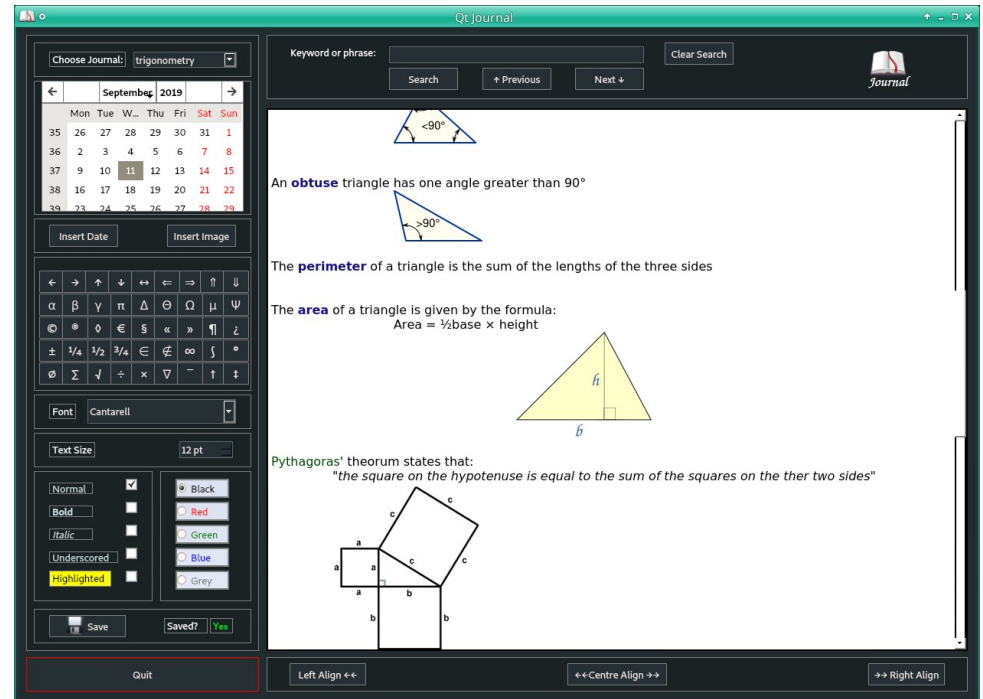
The recognized metacharacters are:

"."	Matches any character except a newline.
"^"	Matches the start of the string.
"\$"	Matches the end of the string or just before the newline at the end of the string.
"*"	Matches 0 or more (greedy) repetitions of the preceding RE.
"+"	Matches 1 or more (greedy) repetitions of the preceding RE.
"?"	Matches 0 or 1 (greedy) of the preceding RE.
"*?,+?,??"	Non-greedy versions of the previous three special characters.
{m,n}	Matches from m to n repetitions of the preceding RE.
{m,n}?	Non-greedy version of the above.
"\""	Either escapes special characters or signals a special sequence.
[]	Indicates a set [abc] or range [a-c] of characters. A "^" as the first character indicates a complementing set. No special meaning is attached to the characters + * . () \$ { } inside the brackets
" "	A B, creates an RE that will match either A or B.
"(...)"	Matches the RE inside the parentheses. The contents

HTML Browser – the html code is generated by sphinx, the python documentation generator



Video Player



Qt Journal

Choose Journal: trigonometry

Keyword or phrase: Search Previous Next Clear Search

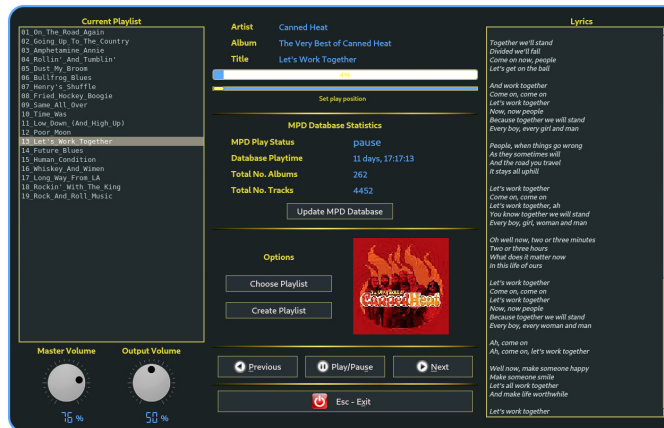
An obtuse triangle has one angle greater than 90°

