

The PCLinuxOS magazine

Volume 102

July, 2015

* HTPC:
Using Kodi
To Display
Picture
Slideshows

GIMP
Tutorial:
More Photo
Editing
Tricks

Inkscape
Tutorial:
Creating A
Ketupat
Icon

Tip Top
Tips:
Terminal
With
Talking Cow

Game Zone: Slinki

Monitor Your System
With Conky, Part 2

HTPC: Kodi Keyboard
Shortcuts



And more inside ...

In This Month's Issue...

- 3** *Welcome From The Chief Editor*
- 4** *Monitor Your System With Conky, Part Two*
- 11** *ms_meme's Nook: Going Out Of My Mind*
- 12** *PCLinuxOS Recipe Corner*
- 13** *Screenshot Showcase*
- 14** *HTPC: Using Kodi To Display Picture Slideshows*
- 20** *Screenshot Showcase*
- 21** *GIMP Tutorial: More Photo Editing Tricks*
- 22** *Screenshot Showcase*
- 23** *Tip Top Tips: Terminal With "Talking Cow"*
- 26** *Inkscape Tutorial: Creating A Ketupat Icon*
- 27** *Screenshot Showcase*
- 28** *HTPC: Kodi Keyboard Shortcuts*
- 32** *PCLinuxOS Family Member Spotlight: The Chief*
- 33** *Screenshot Showcase*
- 34** *Game Zone: Slinky*
- 36** *Testimonial: My Thoughts About PCLinuxOS*
- 37** *Screenshot Showcase*
- 38** *PCLinuxOS Puzzled Partitions*
- 41** *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

daiaoshi

Khadis Thok

loudog

YouCanToo

Pete Kelly

Antonis Komis

Patrick Horneker

Smileeb

Contributors:

LinuxBorn

The PCLinuxOS Magazine is released under the Creative Commons Attribution-NonCommercial-Share-Alike 3.0 Unported license. Some rights are reserved.
Copyright © 2015.



Welcome From The Chief Editor

Ahhh! You just know (in the Northern Hemisphere, anyways) that summer is here. Birds are singing. Kids are playing, home during their summer break from school. Mosquitos are feasting on unprotected flesh. On most evenings, you can smell the sweet acrid smoke from barbecue grills wafting through the air. You can usually find me watching the Kansas City Royals baseball games on TV, while my son yells "Go Royals" whenever the crowd cheers.

Where I'm located, dab smack in the middle of the U.S., it's as if someone just flipped a switch. We went from mild temperatures to broiling literally overnight. Of course, in the American Heartland, late spring and early summer bring intense thunderstorms and that literally pop up over the top of you. And they bring severe weather conditions with them, such as tornadoes, hail, damaging straight line winds and flash flooding.

Seems everyone is experiencing some weather extremes. Sammy2fish recently posted in the forum, lamenting the lack of rain up in the area where he lives. California is experiencing one of its worst droughts in history. In Houston, Texas (the home of Texstar), they have experienced some horrendous flash flooding after receiving atrocious amounts of rain when tropical storm Bill made landfall from the Gulf of Mexico. Near

Meemaw's location in Southeast Kansas, flooding is a threat after receiving excessive rainfall totals.

Around my neck of the woods, there is ongoing scattered flooding. The ground is saturated, increasing the risk of flash flooding. The high humidity and heat of the day give birth to popup thunderstorms in the afternoon and evening. Recently, we had a line of very strong thunderstorms pass through in the middle of the night. We had 80 mph straight line winds. Limbs were sheared from trees. Some trees were uprooted completely, thanks in part to the saturated ground. We lost electrical power for most of a day, until service was restored. Over 120,000 electrical customers were left in the dark from the storm, as repair crews hurried to repair the damage.



Of course, without power, there is no internet connection via my cable broadband service. All of my laptops (which are always on) had run out of battery juice before I even woke in the morning. There is no TV. No fans. No air conditioning. No lights.

It's amazing how things you take for granted – electricity, fans, lights, air conditioning, computers, being able to connect to the internet – give you a pause when they are no longer available. It forces you to find other things to fill your time, causes you

to think differently about how to overcome your new "obstacles," and gives you time to reflect.

No matter where you may live, every area of the world has its pluses and minuses. Some live with the ever present threat of earthquakes, hurricanes, monsoons, and a whole host of other natural disasters.

Wherever you happen to live, I bid you peace, happiness, serenity and prosperity.

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.

Monitor Your System With Conky, Part Two

by Peter Kelly (critter)

Last month we learned the basics of conky, but now we are looking for WOW, so we will add Lua and Cairo.



Lua and Cairo

Now Lua and Cairo may sound like that nice couple you met on vacation and exchanged your e-mail addresses with, but that's not this couple.

Lua is an open source scripting language, developed and everything by the Pontifical Catholic University of Rio de Janeiro in Brazil and is spelled Lua, with a capital 'L'. Cairo is an advanced 2D graphics library. Conky can interface to Lua through a couple of hooks, and implements a few Lua specific functions.

What all of this means is that you need to provide a Lua script to Conky that perhaps uses one or more functions from the Cairo graphics libraries. Don't worry, it really isn't all that difficult. If you really don't want to go further, then there are plenty of conky configuration files that use Lua scripts available on-line, ready for you to use for your own purposes. But read on, you just might learn something you like.

Why Lua and Cairo?

Time to explain what we are trying to do here. Conky on its own can produce some fine output, but is limited to its own pre-defined functions. By adding the ability to use another scripting language, we gain access to many of the abilities of that language. Lua is an advanced and feature rich scripting language. Its addition removes many of the limitations of the basic Conky. How much it does this really depends on how far you want to go. Also, Lua is able to import external libraries of functions, and the Cairo graphical libraries feature some very useful routines using a consistent drawing model. Both Lua and Cairo are big subjects to approach, and here I will only just scratch the surface. I'll show the basic usage with some examples, and leave you to explore further on your own.

Conky doesn't *need* Lua and Lua doesn't *need* Cairo but Conky + Lua + Cairo = WOW.

Using Lua with Conky

Lua is a scripting language, and it reads a list of instructions in a plain text file known, not surprisingly, as a script. Conky therefore needs access to a Lua script, and this is achieved by adding these two lines before the TEXT section of the conky configuration file. The first line is:

lua_load the_path_to_your_lua_script (Lua scripts usually end in .lua)

This tells Conky which Lua script it should load. Obviously you need to provide a path to an actual Lua script. The script I shall use for purposes of demonstration will be

/home/user/demo.lua

The second line is:

lua_draw_hook_pre conky_demo_mag

This line tells Conky which function from the script to execute. The name of the function in the script should begin with conky_ as above but you may omit it here.

lua_draw_hook_pre demo_mag works just as well.

Also change:

update_interval 5 to ***update_interval 1*** or a suitable value that gives a good response.

And to give us a decent display area add this line

minimum_size 400 400

As always, the ways to achieve Lua integration with Conky are many and varied, but this method is fairly simple and has worked for me.

If you add just those first two lines to your template file, leave the text section blank and then re-save the file to perhaps ***.conkyrc_lua***, it will work. You can, of course, add more stuff to the TEXT section and have both Conky and Lua output together, but for now I'll keep things simple.

Now you need a Lua script named `demo.lua`. The Cairo libraries create their graphics by copying a source graphic onto a Cairo surface using masks and paths to produce the final output. That is not a very accurate description of the Cairo drawing model, but it is simplistic enough for us to use while we investigate further.

The first demo uses a text graphic as the source. Extra spaces are ignored in Lua scripts, so use spaces as required for readability. This is the basic Lua script, with reference numbers added.

```
1 require 'cairo'
2 function conky_demo_mag()
3 if conky_window == nil then return end
4 local cs = cairo_xlib_surface_create(conky_window.display, conky_window.drawable,
conky_window.visual, conky_window.width, conky_window.height)
5 cr = cairo_create(cs)
6 -- Start of output
7 cairo_select_font_face (cr, "Liberation Sans", CAIRO_FONT_SLANT_NORMAL,
CAIRO_FONT_WEIGHT_NORMAL);
8 cairo_set_font_size (cr, 24)
9 cairo_set_source_rgba (cr,1,0,0,1)
10 cairo_move_to (cr,80,200)
11 cairo_show_text (cr,"PCLinuxOS Magazine")
12 cairo_stroke (cr)
13 -- End of output
14 cairo_destroy(cr)
15 cairo_surface_destroy(cs)
16 cr=nil
17 end
```

This will simply output a line of text. For most purposes, you will only need to change the code between lines 6 and 13.

Line 1 tells Lua that the Cairo libraries are required to run this script.

Line 2 starts the definition of the function that we will call from Conky.

Line 3 checks that the Conky window exists, and if not, exits the function and, as there is nothing after the function, also exits the script.

Line 4 is more interesting. **local** means that what follows is local to this function only and will not be changed by other parts of the script, even if there is a name conflict. After this, a Cairo surface object is defined with properties that will: use the Conky window as a display, allow us to draw on to the surface on the Conky

window, make the graphics visible, and make the surface the same width and height as the Conky window.

This allows us to use coordinates relative to Conky. The surface type is assigned to the object `cs`, which is how we reference it.

Line 5 a surface of type `cs` is created and assigned to the variable `cr`.

Line 6 is a comment and will be ignored. Anything after "--" is considered a comment.

Line 7 sets the font to be used on surface `cr`. The font face is 'Liberation Sans.' The font slant is given by the cairo constant `CAIRO_FONT_SLANT_NORMAL`. Another option is `CAIRO_FONT_SLANT_ITALIC`. The font weight is given by the Cairo constant `CAIRO_FONT_WEIGHT_NORMAL`. Another option is `CAIRO_FONT_WEIGHT_BOLD`.

Line 8 Sets the font size.

Line 9 sets the color. The format is red, green, blue, alpha and uses values in the range from 0 to 1 instead of the 0 to 255 that we are more used to. Lua can do the conversion for us on the fly by using a statement such as `cairo_set_source_rgba (cr,234/255,0,107/255,1)`.

Line 10 moves the insertion point.

Line 11 sets the text to be shown.

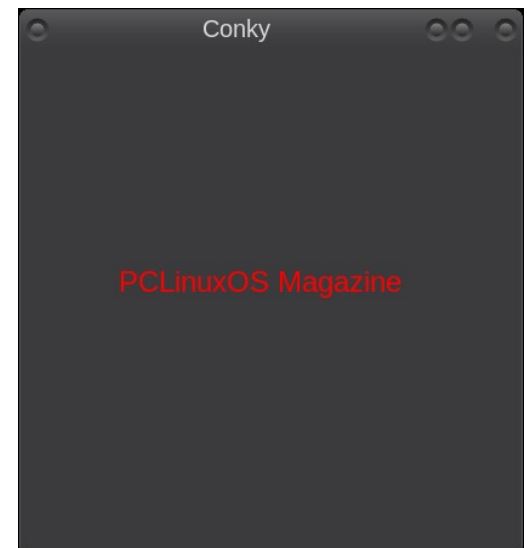
Line 12 finally draws the text.

Line 13 is another comment line.

Lines 14 – 16 clean things up before leaving the function.

Running `conky -d .conkyrc-lua` we get this (right). Well, it works but there is no WOW factor there. Nothing we couldn't do without all this extra effort.

Note that there may be a slight delay when using Lua and Cairo before anything appears on the Conky window, particularly when



parsing conky variables (as you'll see later). You may see a few warnings appear in the terminal until Conky and Lua get their stuff together. This is normal.

Change the part of the script between the two comment lines to read:

```
cairo_select_font_face (cr, "Liberation Sans", CAIRO_FONT_SLANT_NORMAL,
CAIRO_FONT_WEIGHT_NORMAL)
cairo_set_font_size (cr, 24)
cairo_set_source_rgba (cr,1,0,0,1)
cairo_move_to (cr,80,150)
cairo_rotate (cr, -math.pi / 6)
cairo_show_text (cr,"PCLinuxOS Magazine")
cairo_set_font_size (cr, 36)
cairo_set_source_rgba (cr,1,1,0,1)
cairo_move_to (cr,210,210)
cairo_rotate (cr, math.pi / 4)
cairo_show_text (cr,"Rocks!")
cairo_set_font_size (cr, 18)
cairo_set_source_rgba (cr,19/255,197/255,11/255,1)
cairo_move_to (cr,333,210)
cairo_rotate (cr, -3 * math.pi / 4)
cairo_show_text (cr,"Linux Rules OK")
cairo_stroke (cr)
```

Note that the `cairo_stroke (cr)` call is required only once for the three blocks of text, but that is not always the case. If in doubt, include the line. Run the Conky again and you should get a display similar to the one above. Progress!

But then again, I could have just created images of rotated text in almost any graphics application and displayed them with the 'normal' Conky as images.

The big step comes when you can display 'live' text and have it appear like this, as opposed to displaying static text. You can display text that comes from



interrogating the system, text that, for instance, reports the current battery charge level. It's quite easy to do.

So that's what we'll do next.

First I'd better explain some of those new or changed lines. I've added several lines, but most are repetitions of previous lines.

The first line that is new is

```
cairo_rotate (cr, -math.pi / 6).
```

This rotates the output by an amount equal to the result of `-math.pi / 6`. Lua uses radians for angles, starts at 3 o'clock or due east from the centre of a circle and rotates clockwise. Lua has a good range of mathematical functions and `-math.pi` returns the negated value of pi (π). As there are 2π radians in a circle, this equates to 30° counter-clockwise. Most people are happier using degrees, so we can get Lua to do our calculations for us.

```
cairo_rotate (cr,-30 * (math.pi / 180))
```

Tip: to make sure that you are getting results you expect you can use the `print` command to echo the output to the terminal

```
print (-30 * (math.pi / 180))
```

This is a general purpose debugging method but is invaluable when using math.

The second line of text has an *effective* rotation of 15° as the rotations are cumulative. To position the text where I have, it was necessary to use coordinates of 210,210 (which I found by trial and error), but this is obviously not 210,210 in a window of 400 pixels square, that would be the approximate centre. To retain a sensible coordinate system, it is necessary to rotate back to 0° , apply the translation and then rotate as required. This then becomes

```
cairo_rotate (cr, 30 * (math.pi / 180))    -- Rotate back to where we were
cairo_move_to (cr, 283,40)
cairo_rotate (cr, 15 * (math.pi / 180))
```

These are much more reasonable coordinates.

We can also tidy up the script a little by assigning the value of the Cairo constants to variables in our template before the beginning of the output, i.e. before line 6.

