

# Conky 1.10 - Conky lua Horloge, Caps/Num Lock, Mail, Clementine

- Objet : conky par l'exemple
- Niveau requis :  
[débutant](#), [avisé](#)
- Commentaires : *page à piller.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande](#), tout commence là ! 😊



- prenez soin d'éditer les fichiers pour les adapter à votre système

## Conky lua Horloge, Caps/Num Lock, Mail, Clementine



Fichier .conkyrc :

[.conkyrc](#)

```
conky.config = {  
  --## Begin Window Settings #####  
  own_window = true,  
  own_window_type = 'normal',  
  own_window_transparent = true,  
  own_window_hints = 'undecorated,sticky,below,skip_taskbar,skip_pager',  
  own_window_colour = '#000000',  
  own_window_class = 'Conky',  
  own_window_title = 'Sidepanes Conky',  
  --## ARGB can be used for real transparency  
  --## NOTE that a composite manager is required for real transparency.  
  --## This option will not work as desired (in most cases) in  
  conjunction with  
  --## own_window_type normal  
  -- own_window_argb_visual yes # Options: yes or no  
  -- own_window_argb_visual = true,  
  
  --## When ARGB visuals are enabled, this use this to modify the alpha  
  value  
  --## Use: own_window_type normal  
  --## Use: own_window_transparent no  
  --## Valid range is 0-255, where 0 is 0% opacity, and 255 is 100%  
  opacity.  
  -- own_window_argb_value 50
```

```
minimum_width = 300,  
minimum_height = 670,  
maximum_width = 300,  
  
alignment = 'top_right',  
gap_x = 15,--## left | right  
gap_y = 5,--## up | down  
--##### End Window Settings ###  
--## Font Settings #####  
-- Use Xft (anti-aliased font and stuff)  
use_xft = true,  
font = 'Roboto Regular:size=10',  
  
-- Alpha of Xft font. Must be a value at or between 1 and 0 ###  
xftalpha = 1,  
-- Force UTF8? requires XFT ###  
override_utf8_locale = true,  
  
uppercase = false,  
--##### End Font Settings ###  
--## Colour Settings #####  
draw_shades = false,--yes  
default_shade_color = 'black',  
  
draw_outline = false,-- amplifies text if yes  
default_outline_color = 'black',  
  
default_color = '#d8dee8',-- White  
color1 = '#bf616a',-- Red  
color2 = '#a3be8c',-- Green  
color3 = '#ebcb8b',-- Yellow  
color4 = '#81a1c1',-- Blue  
--##### End Colour Settings ###  
--## Borders Section #####  
draw_borders = false,  
-- Stippled borders?  
stippled_borders = 0,  
-- border margins  
border_inner_margin = 0,  
border_outer_margin = 0,  
-- border width  
border_width = 0,  
-- graph borders  
draw_graph_borders = true,--no  
--default_graph_size 15 40  
--##### End Borders Section ###  
--## Miscellaneous Section #####  
-- Boolean value, if true, Conky will be forked to background when  
started.
```

```
background = true,
-- Adds spaces around certain objects to stop them from moving other
things
-- around, this only helps if you are using a mono font
-- Options: right, left or none
use_spacer = 'none',

-- Default and Minimum size is 256 - needs more for single commands
that
-- "call" a lot of text IE: bash scripts
--text_buffer_size 6144

-- Subtract (file system) buffers from used memory?
no_buffers = true,

-- change GiB to G and MiB to M
short_units = true,

-- Like it says, it pads the decimals on % values
-- doesn't seem to work since v1.7.1
pad_percents = 2,

-- Imlib2 image cache size, in bytes. Default 4MiB Increase this value
if you use
-- $image lots. Set to 0 to disable the image cache.
imlib_cache_size = 0,

-- Use the Xdbe extension? (eliminates flicker)
-- It is highly recommended to use own window with this one
-- so double buffer won't be so big.
double_buffer = true,

-- Maximum size of user text buffer, i.e. layout below TEXT line in
config file
-- (default is 16384 bytes)
-- max_user_text 16384

-- Desired output unit of all objects displaying a temperature.
Parameters are
-- either "fahrenheit" or "celsius". The default unit is degree
Celsius.
-- temperature_unit Fahrenheit
--##### End Miscellaneous Section ###

update_interval = 1,

lua_load = '~/.conky/clock_rings.lua',
lua_draw_hook_pre = 'clock_rings',

};
```

```
conky.text = [[

${voffset 8}${color 1B708D}${font caviar dreams:size=16}${time
%A}${font}${voffset -8}${alignr 0}${color FFFFFFF}${font caviar
dreams:bold:size=38}${time %e}${font}
${color FFFFFFF}${voffset -30}${color FFFFFFF}${font caviar
dreams:size=18}${time %b}${font}${voffset -3} ${color FFFFFFF}${font
caviar dreams:size=20}${time %Y}${font}${color 1B708D}${hr}
${image ~/.conky/debian-logo.png -p 155,150 -s 40x40}
${execPi 2 ~/.conky/caps_num.sh}
${voffset 90}

${image ~/.conky/icones/weather.png -p 10,255 -s 72x72}
${offset 75} Météo
${voffset 5}${offset 75} Aujourd'hui : ${execi 1800
~/.conky/sidepanes/res/weather}
${voffset 5}${offset 25}${font Roboto Regular:size=8}${execi 1800
~/.conky/sidepanes/res/weather -t}

${font Roboto Regular:size=10}
${image ~/.conky/icones/wifi.png -p 10,361 -s 72x72}
${offset 75} Wifi
${voffset 5}${offset 75} ${wireless_essid}

${image ~/.conky/icones/email.png -p 10,438 -s 72x72}
${offset 75} Email
${voffset 5}${offset 75} ${execi 600 ~/.conky/sidepanes/res/email} non
lu

${image ~/.conky/icones/infos.png -p 10,510 -s 72x72}
${offset 75} Uptime: ${uptime_short}
${offset 75} Processes: ${processes}
${offset 75} Running: ${running_processes}

${image ~/.conky/icones/system_debian.png -p 10,597 -s 72x72}
${offset 75} ${exec cat /etc/issue.net} $machine
${offset 75} Kernel: ${kernel}

${image ~/.conky/icones/music.png -p 10,664 -s 72x72}
${offset 75} Musique
${voffset 5}${offset 75} ${color b5f9ff}${voffset 2}${goto 91}${execi
10 ~/.conky/sidepanes/res/clementine artist}
${voffset 45} ${goto 40}${execi 10 ~/.conky/sidepanes/res/clementine
title}

${image ~/.conky/cover.jpg -p 220, 690 -s 65x65 -f 10}
${execi 10 ~/.conky/sidepanes/res/clementine cover}]
```

```
]];
```