The perimeter of a triangle is the sum of the lengths of the three sides

The area of a triangle is given by the formula:
Area = 1/2 base x height

Pythagoras' theorem states that:
"the square on the hypotenuse is equal to the sum of the squares on the other two sides"

Personal Journal/Note Taker



Current Playlist

Artist: Canned Heat
Album: The Very Best of Canned Heat
Title: Let's Work Together

MPD Database Statistics

MPD Play Status	pause
Database Playtime	11 days, 17:17:13
Total No. Albums	262
Total No. Tracks	4452

Options: Choose Playlist, Create Playlist


Master Volume: 75% Output Volume: 50%

Previous Play/Pause Next

Esc - Exit

Lyrics: Together we'll stand, Divided we'll fall, Come on, now, people, Let's get on the ball, And work together, Come on, come on, Let's work together, Now, now, people, Because together we will stand, Every boy, every girl and man, Let's work together, Come on, come on, Let's work together, You know together we will stand, Every boy, girl, woman and man, Oh we're young, two or three minutes, Two or three hours, What does it matter now, in this life of ours, Let's work together, Come on, come on, Let's work together, And now, make someone happy, Make someone smile, Let's all work together, And make life worthwhile, Let's work together

Music player with online lyrics fetcher



Vim Quick Reference

Start up

Actions

Objects

Movement

Windows

Tabs

Spelling

Marks

Buffers

Registers

Macros

Search

Visual mode

Shell Cmd

Folding

Sessions

RegEx

Exit - Write

Search and replace

/pattern - search for pattern

?pattern - search backward for pattern

* - search for current word

n - repeat search in same direction

N - repeat search in opposite direction

:vimgrep /pattern/ files - search for pattern in multiple files permits filename globbing

:%s/old/new/g - replace all old with new throughout file

:%s/old/new/gc - replace all old with new throughout file with confirmation

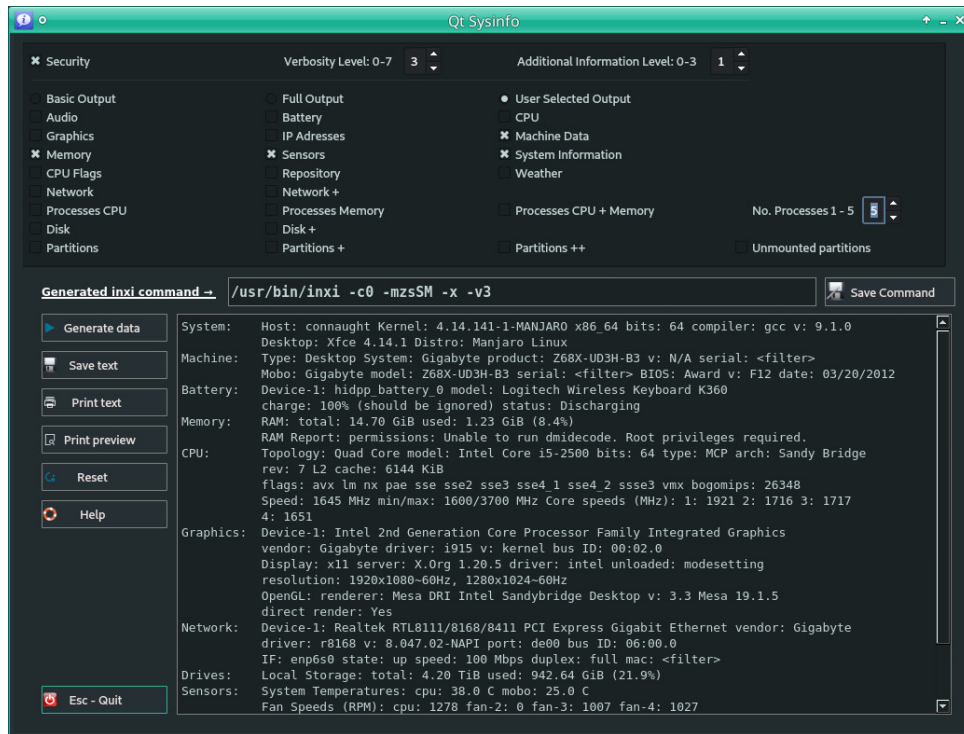
:noh - remove highlighting of search matches

F1 Help F2 LineNo F3 RelLineNo F4 Spell On F5 Spell Off F6 Spell Sugg

F7 HiLite off F8 Date F9 Split vert - F10 Split Hor F11 F12

Esc - Exit

Vim Quick Reference



Front end to inxi system analysis utility



Keyboard shortcut reminder – keypress sensitive



Weather forecast



PCLinuxOS.