Monitor Your System With Conky, Part Two

```
f_noslant=CAIRO_FONT_SLANT_NORMAL
f_italic=CAIRO_FONT_SLANT_ITALIC
f_nobold=CAIRO_FONT_WEIGHT_NORMAL
f_bold=CAIRO_FONT_WEIGHT_BOLD
```

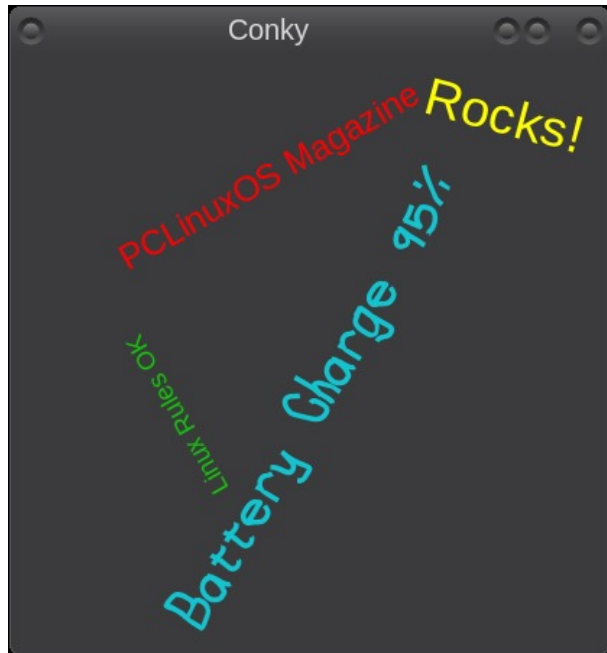
Then call them like this:

```
cairo_select_font_face (cr, "Liberation Sans", f_noslant, f_nobold);
```

To get the text from a conky variable into Lua, use the `conky_parse()` function. Insert these lines into the Lua script just before the `cairo_stroke (cr)` line.

```
cairo_select_font_face (cr, "Junkyard", f_noslant, f_bold)
cairo_set_font_size (cr, 38)
cairo_set_source_rgba (cr,19/255,197/255,208/255,1)
cairo_rotate (cr, 120 * (math.pi / 180))
cairo_move_to (cr,130,400)
cairo_rotate (cr, -60 * (math.pi / 180))
cairo_show_text (cr,"Battery Charge "..conky_parse("${battery_percent}".."%".."))
```

To concatenate the various parts of the output line we use two periods ..



Now, when you see a screenshot like the one below, you will know how it was done.



Drawing

We've played around with some text in Lua, but Cairo is a powerful library of graphic functions. We can use Cairo to draw lines, arcs and rectangles. Cairo is also capable of drawing more advanced items, such as curves.

Cairo curves are cubic Bezier curves, sometimes called paths or splines. They have a start point and an end point and various intermediate control points through which they smoothly flow but don't necessarily touch. The positions of the control points determine the shape of the curve and for this reason they are known as parametric curves. Bezier curves have infinite scalability, and as a graphical element, were initially developed for the production of car bodies. These are probably not so useful in Conky, and since they are quite complex topics, I shall not discuss curves in this article.

Lines

To draw a line, you use `cairo_move_to` to establish the start point, and the `cairo_line_to` function to draw a line segment. The `cairo_line_to` function can be repeated to draw multiple line segments, and the `cairo_close_path` function used to complete a polygon. The thickness of the line can be controlled by `cairo_set_line_width`, and the width is distributed equally about the start or apex point. The width cannot be changed within multiple line segments. The line width remains constant between invocations of the `cairo_stroke` function. For open ended lines, the line ending can be one of

CAIRO_LINE_CAP_BUTT, *CAIRO_LINE_CAP_ROUND* *or*
CAIRO_LINE_CAP_SQUARE

and the joints at the vertices of unclosed lines segments take the form of

CAIRO_LINE_JOIN_MITER, *CAIRO_LINE_JOIN_BEVEL* *or*
CAIRO_LINE_JOIN_ROUND

You might like to assign these to variables in your template file. If you draw a closed polygon, you can make it filled by replacing `cairo_stroke` by `cairo_fill`. Then the line width is reset to one. The following will draw a regular hexagon.

```
cairo_set_line_width (cr,4)
cairo_move_to (cr,300,200)
cairo_line_to (cr, 250,113)
cairo_line_to (cr, 150,113)
cairo_line_to (cr, 100,200)
cairo_line_to (cr, 150,287)
cairo_line_to (cr, 250,287)
cairo_close_path (cr)
cairo_stroke (cr)
```

That was a brief look at using lines in Lua with the Cairo libraries.

To do something more useful, as Cairo can draw arcs as well as lines, I'm going to build a pair of speedometer type gauges for the two cores of my processor. But first, we need to look at using arcs.

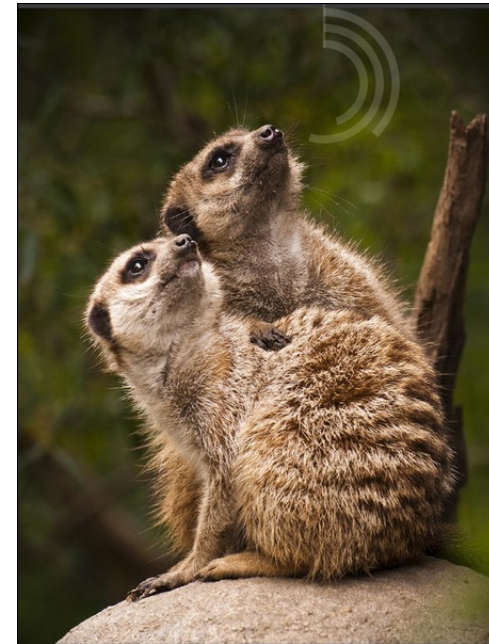
Arcs

An arc is part of a circle or, if the start and end point coincide, a full circle. Arcs can also be mixed with lines to draw compound shapes. To draw an arc, you need to specify the centre coordinates, the radius, and the start and end angles of the arc. It is also necessary to control the width, color and end type of the arc, although there are defaults.

There are two cairo arc commands:

cairo_arc draws the clockwise
cairo_arc_negative draws the arc counterclockwise

They both require the following arguments: surface reference (we are referencing our surface with `cr`), centre point x,y coordinates, start angle in radians, and end angle in radians.



That's it. Arcs are easy as they accept angles. If you know what it is you are monitoring, then you can make your Conky display unobtrusive. For this one I resized the Conky window to accommodate a graphic and then put these lines in my Lua template.

```
cairo_set_source_rgba (cr,220/255,218/255,213/255,0.2)
cairo_set_line_width (cr, 10)
cairo_arc (cr,390,100,90,-90 * (math.pi/180),45 * (math.pi/180))
cairo_stroke (cr)
cairo_arc (cr,390,100,70,-90 * (math.pi/180),105 * (math.pi/180))
cairo_stroke (cr)
cairo_arc (cr,390,100,50,-90 * (math.pi/180),72 * (math.pi/180))
cairo_stroke (cr)
cairo_set_line_width (cr, 3)
cairo_move_to (cr, 390,55)
cairo_line_to (cr, 390,0 )
cairo_stroke (cr)
```

I added a little transparency to the arcs. The angular values are random for the demo, but they could easily be parsed from Conky variables.

A new type of gauge

Tip: Use an editor that highlights matching parentheses for this (such as Kate or Geany).

First up is the pointer. I will only show the code for one core. The pointer has a fixed centre point about which it swings according to the value it is gauging. The lowest point is in the lower left quadrant and the highest in the lower right. I have chosen a value of 45° below the horizontal for both, which is equal to -225° for 0% and 45° for 100%, making up a total swing of 270° or 2.7° for each 1% of CPU activity. Lines don't understand angles, so we need to use some math to calculate the endpoints.

I am going to store the values I need in variables, and this allows us to globally change things more easily:

The CPU work rate is found by this: `cpu1=((conky_parse("cpu cpu1")))`

The angle becomes `angle1=((cpu1 * 2.7) -225)`

The centre point is set as `ctrx1,ctry1=100,200` This sets both x and y values

The length of the pointer is `p_len p_len=60`

The x coordinate of the end of the pointer is the length of the pointer multiplied by the cosine of the angle, the y value by the sine. Lua will do the math for us.

```
xval1=ctrx1 + (p_len * (math.cos ((angle1 * (math.pi / 180)))))
yval1=ctry1 + (p_len * (math.sin ((angle1 * (math.pi / 180)))))
```

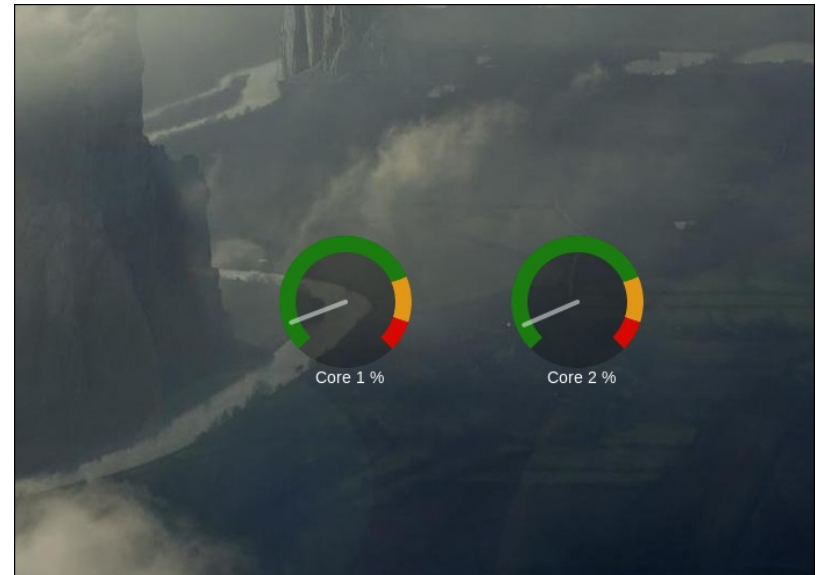
After we have set the color, width and end type for the line we are done.

```
cairo_set_source_rgba (cr,1,1,1,1)
cairo_set_line_width (cr,4)
cairo_set_line_cap (cr,CAIRO_LINE_CAP_ROUND)
ctrx1,ctry1=100,200
cpu1=(conky_parse("${cpu cpu1}"))
cairo_move_to (cr, ctrx1,ctry1)
angle1=((cpu1 * 2.7) -225)
xval1=ctrx1 + (80 * (math.cos ((angle1 * (math.pi / 180)))))
yval1=ctry1 + (80 * (math.sin ((angle1 * (math.pi / 180)))))
cairo_line_to (cr, xval1, yval1)
cairo_stroke (cr)
```

Now for the outer arc of the gauge. This should appear before the code for the line in the Lua script, as we want the line to appear 'above' the arc. The colours

that I want to use are: green, amber and red to represent the normal, caution and danger levels. I also want a black background. The arc centre will be the same as the pointer centre, so that variable can be re-used. The start angle is -225, the caution start angle I decided should be 75% of cpu usage and the danger start angle 90%. The danger end angle is obviously 100%. The line width needs to be greater, perhaps 14 pixels. As the centre point of the arc is the same as the centre point of the pointer and the pointer length is the same as the radius, we can re-use these values.

```
cairo_set_line_width (cr,14)
cairo_set_source_rgba (cr,0,0,0,0.3)
cairo_arc (cr,ctrx1,ctry1,p_len + 7,0,(2*math.pi))
cairo_fill (cr)
cairo_set_source_rgba (cr,27/255,124/255,16/255,1)
cairo_arc (cr,ctrx1,ctry1,p_len,-225 * (math.pi/180),((75 * 2.7)-225) * (math.pi/180))
cairo_stroke (cr)
cairo_set_source_rgba (cr,226/255,152/255,22/255,1)
cairo_arc (cr,ctrx1,ctry1,p_len,((75 * 2.7)-225) * (math.pi/180),((90 * 2.7) * (math.pi/180))
cairo_stroke (cr)
cairo_set_source_rgba (cr,217/255,8/255,0,1)
cairo_arc (cr,ctrx1,ctry1,p_len,((90 * 2.7) * (math.pi/180),45 * (math.pi/180))
cairo_stroke (cr)
```



Rectangles

Although lines can be used to draw rectangles, they are such a common object that cairo has a special function to create them, `cairo_rectangle`. The arguments you supply are the x,y coordinates of the bottom left corner, the width and the height. As you can imagine this makes drawing bar gauges a relatively easy task.

```
cairo_select_font_face (cr, "Liberation Sans", f_noslant, f_nobold)
cairo_set_font_size (cr, 18)
core1=(conky_parse("${cpu cpu1}"))
core2=(conky_parse("${cpu cpu2}"))
cairo_translate (cr, 100,200)
cairo_set_source_rgba (cr,1,1,1,1)
cairo_rectangle (cr,0,0,30,-100)
cairo_stroke (cr)
cairo_set_source_rgba (cr,0,1,0,1)
cairo_rectangle (cr,0,0,30,-core1)
cairo_fill (cr)
cairo_set_source_rgba (cr,1,1,1,1)
cairo_move_to (cr,-5,25)
cairo_show_text (cr,"core1")
cairo_move_to (cr,22,-25)
cairo_rotate (cr, -90 * (math.pi / 180))
cairo_show_text (cr,core1.." %")
```



I have used a new function here to move the insertion point `cairo_translate`.

This is by no means the extent of the capabilities of the Lua/Cairo partnership. That limit is your imagination which I hope to have fired up a little by these brief introductions.





Visit Us On IRC

- Launch your favorite IRC Chat Client software (xchat, pidgin, kopete, etc.)
- Go to freenode.net
- Type `/join #pclosmag` (without the quotes)



ms_meme's Nook: Going Out Of My Mind

Windows 8 is driving me insane
I have only myself to blame
I'm so blue you'd be too

It runs so badly I'm feeling so sadly
I can't think of anything to do

And I think I'm goin' out of my mind
A new OS I am trying to find
Overdue need something new

I boot up each morning
It resists me no matter how hard
I try to persist

Windows 8 is driving me insane
I'm losing all my brain
Out of my mind day and night
Night and day and night nothing right

OGG



MP3

I must think of a way
From Windows to part
There's no reason why
I should not have a new start

PCLinuxOS is now in sight
Soon everything will be alright
Oh so true that bull came through

No longer feel badly
I love it so madly
And so will you

Out of my mind over Tux
Never need to spend big bucks
Out of my mind day and night
Night and day and night all is right

PCLinuxOS Recipe Corner



*from the kitchen of
youcantoo*

Beef Parmesan with Garlic Parmesan Pasta

Ingredients

1 ½ lbs beef cube steak
½ tsp kosher salt
¼ tsp black pepper
¼ C prepared Italian salad dressing
½ C bread crumbs
½ C all purpose flour
½ C parmesan cheese, shredded
1 tsp garlic salt seasoning
¼ C olive oil
3 garlic cloves, sliced
1-1 ½ C prepared spaghetti sauce
1 C mozzarella cheese, shredded
½ lb angel hair pasta
4 tbsp butter
½ C parmesan cheese

Instructions

1. Cut cube steak into individual portions and season with kosher salt and pepper. Combine breadcrumbs and flour with parmesan cheese and garlic salt. Dip each piece of cube steak into the italian dressing and then coat in the breadcrumb mixture. Set aside.