Fichier Clock\_rings.lua :

[clock\\_rings.lua](#)

```
--[[
Clock Rings by Linux Mint (2011) reEdited by despot77

This script draws percentage meters as rings, and also draws clock
hands if you want! It is fully customisable; all options are described
in the script. This script is based off a combination of my clock.lua
script and my rings.lua script.

IMPORTANT: if you are using the 'cpu' function, it will cause a
segmentation fault if it tries to draw a ring straight away. The if
statement on line 145 uses a delay to make sure that this doesn't
happen. It calculates the length of the delay by the number of updates
since Conky started. Generally, a value of 5s is long enough, so if you
update Conky every 1s, use update_num>5 in that if statement (the
default). If you only update Conky every 2s, you should change it to
update_num>3; conversely if you update Conky every 0.5s, you should use
update_num>10. ALSO, if you change your Conky, is it best to use
"killall conky; conky" to update it, otherwise the update_num will not
be reset and you will get an error.

To call this script in Conky, use the following (assuming that you save
this script to ~/scripts/rings.lua):
    lua_load ~/scripts/clock_rings.lua
    lua_draw_hook_pre clock_rings

Changelog:
+ v1.0 -- Original release (30.09.2009)
  v1.1p -- Jpope edit londonali1010 (05.10.2009)
*v 2011mint -- reEdit despot77 (18.02.2011)
]]

settings_table = {
    {
        -- Edit this table to customise your rings.
        -- You can create more rings simply by adding more elements to
        settings_table.
        -- "name" is the type of stat to display; you can choose from
        'cpu', 'memperc', 'fs_used_perc', 'battery_used_perc'.
        name='clock',
        -- "arg" is the argument to the stat type, e.g. if in Conky you
        would write ${cpu cpu0}, 'cpu0' would be the argument. If you would not
        use an argument in the Conky variable, use ''.
        arg='heure',
        -- "max" is the maximum value of the ring. If the Conky
```

```
variable outputs a percentage, use 100.
    max=12,
    -- "bg_colour" is the colour of the base ring.
    bg_colour=0xffffffff,
    -- "bg_alpha" is the alpha value of the base ring.
    bg_alpha=0.1,
    -- "fg_colour" is the colour of the indicator part of the ring.
    fg_colour=0x1B708D,
    -- "fg_alpha" is the alpha value of the indicator part of the
ring.
    fg_alpha=0.2,
    -- "x" and "y" are the x and y coordinates of the centre of the
ring, relative to the top left corner of the Conky window.
    x=180, y=170,
    -- "radius" is the radius of the ring.
    radius=50,
    -- "thickness" is the thickness of the ring, centred around the
radius.
    thickness=5,
    -- "start_angle" is the starting angle of the ring, in degrees,
clockwise from top. Value can be either positive or negative.
    start_angle=0,
    -- "end_angle" is the ending angle of the ring, in degrees,