Radically Simple.

PCLinuxOS

Available in the following desktops:

KDE LXDE Xfce

Openbox Gnome

Enlightenment e17



Like Us On Facebook!
The PCLinuxOS Magazine
PCLinuxOS Fan Club



Screenshot Showcase



Posted by mutse, on December 5, 2019, running Mate.

ms_meme's Nook: PCLOS Capers



A way down south near Houston town
You will find Texstar hanging around
He's got an OS of his very own
With every device you will like the price

Users in the forum night and day
With this OS they will always stay
Read the papers about PCLOS CAPERS
Download it don't delay

Not hard at all to install
You will find it will enthrall
Never ever will it let you down
Everything about it is so very sound

A Linux art easy to start
From it you will not part
Love it with all your heart
Be smart

Come behold and be bold
On this OS you will be sold
PCLOS is on the rise
Having it is very very wise

Toot it up scoot it up
Root it up boot it up
Read the PCLOS CAPERS
You will find Hot Stuff

PCLinuxOS Puzzled Partitions

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

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 277, average score 194.

Download Puzzle Solutions Here

PCLinuxOS Word Find: January 2020

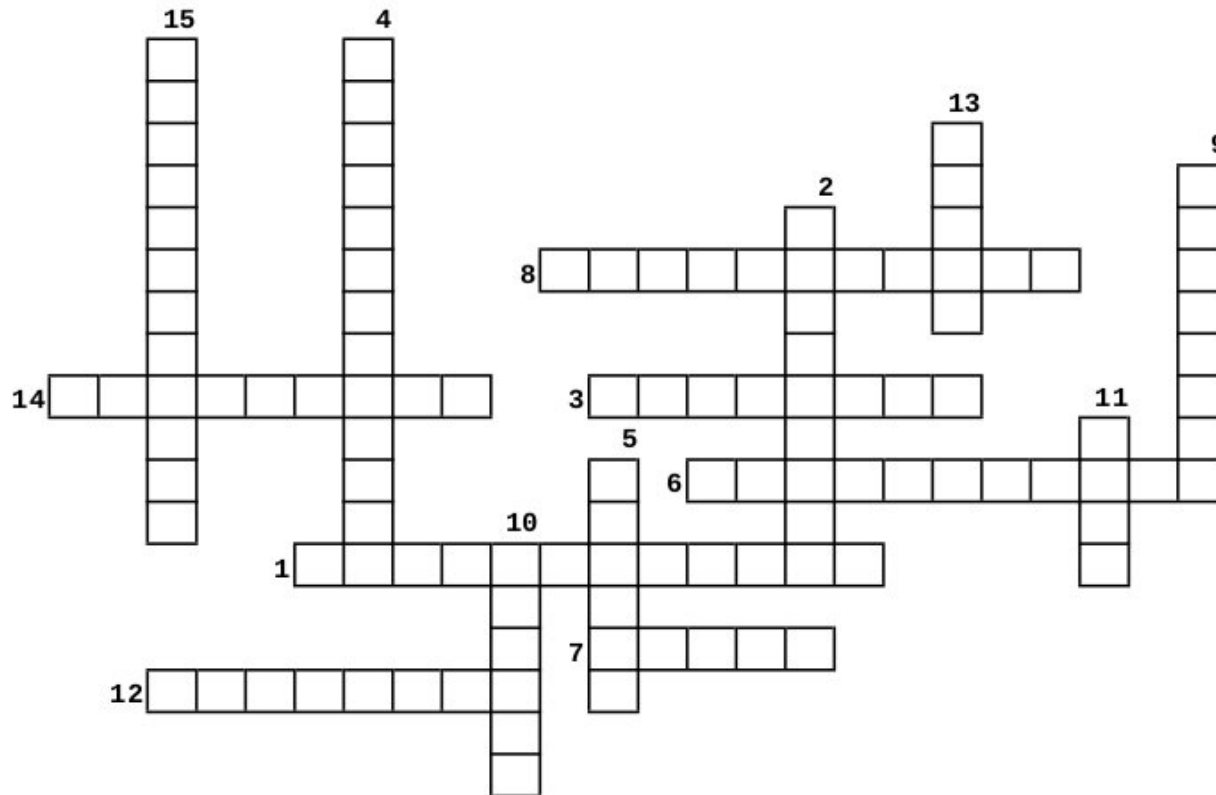
Archaeology

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

Antiquity	Archaeology
Artifacts	Bones
Burial	Chamber
Civilization	Corpse
Crypt	Curator
Decomposition	Deity
Desecration	Digging
Dolmen	Earthen
Entomologist	Epoch
Excavate	Exhibit
Expedition	Fossil
Fragments	Grave
Idol	Inscriptions
Looter	Megalith
Menhir	Monolith
Mummification	Rarified
Relic	Remains