2. Heat 2 tbsp of olive oil in a large skillet over medium high heat and cook sliced garlic in the oil until very lightly browned, remove just the garlic from pan and set aside. Brown the steaks in batches until golden brown on each side, adding more oil as needed and remove from the heat -- steaks do not need to be cooked all the way through. Arrange the steaks in the skillet and pour spaghetti sauce over the top and finish with shredded mozzarella cheese.

3. Bake uncovered at 400 degrees for 30-35 minutes or until bubbly and the steaks are cooked through, time will vary depending on thickness of the steaks.

4. While steaks are in the oven, cook pasta according to package directions and drain, saving 1 C of the starchy cooking water. Add reserved garlic, butter and parmesan cheese to hot pasta, adding the cooking water as needed to loosen it up. Serve steaks and sauce alongside the pasta.





Support PCLinuxOS! Get Your Official

PCLinuxOS
Merchandise Today!

PCLinuxOS



PCLOS-Talk

Instant Messaging Server

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



Screenshot Showcase



Posted by Meemaw, June 24, 2015, running Xfce.

HTPC: Using Kodi To Display Picture Slideshows

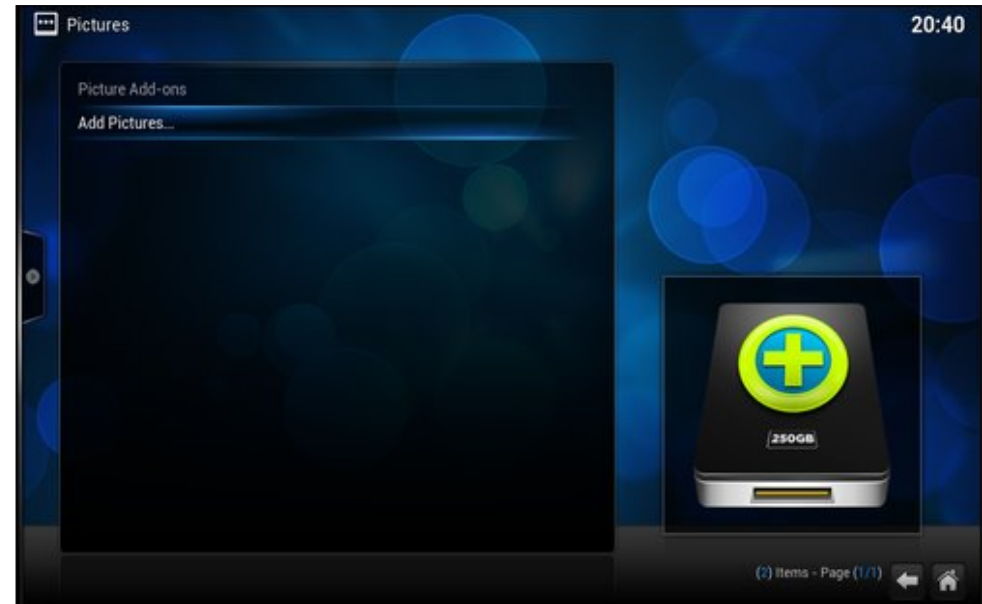
by Paul Arnote (parnote)

I know that I mentioned that Kodi seems to run on plugins, but in the case of displaying photos/images in a slideshow, nothing could be easier. Plus, it doesn't **require** any plugins – unless you're trying to display online content in a slideshow. We'll start with how to display images that are stored locally on your computer's hard drive.

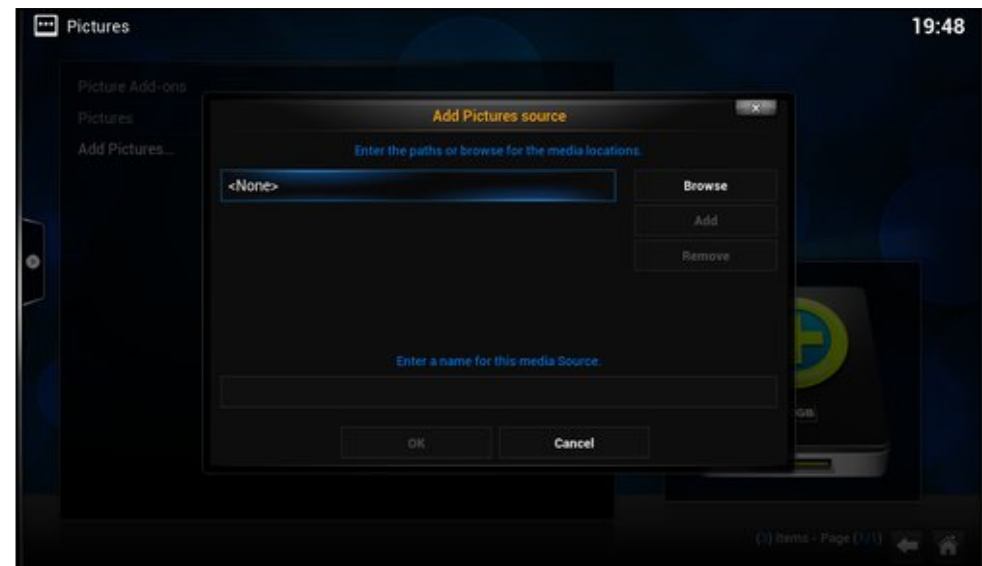
Displaying local images



First, you'll need to go to the “Pictures” category. Just click your mouse on the “Pictures” label.



Your screen should look like the image above. Click your mouse on the “Add Pictures...” item.



HTPC: Using Kodi To Display Picture Slideshows

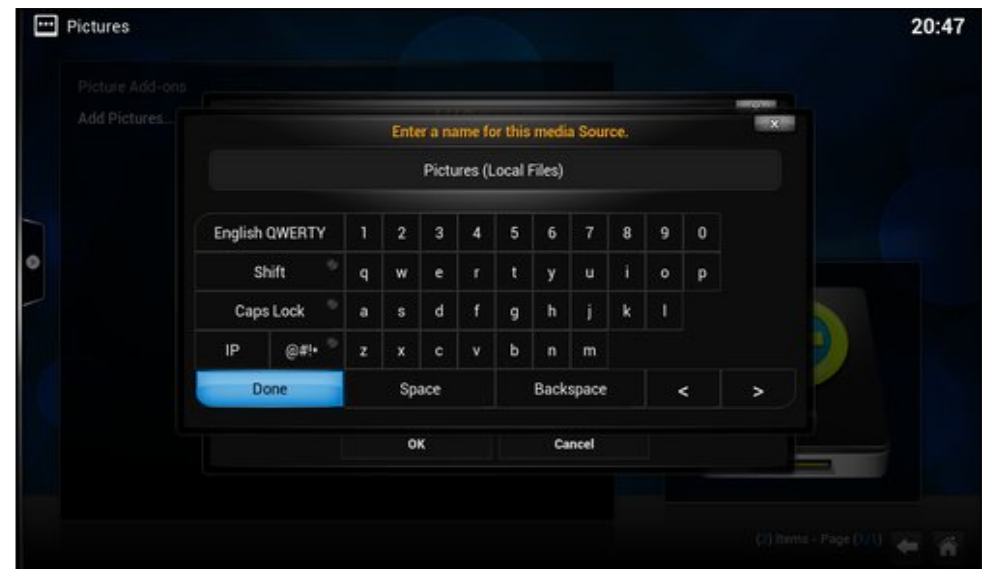
You will then see the “Add Pictures source” window, as in the image above. You can type in the full path to the directory containing your images, or you can use the “Browse” button on the right side of the window. The latter is the easiest way, and helps to insure that you avoid typing errors in the path statement.



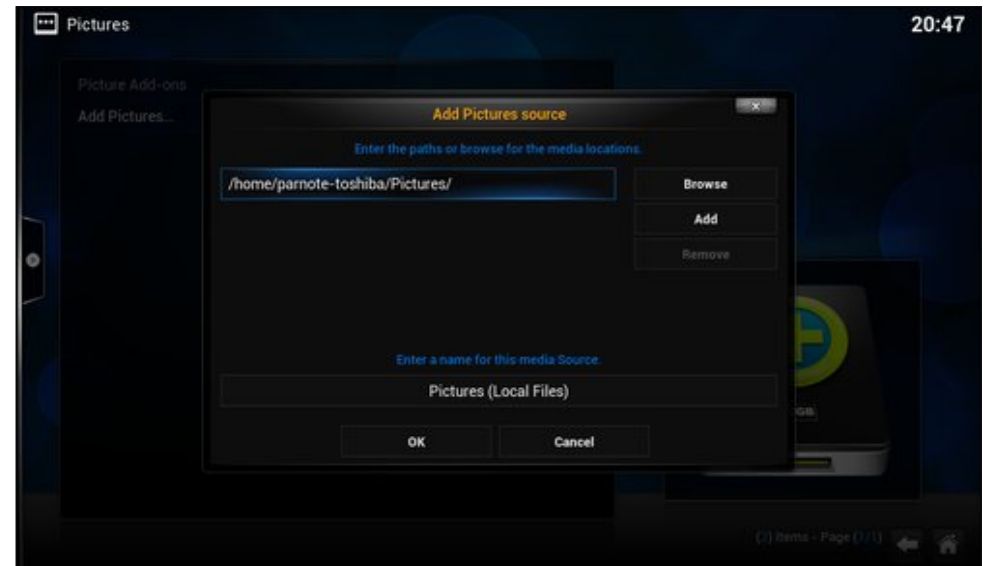
When you select the “Browse” button, you should see the image above on your screen. Select the “Home folder,” then navigate to the directory where you have your images stored that you want to display in the slideshow.



Then, click on the “OK” button.

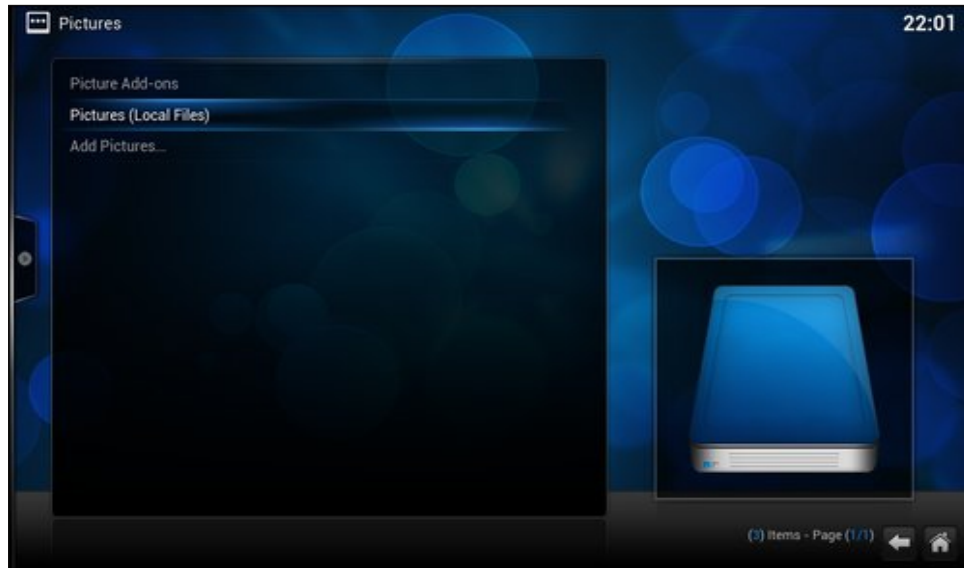


Now, click your mouse in the lower text entry box, and give the media source a name. I chose “Pictures (Local Files)” as the media source name so that it is immediately apparent that these files are stored locally on my computer’s hard drive. Click on the “Done” button on the on-screen keyboard (lower left in the image).



HTPC: Using Kodi To Display Picture Slideshows

When you are finished, you should see something like the image above (bottom second column, preceding page). Click on the “OK” button, and your local directory will be added to your list of media sources for the “Pictures” category.



You should see something like the image above, with your image source listed.



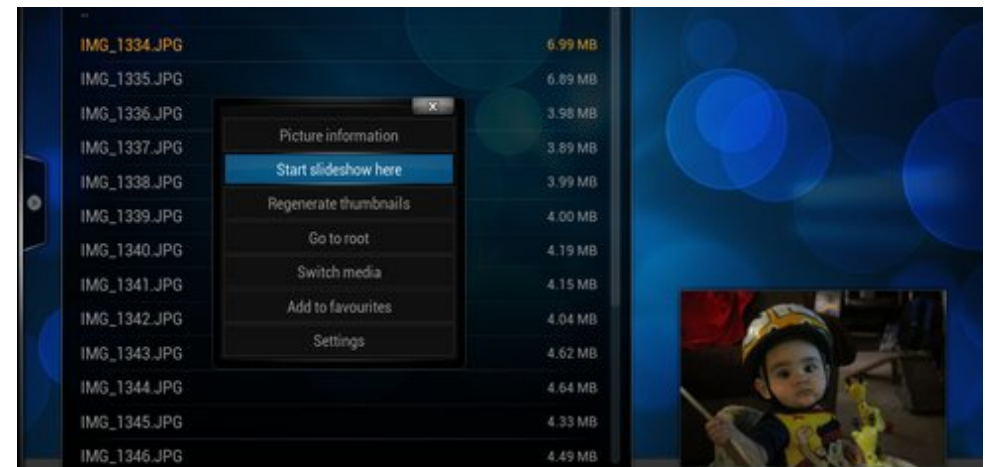
On my computer, I have a LOT of images, most of them sorted and stored in 64 different directories. Clicking on “Pictures (Local Files)” in the previous screen will display all of the subdirectories under my /home/parnote-toshiba/Pictures

directory. It will also display all of the images that aren’t sorted, and just dumped in that directory.

Highlighting the directory you want to display will cause Kodi to display thumbnail images (lower right of image) of the first four images in the directory. As an added bonus, Kodi can read archive files, such as .tar.gz and .zip files, and display any images that are contained in them – without you having to first decompress the archive.



Double clicking on the directory (or archive file) will display a list of all of the available images.



HTPC: Using Kodi To Display Picture Slideshows

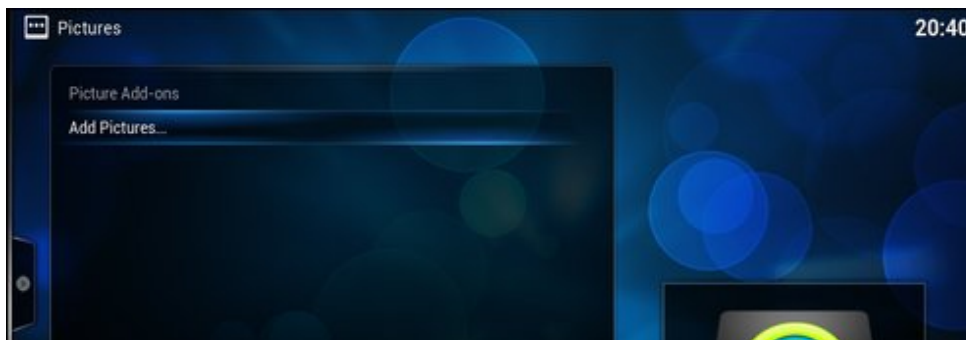
Right clicking your mouse on the first image will bring up a context menu window, similar to the one depicted above (bottom of second column, previous page). Select "Start slideshow here" from the menu.



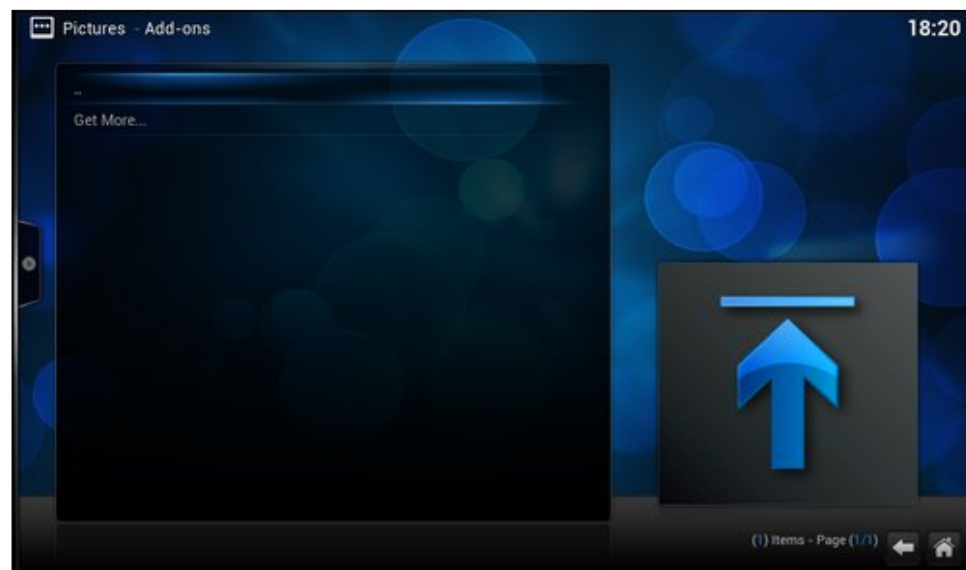
The slideshow will proceed with the settings from the "System > Settings > Pictures" section – which we reviewed in last month's "Getting Started" article.

Displaying images from online resources

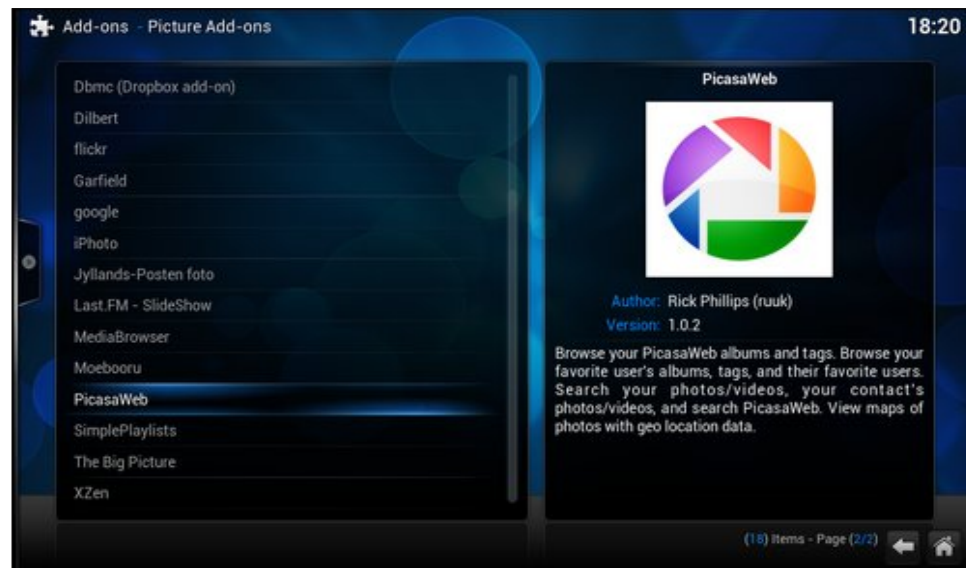
Of course, you're not restricted to displaying images that are stored locally on your computer's hard drive. For example, I store all of our images (mostly of our son Ryan, but also photos of other family members) on Google's Picasa website, to facilitate sharing them with friends and family members. Fortunately, Kodi makes it easy to create slideshows from images that you have stored in the cloud.



Click on the "Picture Add-ons" item.



Next, click on the "Get more..." item.



Now, select the online resource you want to use to display images. In my case, I chose "PicasaWeb" from the list.

HTPC: Using Kodi To Display Picture Slideshows



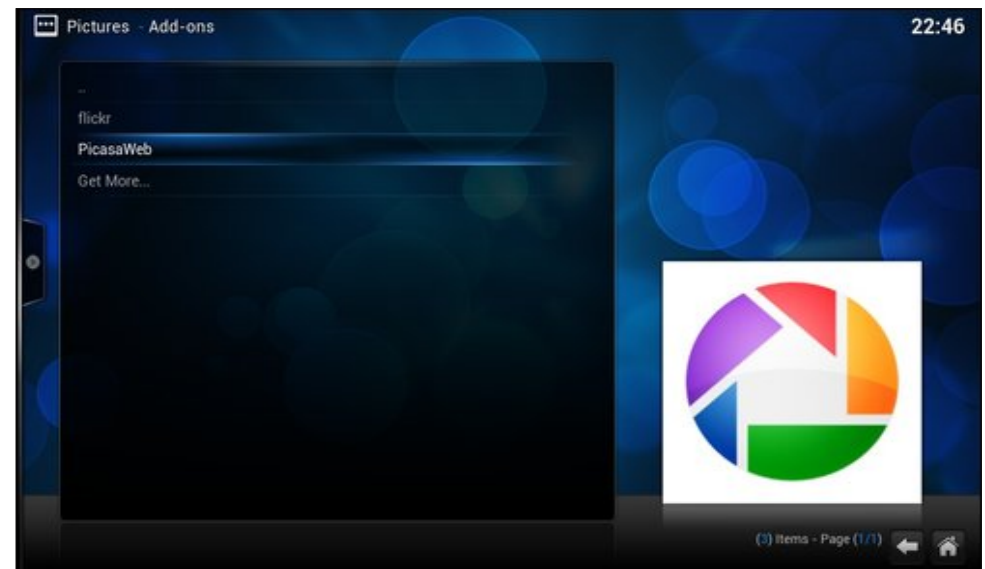
Clicking on the “PicasaWeb” add-on brings up the image above. Select the “Install” button.



Enable the “Use Login” option, then enter your Login Email and Login Password (click on each field to display the on-screen keyboard). Make adjustments to any of the other settings (self explanatory ... default values shown) you might like, then click on the “OK” button.

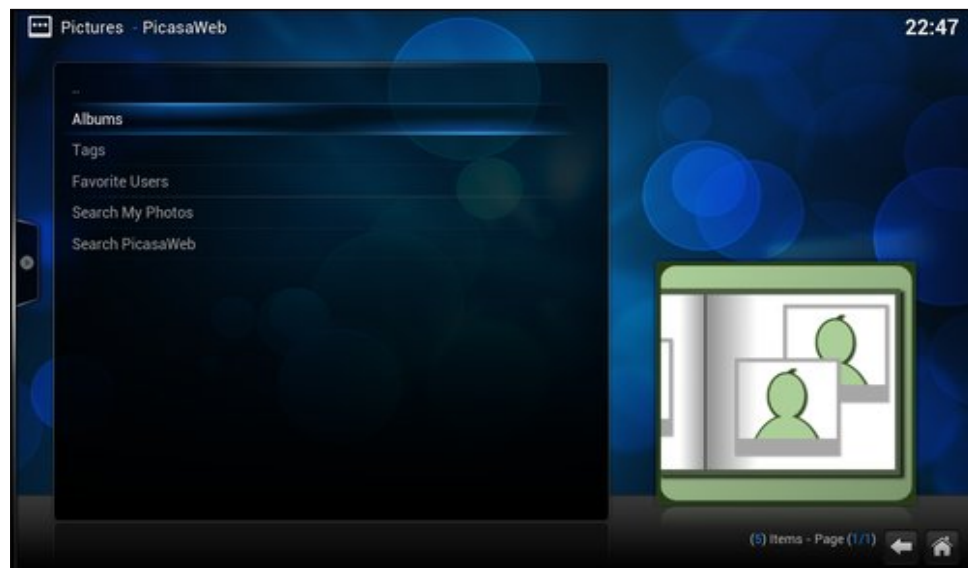


Once it's installed, you will want to double click on the PicasaWeb add-on and select the “Configure” button.

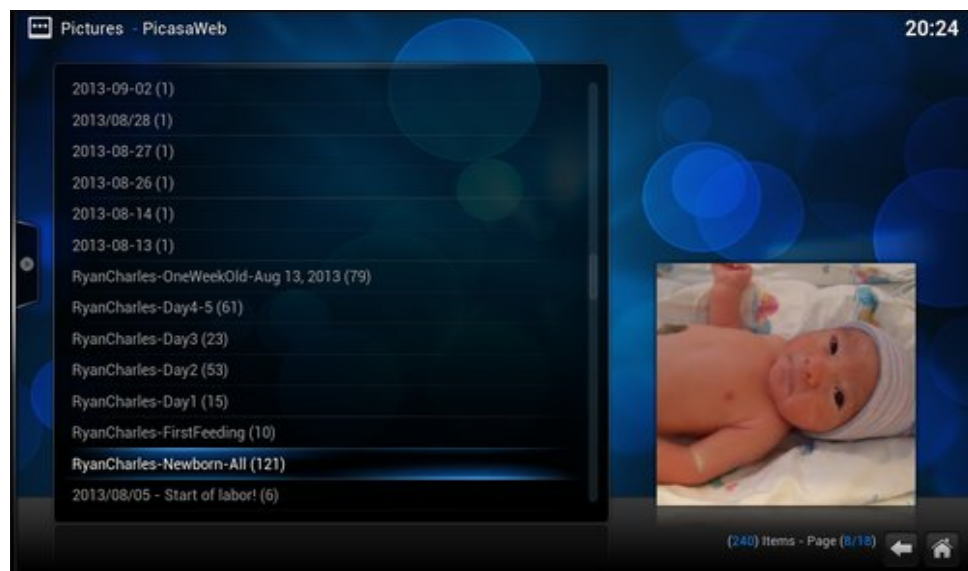


HTPC: Using Kodi To Display Picture Slideshows

Clicking on “Picture Add-ons,” you should see something like the image above (bottom of second column, previous page) on your screen. I selected “PicasaWeb” from the installed add-ons.



Select what you want to display from your online resource. Here, I selected “Albums” from the list. You can also select by tags, view the albums from “favorite users,” search your photos, or search PicasaWeb for specific images.



In just a few seconds, your album list should appear. Select the album you want to display, and your album’s “cover photo” will be displayed as a thumbnail in the lower right corner of the screen.

Click your mouse on the album you want to display, and a list of images in that album will be displayed. From this point on, you can display the slideshow in nearly the same way as you do with locally stored images, with one slight difference. With the PicasaWeb add-on, I’ve not been able to get Kodi to playback a slideshow by right clicking an image in an album and selecting “Start slideshow here....” Instead, go back one screen (so you’re not viewing the image list of the album), right click on the album itself, and select “Recursive slideshow” from the context menu.

You can also choose to stream images from Flickr, Google, or any of the other listed online resources. Even though I have a seldom used Flickr account, I installed/enabled the “flickr” add-on. Then, selecting the “flickr” add-on, I selected the “Interesting Today” category. The images then loaded, and I started a slideshow with the first listed image – all without logging in to my Flickr account.

Correcting image orientation

Sometimes, when browsing images – either locally stored images or those stored in the cloud – you may (most likely will) come across images that are not displayed with the proper image orientation. Thankfully, Kodi makes it simple to correct the image orientation.

Simply press the “R” key on the keyboard to rotate the currently displayed image 90 degrees clockwise (to the right). Refer to the **HTPC: Kodi Keyboard Shortcuts** article, elsewhere in this issue of The PCLinuxOS Magazine, for additional keyboard controls. There’s a whole set of keyboard shortcuts specifically for use by Kodi during the display of picture slideshows.

Playing music with your slideshow

Kodi uses a rather “low tech” way to allow you to to play background music with your slideshow. The only thing that would be lower tech would be to use another device (MP3 player, cassette tape, radio, etc.). To play music with your slideshow, you will need to start your music source (song, playlist, CD), then start your slideshow. The music will play in the background while your slideshow plays. We’ll talk about using Kodi to play music in a future article.

Summary

Kodi provides a very entertaining and easy way to view your images. You will end up with very “professional” looking results, very easily. The only thing that could make this better is if someone would come out with a plugin that allows you to save the slideshow as a video file, complete with background music. If it has been created (and who knows ... maybe it has), I haven’t been able to find it among the thousands of Kodi plugins that are out there in the wild. That would allow me to save the slideshow to a DVD, for posterity and provide another avenue by which I can share with others. If you do discover such an add-on, please let me know.