Ruins	Sacred
Skeleton	Skull
Statuary	Temple
Tomb	Treasures
	Unwrap

[Download Puzzle Solutions Here](#)

Archaeology Crossword



1. a person who studies or is an expert in the branch of zoology concerned with insects.
2. objects made by human beings, typically an item of cultural or historical interest.
3. (of air) with little oxygen or without any of the problems of ordinary life
4. the state or process of rotting; decay.
5. a megalithic tomb with a large flat stone laid on upright ones
6. a scientific study of the ancient past
7. the beginning of a distinctive period in the history of someone or something.
8. treating a sacred place or thing with violent disrespect
9. a collection of statues
10. a tall upright stone of a kind erected in prehistoric times in western Europe.
11. a large vault, typically an underground one, for burying the dead.
12. a large stone that forms a prehistoric monument or part of one
13. an object surviving from an earlier time, especially one of historical or sentimental interest.
14. the ancient past, especially the period before the Middle Ages.
15. the society, culture, and way of life of a particular area.

[Download Puzzle Solutions Here](#)

Mixed-Up-Meme Scrambler



NO

for PCLinuxOS
Please Donate

Creed
GADMO

Contact
CHEAR

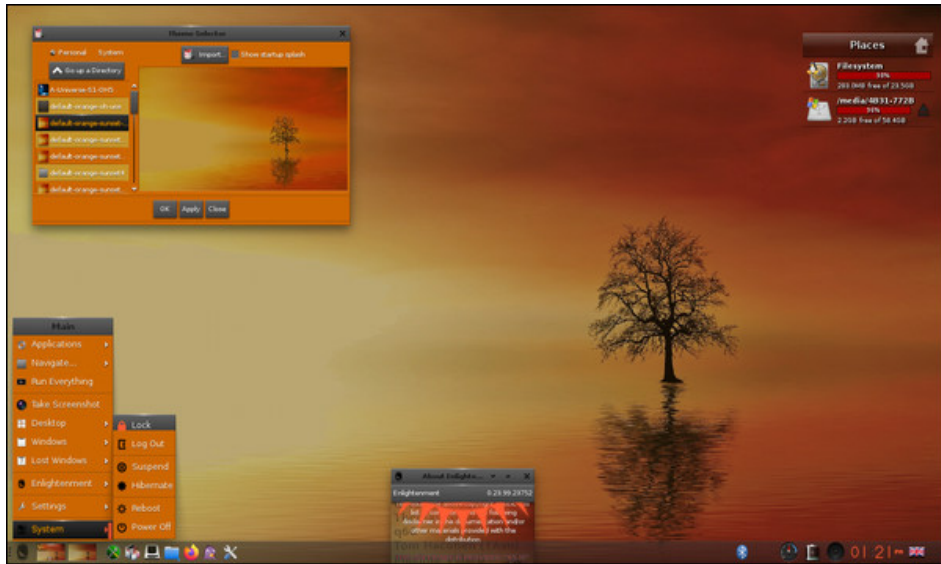
Alter
GANCHE

Untrue
RONGW

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 Only Human, on December 3, 2019, running enlightenment.



Posted by parnote, on December 4, 2019, running Xfce.



Posted by tuxlink, on December 5, 2019, running KDE.



Posted by Yankee, on December 6, 2019, running MATE.