Donate To PCLinuxOS

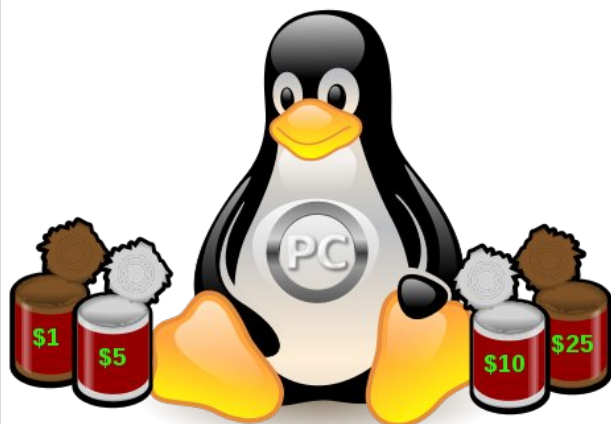
Community Supported.

No Billionaires/Millionaires.

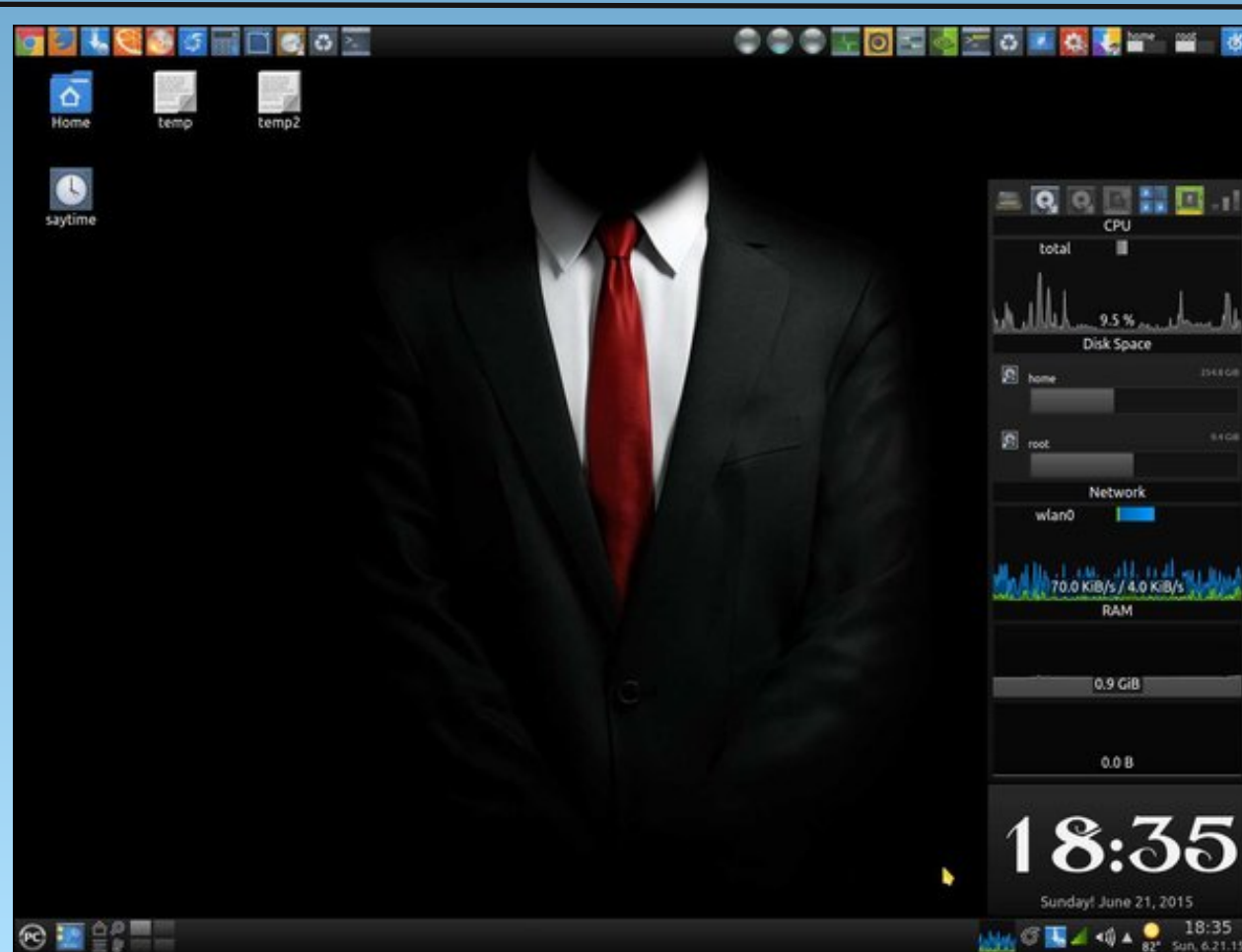
No Corporate Backing Or Funding.

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

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



Screenshot Showcase

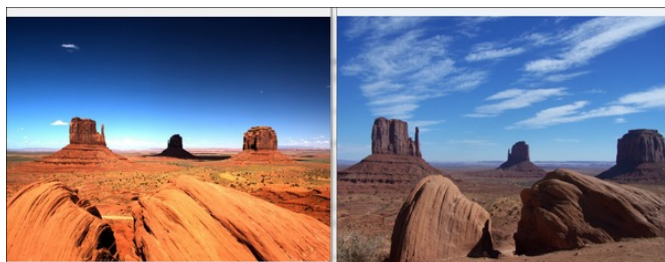


Posted by trytip, June 2, 2015, running KDE.

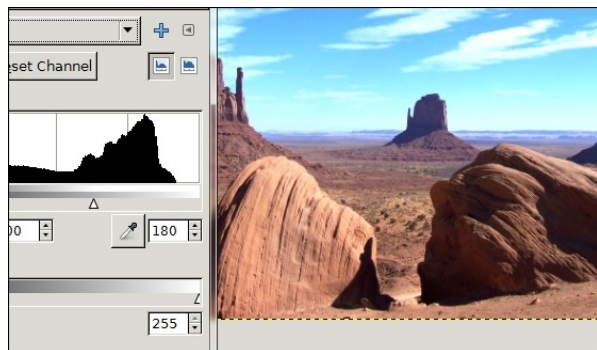
GIMP Tutorial: More Photo Editing Tricks

by Meemaw

I have a couple of wallpapers that I like very much, probably since I have been to the location pictured. Several years ago, I downloaded a wallpaper of "The Mittens" in Monument Valley in southeast Utah, USA. Then, I went there during a trip to Utah several years ago, and took one of my own. As you can see, the wallpaper I downloaded, on the left, is much more colorful than the photo I took, which is on the right. Bear in mind, the original photo on the left may have been a little more dull as well, until the photographer enhanced it.

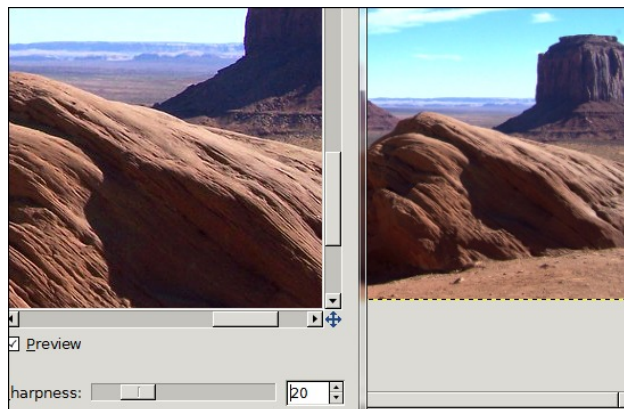


So, how am I to brighten up the one on the right? Let's use GIMP. Choose a photo you want to edit, and open a copy of it in Gimp. Navigate to **Colors** and select **Levels**. You'll see a histogram, which tells you how the pixels are distributed. If most of the

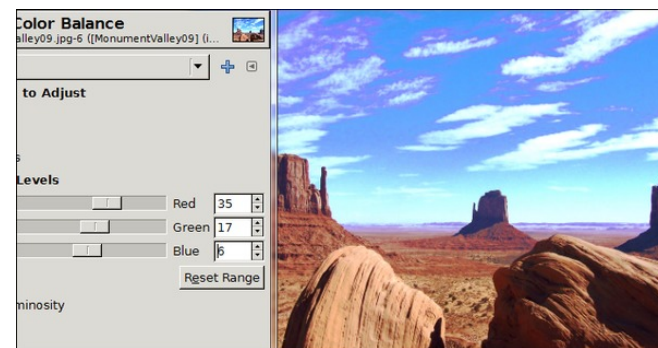


black in the histogram is to the left, then your image is probably dark and underexposed. If they're to the right, then the picture may be too bright. A perfect shot will show most of the action in the middle. Mine had some in the center but lots in the right. Grabbing the arrow on the right and moving it left cleared up my photo a little (bottom, left).

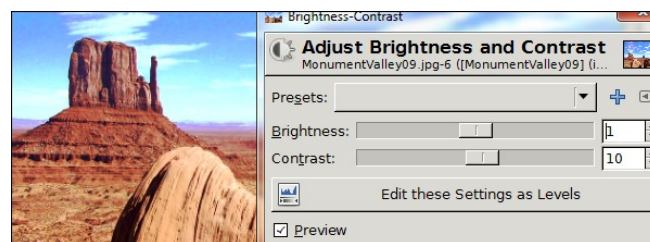
Now let's navigate to **Filters > Enhance > Sharpen**. The more you sharpen the image, the more detail you can see, but be careful! I went too far with the sharpen and almost had white detail spots on top of everything and white outlines around objects. Reducing the setting solved that problem.



I also used **Colors > Color Balance**. I had way too much brown in the image, and that location really isn't dull brown, but many shades of orange. Be careful with this one as well, or you can drastically change the color of everything. Obviously I put too much red into this one because the clouds now have some purple in them that wasn't there before. Fortunately, all that is needed is to reduce the red setting in the color balance window. The arrow keys are very useful, as they allow you to make very small adjustments to get the color you want (right, top).



Colors > Brightness and Contrast can also help to brighten up your image. Just make tiny changes.



I went back to **Color Balance** and then **Levels** and played around a bit more. I am happier with it, but I think I can do better. You can see the before and after photos on the next page.

It is much more colorful than the drab brown photo was. Make sure you keep the original of your photo and only work on a copy. Since every photo is different, there are no absolute settings that will work on everything. The best way is to experiment with the setting until it looks right to you. Don't forget you can undo each change (**<CTRL> + Z**) if you don't like it. With a little more practice, I'm sure I can get it right quicker ... and you can, too!



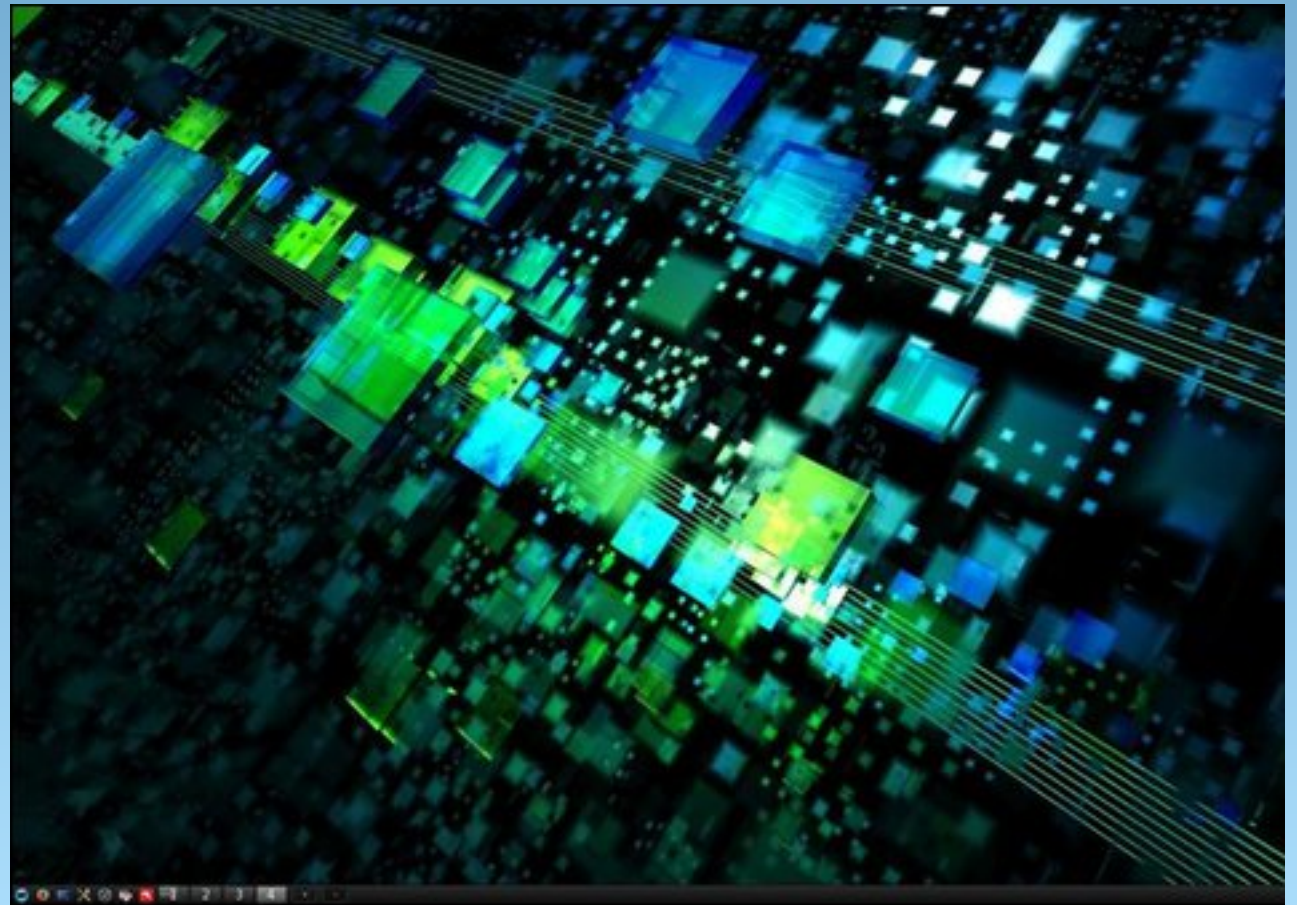
Before (above) and After (below)



Open Source Initiative



Screenshot Showcase



Posted by luikki, June 3, 2015, running KDE.



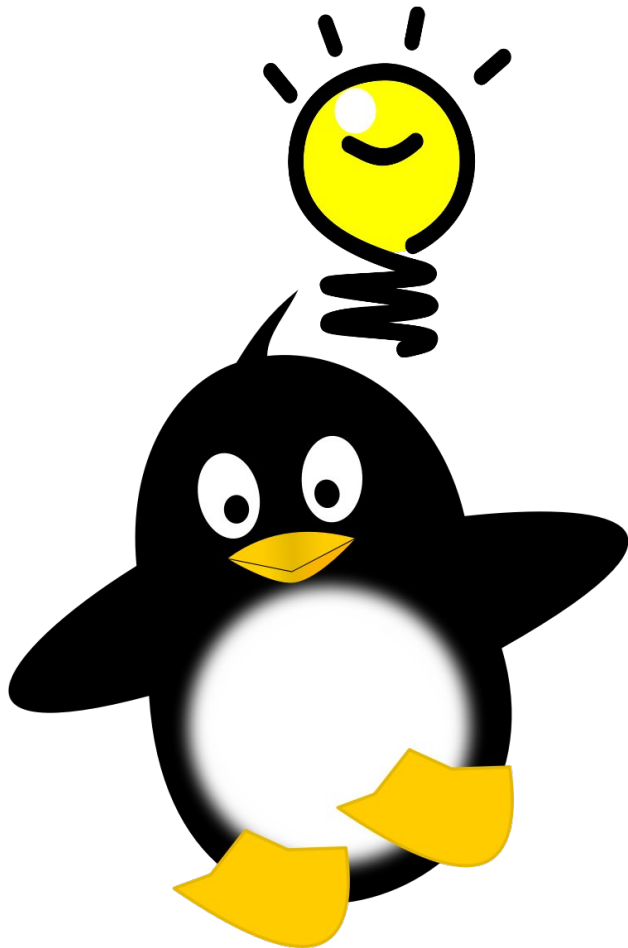
Visit.
Contribute.
Build.

The PCLinuxOS
Knowledge Base

It Belongs To YOU!

Tip Top Tips: Terminal With "Talking Cow"

Editor's Note: *Tip Top Tips* is a new monthly column in *The PCLinuxOS Magazine*. Each month, 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*.



This month's [Tip Top Tip](#) comes from PCLinuxOS forum user zen-subz.

Just to make my 'terminal' experience a bit enjoyable, I have done a bit of customization. These are the steps that I have followed:

1. Install fortune-mod and cowsay (from Synaptic).
2. Create a script with the following code, and name it cowscript.

```
#!/bin/bash
RANGE=5

number=$RANDOM
let "number %= $RANGE"
case $number in
  0)
    cow="default"
    ;;
  1)
    cow="small"
    ;;
  2)
    cow="tux"
    ;;
  3)
    cow="koala"
    ;;
  4)
    cow="moose"
    ;;
esac

RANGE=2
number=$RANDOM
let "number %= $RANGE"
case $number in
  0)
    command="/usr/bin/cowsay"
    ;;
  1)
    command="/usr/bin/cowthink"
```

```
;;
esac
/usr/games/fortune | $command -f $cow
```

3. Make it executable (chmod 755).
4. Place this 'cowscript' in /usr/local/bin.
5. Edit the .bashrc file and insert the word 'cowscript' in the last-line (without quotes). Save and exit.

Open your terminal(s), which will now look like this:



Without a doubt, the combined installation of cowsay and fortune-mod can make your terminal sessions much more fun, and even humorous. Heck ... either program is fun installed and run by itself, so linking them together like this can generate all kinds of laughs. So, let's dig a little deeper into these two command line programs. (Pssst! No one says you have to be serious all of the time!).

Fortune-mod does exactly what you would think it should do: it displays famous (or infamous) quotes and quips on your computer in your terminal window.

Combined with the cowsay program, it can make your cow sound quite witty and smart. The "fortunes" are stored in your /usr/share/games/fortunes/ directory. While some distros package fortune-mod with potentially "offensive" quotes and quips (typically stored in the /off subdirectory), the PCLinuxOS package is lacking those. Still, the **-a** command line switch will allow fortune-mod to display all quotes and quips, both offensive and non-offensive. At a command line prompt in a terminal session, simply entering **fortune -a** will display a random quote or quip in your terminal window. I recommend reading through the fortune-mod man page for more information, should you so choose. These basics will cover our needs for now.

Cowsay is responsible for drawing the fun ASCII pictures in your terminal window. Simply entering **cowsay "Say something"** will display the default cow saying the text in quotes. That's the simplest way to run cowsay. But, there is so much more you can do.

At a command line prompt, enter **cowsay -l** (small L). It will list out all of the various "cows" that are available. Below are the cows that are available on my computer.

```
apt beavis.zen bong bud-frogs bunny
calvin cheese cock cower daemon
default dragon dragon-and-cow elephant
elephant-in-snake eyes flaming-sheep
ghostbusters giraffe gnu head-in
hellokitty kenny kiss kitty koala kosh
luke-koala mech-and-cow meow milk
moofasa moose mutilated phaco pumpkin
ren satanic shark sheep skeleton small
small-duck sodomized sodomized-sheep
stegosaurus stimpys supermilker surgery
suse telebears three-eyes turkey turtle
tux udder vader vader-koala www
```

There is also a command line switch, **-f [name of cow]** text to display that is also supposed to allow you to specify a particular cow file to use. When used incorrectly – as in forgetting to specify text to display – your terminal will just sit there. You will

have to use Ctrl + C to break out of the endless loop you created. Your mileage may vary when using the cowsay command at a terminal prompt.

Depending on whether you want your cow to say something or think something, there are two valid commands to choose from. Using **cowsay** will cause your cow to "say" something, and is indicated by slashes leading from the cow image to the speech bubble. But using **cowthink** will cause your cow to think something, indicated by little "o" characters leading from the cow image to the speech bubble.

You can also specify special characters to use for the cow's eyes, by using the **-e** command line switch. The character string must be two characters, and the default is "oo" (without the quotes). If you choose to use the eye string, it must appear before the **-f [name of cow]** command line switch. Similarly, you can specify the character string to use for the cow's tongue, by using the **-T** command line switch. Just as with the command line switch to control the appearance of the cow's eyes, the tongue switch must be two characters long.

There are several provided modes which change the appearance of the cow, depending on its particular emotional/physical state. The **-b** option initiates Borg mode; **-d** causes the cow to appear dead; **-g** invokes greedy mode; **-p** causes a state of paranoia to come over the cow; **-s** makes the cow appear thoroughly stoned; **-t** yields a tired cow; **-w** is somewhat the opposite of **-t**, and initiates wired mode; **-y** brings on the cow's youthful appearance.

Custom Cows

Want your own "custom cows?" You can have those, as well. The cows are, after all, just ASCII art. Head on over to Christopher Johnson's ASCII Art Collection [website](#), and grab some existing (already created) ASCII art images. His website has been continuously online since 1994 and the days of Netscape 1.0. He has 23 different major categories

of ASCII art, with many of those categories further split down into subcategories. Are you a Trekker looking for some Star Trek ASCII art? There's a category for that (look under the Television category). Can't get enough of Star Wars, and want some Star Wars ASCII art? There's a category for that (look under the Movies category). Want some sports related ASCII art? There's a category for that.

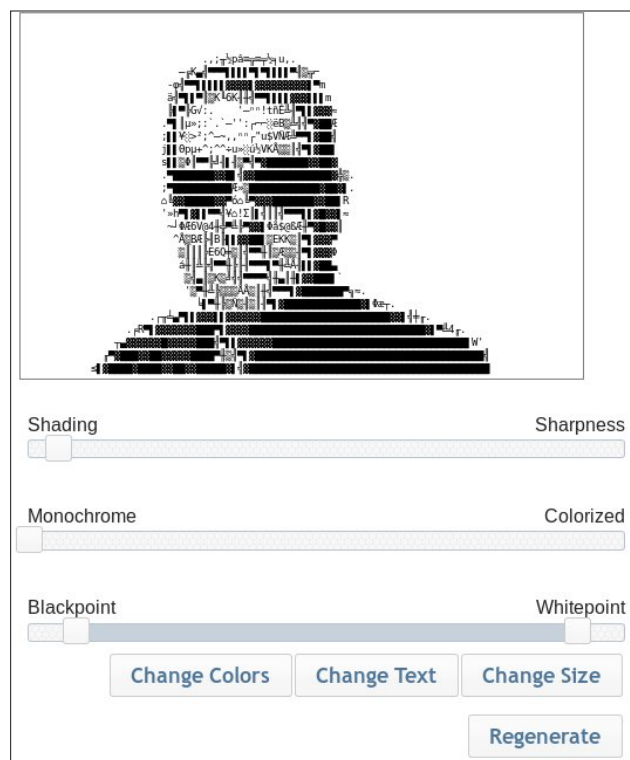
If you can't find what you're looking for there, you might want to try to create ASCII art from a real image. Visit the ASCII Art Generator [website](#). There, you can upload an image to be converted into ASCII art. Simple images without a lot of detail tend to work best, and the simpler the better. I visited the website and converted a picture of Texstar into ASCII art.

This ...



... became this (next page) ...





Don't hesitate to play with the settings on the website. Since we're using cowsay, we will need to create a monochrome ASCII art image. Moving the "whitepoint" slider towards the left makes the generated image lighter, and moving the "blackpoint" slider towards the right makes the generated image darker. I also found that having the top slider more towards the "Shading" end produces better ASCII art images for use with cowsay. After you've adjusted the settings, just click the "Regenerate" button to redraw your image. Don't worry ... the ASCII Art Generator runs right in your browser, and none of the images are stored on their server. Not even the images that you "upload." Once your ASCII art image is created to your satisfaction, simply copy the ASCII text from the website and paste it into your text editor.

If you're so inclined (and if you have the time), you can even create your own ASCII art from scratch. Just be sure to do it in a simple text editor, using a monospaced font.

Note that when you “grab” ASCII art images, you will have to copy the text, instead of right clicking and choosing “Save image as...” as you would do with “real” images, like with JPG and PNG files. Just save your ASCII art images as plain text files – for now.

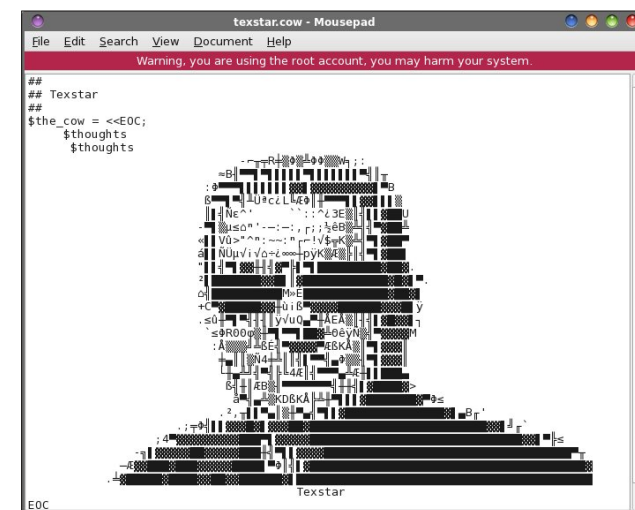
Next, as root, open one of the installed cow files on your computer. Don't worry ... if you installed cowsay from the PCLinuxOS repos, and they will be in the `/usr/share/cows` directory. Below is what the `kitty.cow` file looks like in a simple ASCII text editor.

```
##  
## A kitten of sorts, I think  
##  
$the_cow = <<EOC;  
    $thoughts  
        $thoughts  
            ("\' \' /") . _ _ _ _ _ " \' \'  
                * * ) _ _ _ _ _ ( ) _ _ _ _ _  
                ( Y _ ) _ _ _ _ _ ; _ _ _ _ _  
                    _ _ _ _ _ / _ _ _ _ _ , 4  
            ( i l ) , _ _ _ _ _ ( l i ) , _ _ _ _ _ ( ! _ _
```

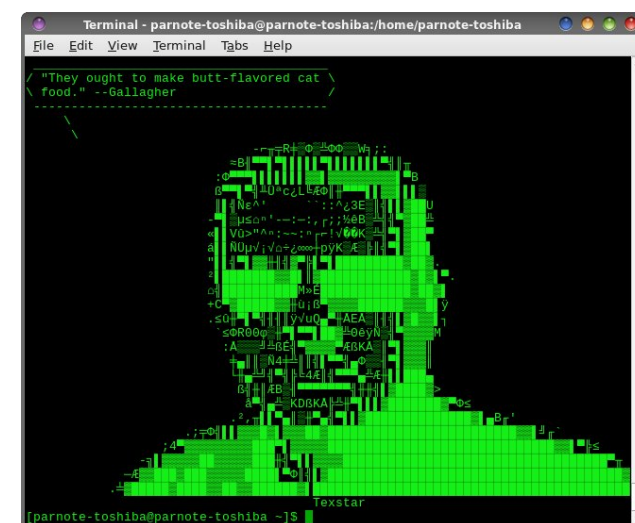
EOC

Simply replace the “text” between the second \$thoughts and the EOC line at the end with your ASCII art that you copied from the above website. You'll get a much better feel for what your ASCII art image will look like if you a) use a plain, ordinary text editor (Gedit, Kate, Mousepad, Leafpad, Geany, etc.) and b) if you use a monospaced font to display your text files (such as Monospace, Liberation Mono, Courier New, etc.). **DO NOT** use LibreOffice or any other word processor, such as Abiword or the one in the Calligra office suite, to edit or save your ASCII art image text!

Now, just resave your cow file, with its own unique name. For the image of Texstar, I called the file `texstar.cow`. Here's how the `texstar.cow` file appears in my text editor, Mousepad:



Here is how the `texstar.cow` file appears in my terminal:



Summary

See? Whoever said that working in a terminal session was boring never had cowsay and fortune-mod installed on their computer. Plus, with zen-subz combination of the two programs into one script, it combines both fun programs into a really fun time.

Inkscape Tutorial: Creating A Ketupat Icon

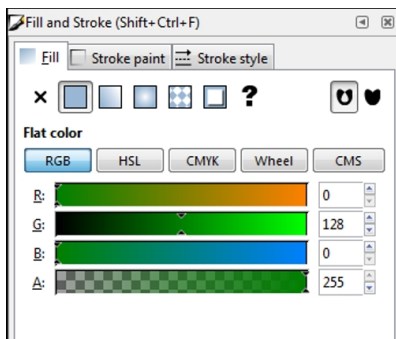
by Khadis

Here in Indonesia, Lebaran Day, which is celebrated with “ketupat” — cooked rice that is wrapped with coconut leaves—is only several days ahead. On this day, people usually send greeting cards—and of course, with a “ketupat” image or icon. In the image below you can see ketupat before it's cooked.

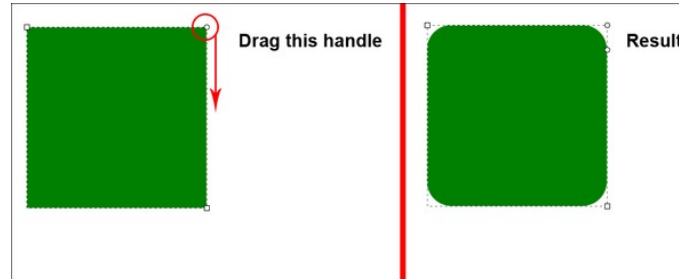


There are various techniques that we can apply to create the image or icon as realistic as possible. But now, I would like to use a simple technique to create a flat “ketupat” icon. Let's go!

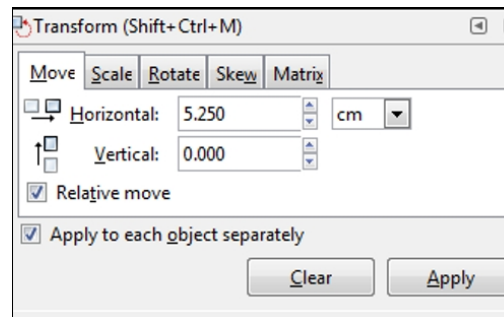
Open up your Inkscape and create a 5 cm x 5 cm rounded rectangle. Give it a dark green color (R: 0, G: 128, B: 0, A: 255) from the **Fill and Stroke (Shift + Ctrl + F)** panel. Outline is optional.



To create a rounded rectangle, just draw an ordinary rectangle, then modify it by dragging down the circle handle (top right corner of your active rectangle) using the **Edit path by nodes (F2)** tool.



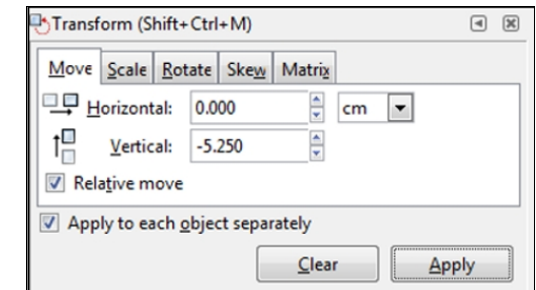
Duplicate the rectangle using the **Ctrl + D** command. After that, move this new rectangle through the **Transform (Shift + Ctrl + M)** panel. Set the parameters as shown below, then click Apply:



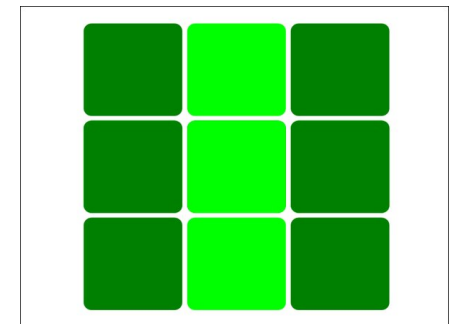
The setting above will move the duplicated rectangle 5 cm to the right and additional 0.25 cm for space.

Set the second rectangle with brighter green (R: 0, G: 255, B: 0, A: 255). Duplicate (again) the rectangle using **Ctrl + D** and return the color into darker green (R: 0, G: 128, B: 0, A: 255).

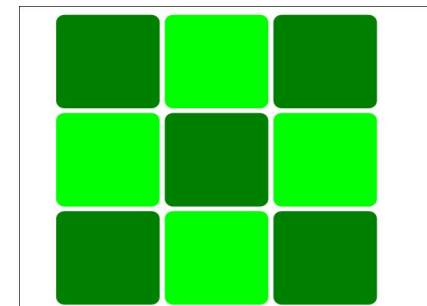
Select all three rectangles and group them (**Ctrl + G**). Then, duplicate and move them using the following parameter:



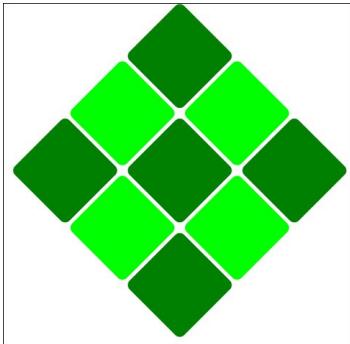
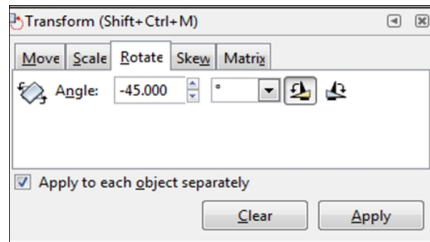
Repeat the previous step so you will get the following formation:



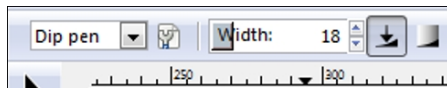
Change the color to get the following color combination. Don't forget to un-group the rectangles (**Ctrl + U**) first:



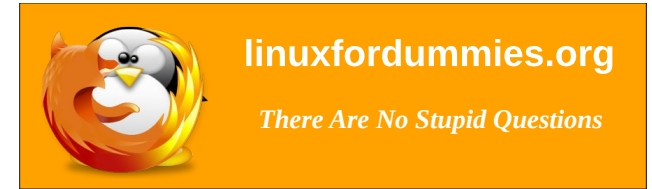
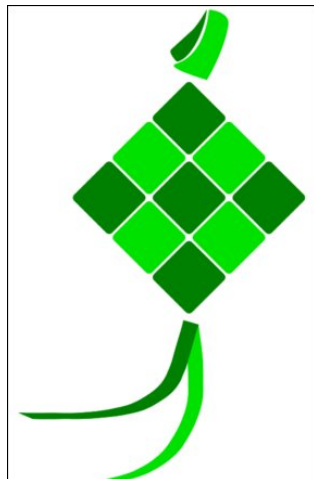
Select all the rectangles and group them (**Ctrl + G**), then rotate them -45 degree (rotating 45 degree will produce the same result as well).



Now, create the “head” and “tails” using the **Calligraphic tool (Ctrl + F6)**. Try to draw the “head” and the “tails” using a different parameter for each.



Final result:



Screenshot Showcase



Posted by billy7720, June 11, 2015, running MATE.

HTPC: Kodi Keyboard Shortcuts

by Paul Arnote (parnote)

Despite it's graphical nature, use of the keyboard makes Kodi much easier to control. Fortunately, Kodi allows the use of not only a predefined set of [keyboard shortcuts](#), but it also allows users to remap those keyboard assignments, via the the keymap editor add-on.

We'll cover the default keyboard shortcuts for Kodi here in this article. If you want to explore the use of the Kodi keymap editor, you can check [here](#) for more information. If you are interested in exploring the format and commands used to alter the Kodi keymaps, you can refer to [this](#) page (which will give you all the information you might possibly want to know about these topics).

As a bonus, the Kodi keymaps work not only for keyboards, but also universal remote controls, gamepads and joysticks. But, by far, the most common input device used by Kodi will be a plain computer keyboard and mouse.

Default Kodi Keyboard Shortcuts

The Kodi keyboard shortcuts are divided into three categories: Global, Video playback, and Music visualization. So, in many instances, the function of the keys will change, depending on what you might be currently doing in Kodi. We'll present the keyboard shortcuts in a list. If you want to view them as an HTML table, please check out the Kodi Wiki topic [here](#).

For brevity's sake, if a keyboard shortcut does not exist for one of the categories, that category will not be listed. When the keyboard shortcut is universal across all three categories, we'll just list "All" as the category. Similarly, when a key doesn't have any assigned function, <none> will appear next to that key character.

0 (zero)

Video Playback: Go between the last two channels (Live TV mode).

A

Video Playback: Audio delay control.

B

Global: Schedule recording timers (Live TV mode).
Video Playback: Schedule recording timers (Live TV mode).

C

Global: Contextual menu.
Video Playback: Playlist.

D

Global: Move item down (playlist editor & favorites window).

E

Global: Live TV EPG/TV guide.
Video Playback: Live TV EPG/TV guide.

F

All: Fast Forward.

G <none>

H

Global: Live TV channels window.
Video Playback: Live TV channels window.

I

All: Info.

J

Global: Live TV radio channels window.
Video Playback: Live TV radio channels window.

K

Global: Live TV recordings window.
Video Playback: Live TV recordings window.

L

Video Playback: Next subtitle.
Music Visualization: Lock preset.

M

All: Player controls (OSD)
Global: Move file (File manager).

N

Music Visualization: Current playlist window.

O

Video Playback: Codec info.
Music Visualization: Codec info.

P

Global: Play.
Video Playback: Play.
Music Visualization: Visualization preset list.

Q

Global: Queue.

R

All: Rewind.
Global: Rename (File manager).

S

Global: Shutdown menu.
Video Playback: Shutdown menu.
Music Visualization: Show preset.

Ctrl + S

All: Screenshot.

T

Video Playback: Toggles subtitles on and off.

Ctrl + T

Video Playback: Subtitle position control.

U

Global: Move item up (Playlist editor & Favorites window).

V

Video Playback: Teletext.
Music Visualization: Visualization settings.

W

Global: Mark as watched/unwatched.

X

All: Stop.

Y

All: Switch/choose player. For example, can be used to switch between internal player or a UPnP target player.

Z

Video Playback: Zoom/aspect ratio.

Space

All: Pause/Play.
Global: Current playlist window (video & music listings).
Highlight (File manager).

← (Left cursor arrow)

Global: Left.
Video Playback: Step back 30 seconds (videos), Previous channel group (Live TV).
Music Visualizations: Previous.

→ (Right cursor arrow)

Global: Right.
Video Playback: Step forward 30 seconds (videos), Next channel group (Live TV).
Music Visualizations: Next.

↑ (Up cursor arrow)

Global: Up.
Video Playback: Step forward 10 minutes, or next chapter (videos), Channel up (Live TV).
Music Visualization: Increase rating.

↓ (Down cursor arrow)

Global: Down.
Video Playback: Step back 10 minutes, or previous chapter (videos), Channel down (Live TV).

Ctrl + ← (Control + left cursor arrow)

Video Playback: Variable seek backwards.
Music Visualization: Variable seek backwards.

Ctrl + → (Control + right cursor arrow)

Video Playback: Variable seek forwards.
Music Visualization: Variable seek forwards.

Ctrl + ↑ (Control + up cursor arrow)

Video Playback: Move subtitles up.

Ctrl + ↓ (Control + down cursor arrow)

Video Playback: Move subtitles down.

PageUp

Global: Page up.

PageDown

Global: Page down.

Enter/Return

Global: Select
Video Playback: Player controls (OSD).
Music Visualization: Player controls (OSD).

Backspace

All: Back.

Esc

Global: Previous menu or Home screen.
Video Playback: Exit full screen.

. (Period)

Global: Skip forward.
Video Playback: Step forward 30 seconds.
Music Visualization: Skip forward.

, (Comma)

Global: Skip backward.
Video Playback: Step backward 30 seconds.
Music Visualization: Skip backward.

' (Quote)

Video Playback: Small step backwards seven (7) seconds.

Tab

All: Fullscreen playback.

PrtScn (PrintScreen)

All: Screenshot.

- (Minus)

All: Volume down.

+ (Plus)

All: Volume up.

= (Equal)

All: Volume down.

\ (Backslash)

All: Toggle between fullscreen/windowed mode.

[(Left bracket)

Video Playback: Step forward 10 minutes.

] (Right bracket)

Video Playback: Step backward 10 minutes.

Ctrl + End

Global: Exit Kodi (Home screen only).

Del

Global: Delete file (if enabled in settings).
Video Playback: Remove from playlist (Playlist editor only).
Music Visualization: Remove from playlist (Playlist editor only).

Home

Global: Jump to the top of the menu (..)

[numbers] then Return

Video Playback: Jump to that time in playback. (E.g., 1+2+3+4+return will jump to 12 minutes and 34 seconds.).
Music Visualization: Jump to that time in playback. (E.g., 1+2+3+4+return will jump to 12 minutes and 34 seconds.).

[numbers] then ←

Video Playback: Jump backwards in that amount of time. (E.g., 1+0+← will jump back 10 seconds.).
Music Visualization: Jump backwards in that amount of time. (E.g., 1+0+← will jump back 10 seconds.).

[numbers] then →

Video Playback: Jump forwards in that amount of time. (E.g., 1+0+← will jump forward 10 seconds.)

Music Visualization: Jump forwards in that amount of time. (E.g., 1+0+← will jump forward 10 seconds.)

Shift + Letter

Global: Jump to that letter in a list.

Kodi default mouse controls

Left Click on item	Select
Middle Click on item	Queue to playlist
Right Click on item	Context menu
Right Click on blank area	Back
Mouse Wheel Up	Up
Mouse Wheel Down	Down

Kodi picture slide show controls

0	Zoom Normal (100%)
1	Zoom Level 1
2	Zoom Level 2
3	Zoom Level 3
4	Zoom Level 4
5	Zoom Level 5
6	Zoom Level 6
7	Zoom Level 7
8	Zoom Level 8
9	Zoom Level 9
R	Rotate
T	Info (EXif data)
. period	Next picture
, comma	Previous picture
+ plus	Zoom in
- minus	Zoom out

Platform specific controls for Windows, Live and Linux

Win (Windows key): Contextual menu

Summary

As you can see, you can control most – if not all – that Kodi does, simply from a keyboard. Married with a mouse, you have complete control over Kodi. It's doubtful that anyone will use all of the keyboard shortcuts available for Kodi, but it's nice to know they are there if you need them. It's just another reason why Kodi has developed such broad appeal among HTPC users, across all platforms.



PCLinuxOS Family Member Spotlight: The Chief

As Told To Smileeb

This month we have The Chief



How old are you?

I was 75 in June 2015.

Married, single or what?

Married since November 1959

Children, grandchildren?

2 daughters, 3 granddaughters, 2 grandsons

Retired or working and for how long and at what.

Retired from U.S. Navy as a Senior Chief Aviation Fire Control Technician in 1980, retired from L-3 communications as a Software Engineer in 1999, went back on the payroll in 2004 and retired again in 2006 (thus doubling my pension).

What is the area you live in like. Weather, Quietness, Scenery.

I live in the southeast U.S. (Georgia). We definitely have 4 seasons, but none are usually very severe. Worst is the high humidity in the summer.

Are you handy with your hands and have any hobbies.

I have had many hobbies over the years - model airplanes, slot cars, for example, but now I'm down to only three: my truck, photography and target shooting.



What is your education level?

I am a 10th grade dropout, with a GED and 2 years of college (Auburn University) and 1 year of night classes (Georgia State University) - no degree.

Do you like to travel, go camping?

Travel: yes, camping: no. But at my age and with a semi-invalid wife, don't get to do much traveling.

What caused you to try Linux and join this forum?

I had some slight familiarity with Linux from my work as a Software Engineer. Once I retired, I decided to explore the possibilities and soon stumbled upon PCLinuxOS. It was like a breath of fresh air, and I was hooked.



Colt SAA 1873

Browning Buckmark





Colt 1911A

PCLinuxOS Family Member Spotlight is an exclusive, monthly column by smileeb, featuring PCLinuxOS forum members. 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 smileeb in the PCLinuxOS forum expressing your interest.



A magazine just isn't a magazine without articles to fill the pages.

If you have article ideas, or if you would like to contribute articles to the PCLinuxOS Magazine, send an email to:
pclinuxos.mag@gmail.com

We are interested in general articles about Linux, and (of course), articles specific to PCLinuxOS.



Linux Docs
Linux Man Pages



Screenshot Showcase



Posted by agmg, June 22, 2015, running KDE.

Game Zone: Slinki

by daiashi



About The Game

Slinki is a side view 2.5D action-platformer where your own prosthetic arm is the ultimate multipurpose weapon. Use it as a bladed boomerang to control the population of mutated critters, pull levers towards you to get through puzzles and zip-line across grappling surfaces. All this to find out what happened to the once peaceful forest you used to live in.

Plot

Slinki's once peaceful and smooth-running home has gone horribly wrong, with every denizen gone crooked and twisted, and irrevocably hostile. Himself affected and warped as well, Slinki can just sit there and waste away, or take the last of his strength to the Forest's Core, and end the misery whatever ill-bred rascal thought to fester in it's depths.

Features

- Eliminate your enemies with your unique bladed boomerang
- Have your boomerang return to you or propel yourself towards it
- More than one level of hot bladed arm action
- Fight against several unique forest critters

- Old-school gaming with a score counter, a timer and lives system
- Unlock the Hardcore mode for added challenge
- Controller support

System requirements:

Fully updated PCLinuxOS and Steam

Hardware:

OS: PCLinuxOS

Processor: 1 Ghz dual core processor

Graphics: OpenGL 2.1 compatible GPU

Memory: 512 MB RAM

Hard Drive: 600 MB available space

Sound Card: Any Sound Card

About The Company

Titan Forged Games is a digital games company from Coimbra, Portugal. They're just starting out but they hope you'll enjoy their games!

History

In the early days of 2008, Nuno Barreto, a little Portuguese boy with the dream of someday making a Spider-Man videogame, had enrolled in a BSC in Computer Engineering. He was accompanied by Pedro Caetano, another young fool already set on sacrificing his life to the gods of videogaming.

Together, they decided to make their own game studio. Soon after, another misled "youth," David Fial, asked to join, to help out with odds and ends. The three agreed with the daring enterprise, and joined forces. The studio was officially formed in 2012, and the first game was planned, named "Brian Storm." The team evolved with the abduction of Sérgio Alves from his peaceful existence, to lead the artistic side of the endeavour, and everything was set in motion.

Some Gameplay Screenshots





Getting It To Run

Install Steam (if you don't have it installed already), then start it. You will need to create a new account, if you do not already have one. Once you have Steam up and running, go to the store tab. Click on the

Linux tab if you wish and search for Slinky. Click on and download the demo. If you have updated your system, including graphics drivers, you should be good to go.

Slinky



*The PCLinuxOS
Magazine*

*Created with
Scribus*

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 ...

PCLinuxOS Full Monty ... Everything you might want or need – plus the kitchen sink!

Testimonial: My Thoughts About PCLinuxOS

by LinuxBorn

Well, I feel this topic's name is so good to introduce ourselves to tell about our first thoughts about PCLinuxOS. I am not opening the same topic name. I hope I will not upset anyone.

By the way, this is my first post here. Let's start, shall we?

I had tried a few other Linux distribution And I like PCLinuxOS for being UNIQUE as STABLE throughout its timeline.

I have been using it for two years now. And I must say it's mostly ROBUST. Some 64-bit software don't always run well out of the Synaptic box. Some are missing from the repository. This is because I do use my Linux more than a regular desktop. Otherwise, it's really good for a simple desktop every day tasks. Beyond that, you will have to put your hand in CLI (Terminal) & config files.

* What I love with PCLinuxOS is that there is no need to reinstall the whole things every end-of-life. For it has none.

* Plus the fact that it was inspired by MANDRAKE 9.2 and it's cheering me a lot, for that specific Distro was my first ever on the Linux World. It's like a return back to past (2003) in our FUTURE (2015+). PCLinuxOS is MUCH more POWERFUL than what I remember about the non-powerpack Mandrake Linux at that time.

* What I like less is that some UPSTREAM updated software to their latest version are not always stable or reliable while using add-ons like with GIMP & VLC. We have to wait for add-ons we are using to

get updated as well. We do not have that choice to go back or pin to the previous version for a limited time until add-ons are upgraded. (Or I do not get the way to do that.)

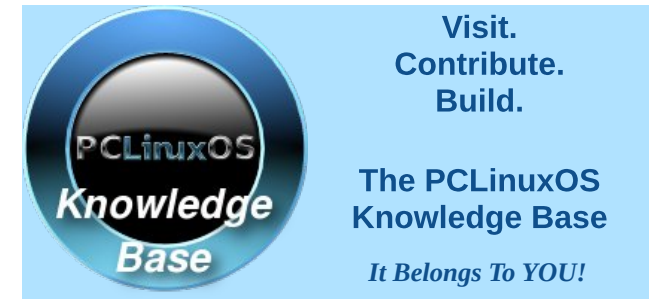
* Sometimes apps are just not working with all functions (Lynx was not integrating GOPHER protocol last year). Some are not launching at all (for some are looking for libraries that are not in the right path, **especially for X86_64 version**) like BIND9, others rarely put their shortcut in the Xfce menu.



I'm not criticizing at all these things, for I signed up here to help and give feedback with what issues I can go through in order to see if any other **64-bit** users have the same issues, and how we can fix them for the benefit of all. I'm doing that in the right forum places after deep searches at first hand.

Giving a **BIG HUG** to ALL the Community that is hanging out here.

And **A WARM HELLO** to **ALL!**

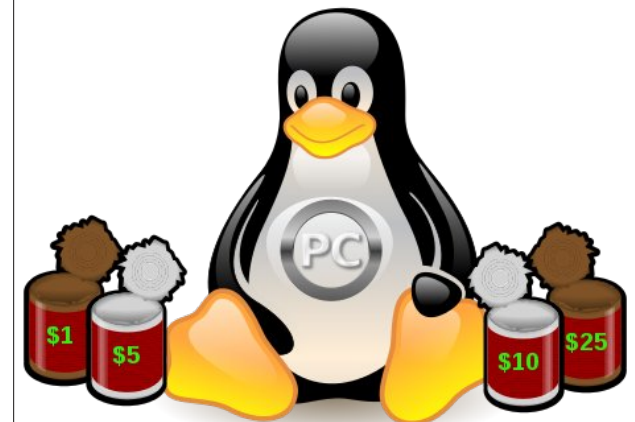


Donate To PCLinuxOS

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

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

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



International Community
PCLinuxOS Sites



secure
private
simple-to-use
PCLinuxOS-Cloud
Sign up TODAY! pclosusers.com/services-signup.php

Screenshot Showcase



Posted by Hertz, June 13, 2015, running KDE.

PCLinuxOS Puzzled Partitions

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

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

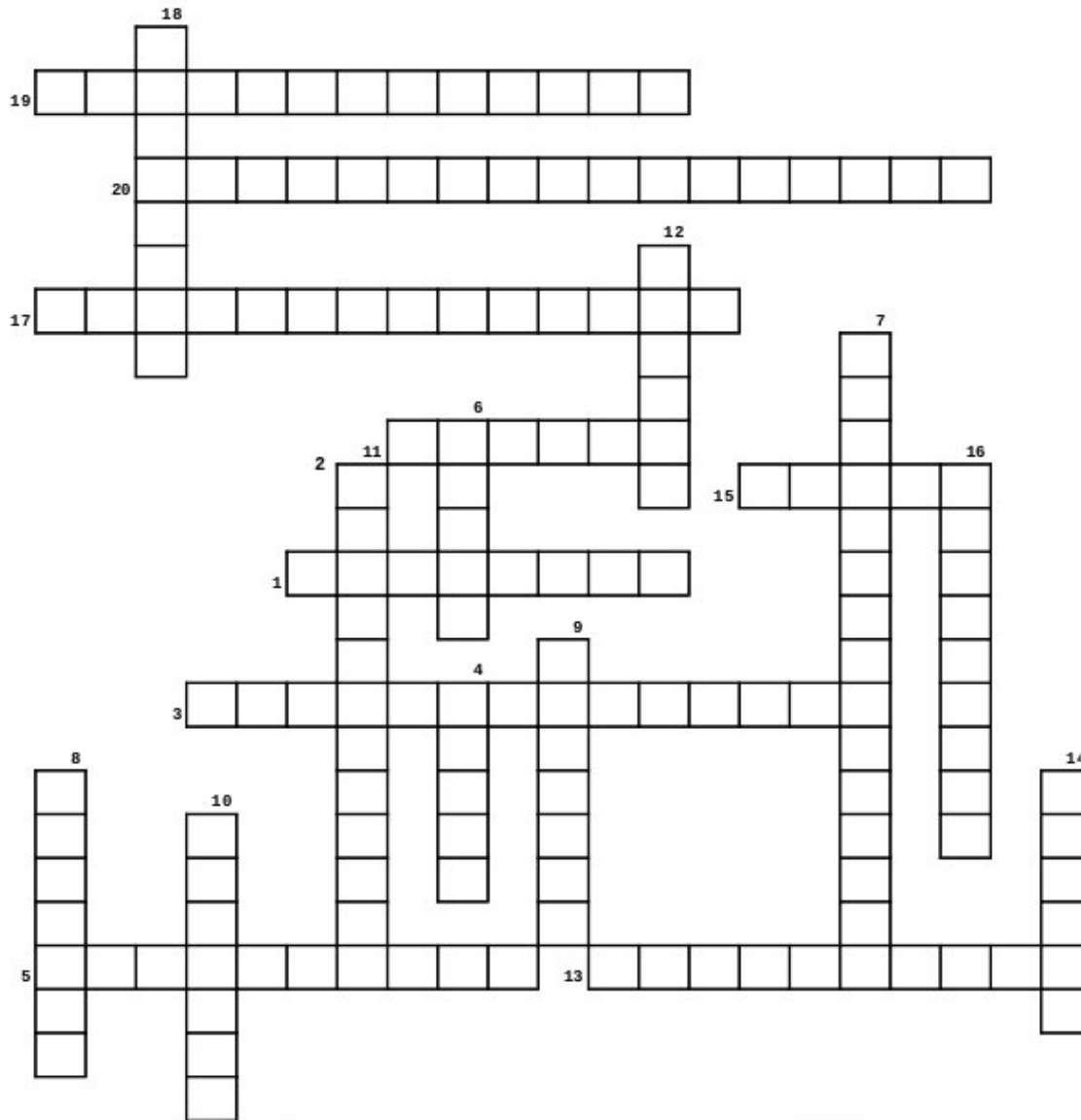


Possible score 239, average score 167.

Download Puzzle Solutions Here

PCLinuxOS Crossword Puzzle: June 2015

Around The Yard



1. Fun to watch the birds splash
2. If your yard is big enough, you can put one in the ground
3. Arrange these so everyone enjoys the conversation
4. May need this to keep the neighbors out
5. Fun for everyone to jump
6. Most people don want to see these
7. Nice place to pick your favorite salad
8. Nice to keep some plants confined
9. Fruit trees
10. Enjoy swinging while you nap in this
11. Outside window shades
12. Put your vehicle and gardening tools here
13. You need this to water and wash
14. Alternative to a fence
15. A bit of shade from these is always nice
16. To water the yard or for the kids to play
17. Kids like to shoot hoops
18. Snoopy sleeps on top!
19. Arrangement of beautiful blooms
20. Watch the birds drink here

[Download Puzzle Solutions Here](#)

Around The Yard Word Find

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

awning
 back porch
 backyard
 basketball goal
 birdbath
 bushes
 doghouse
 fence
 flowers
 flower garden
 front porch
 fruit tree
 garage
 garden hose
 grass
 hammock
 hedges
 herb garden
 hummingbird feeder
 lawnmower
 orchard
 patio
 patio furniture
 planter
 porch
 shrubs
 sprinkler
 stepping stones
 swimming pool
 swing
 terrace
 trampoline
 trees
 vegetable garden
 weeds

[Download Puzzle Solutions Here](#)

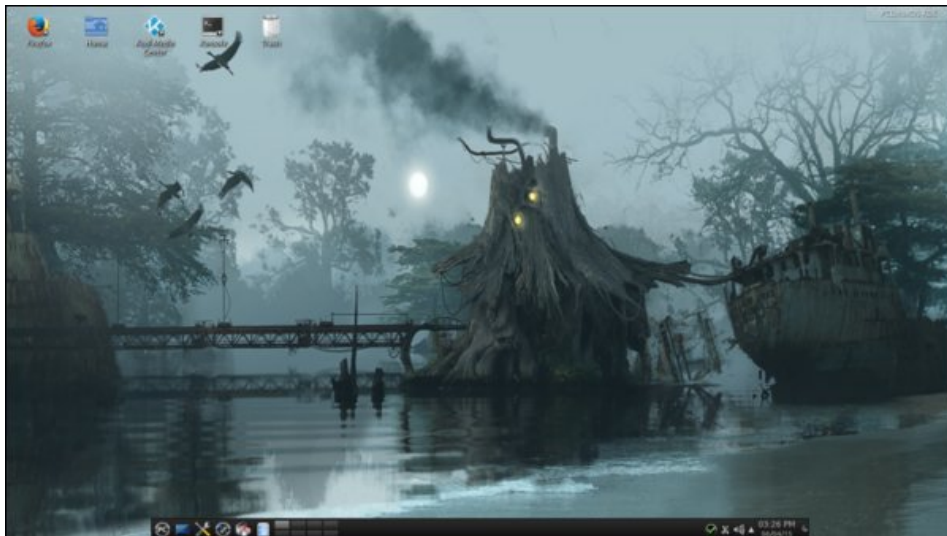
More Screenshot Showcase



Posted by parnote, June 23, 2015, running Xfce.



Posted by present_arms, June 3, 2015, running KDE.



Posted by chilly, June 4, 2015, running KDE.



Posted by weirdwolf, June 2, 2015, running LXDE.