clockwise from top. Value can be either positive or negative, but must
be larger than start_angle.
    end_angle=360
},
{
    name='clock',
    arg='minutes',
    max=60,
    bg_colour=0xffffffff,
    bg_alpha=0.1,
    fg_colour=0x1B708D,
    fg_alpha=0.4,
    x=180, y=170,
    radius=56,
    thickness=5,
    start_angle=0,
    end_angle=360
},
{
    name='clock',
    arg='secondes',
    max=60,
    bg_colour=0xffffffff,
    bg_alpha=0.1,
    fg_colour=0x1B708D,
    fg_alpha=0.6,
    x=180, y=170,
```

```
radius=62,  
thickness=5,  
start_angle=0,  
end_angle=360  
},  
{  
    name='time',  
    arg='%d',  
    max=31,  
    bg_colour=0xffffffff,  
    bg_alpha=0.1,  
    fg_colour=0x1B708D,  
    fg_alpha=0.8,  
    x=180, y=170,  
    radius=70,  
    thickness=5,  
    start_angle=-90,  
    end_angle=90  
},  
{  
    name='time',  
    arg='%m',  
    max=12,  
    bg_colour=0xffffffff,  
    bg_alpha=0.1,  
    fg_colour=0x1B708D,  
    fg_alpha=1,  
    x=180, y=170,  
    radius=76,  
    thickness=5,  
    start_angle=-90,  
    end_angle=90  
},  
}  
  
-- Use these settings to define the origin and extent of your clock.  
  
clock_r=65  
  
-- "clock_x" and "clock_y" are the coordinates of the centre of the  
clock, in pixels, from the top left of the Conky window.  
  
clock_x=180  
clock_y=170  
  
show_seconds=true  
  
require 'cairo'  
  
function rgb_to_r_g_b(colour,alpha)  
    return ((colour / 0x10000) % 0x100) / 255., ((colour / 0x100) %
```

```
0x100) / 255., (colour % 0x100) / 255., alpha
end

function draw_ring(cr,t,pt)
    local w,h=conky_window.width,conky_window.height

    local
xc,yc,ring_r,ring_w,sa,ea=pt['x'],pt['y'],pt['radius'],pt['thickness'],
pt['start_angle'],pt['end_angle']
    local bgc, bga, fg, fga=pt['bg_colour'], pt['bg_alpha'],
pt['fg_colour'], pt['fg_alpha']

    local angle_0=sa*(2*math.pi/360)-math.pi/2
    local angle_f=ea*(2*math.pi/360)-math.pi/2
    local t_arc=t*(angle_f-angle_0)

    -- Draw background ring

    cairo_arc(cr,xc,yc,ring_r,angle_0,angle_f)
    cairo_set_source_rgba(cr,rgb_to_r_g_b(bgc,bga))
    cairo_set_line_width(cr,ring_w)
    cairo_stroke(cr)

    -- Draw indicator ring

    cairo_arc(cr,xc,yc,ring_r,angle_0,angle_0+t_arc)
    cairo_set_source_rgba(cr,rgb_to_r_g_b(fg,fga))
    cairo_stroke(cr)
end

function draw_clock_hands(cr,xc,yc)
    local secs,mins,hours,secs_arc,mins_arc,hours_arc
    local xh,yh,xm,ym,xs,ys

    secs=os.date("%S")
    mins=os.date("%M")
    hours=os.date("%I")

    secs_arc=(2*math.pi/60)*secs
    mins_arc=(2*math.pi/60)*mins+secs_arc/60
    hours_arc=(2*math.pi/12)*hours+mins_arc/12

    -- Draw hour hand

    xh=xc+0.7*clock_r*math.sin(hours_arc)
    yh=yc-0.7*clock_r*math.cos(hours_arc)
    cairo_move_to(cr,xc,yc)
    cairo_line_to(cr,xh,yh)

    cairo_set_line_cap(cr,CAIRO_LINE_CAP_ROUND)
```



```
cairo_set_line_width(cr,5)
cairo_set_source_rgba(cr,1.0,1.0,1.0,1.0)
cairo_stroke(cr)

-- Draw minute hand

xm=xc+0.85*clock_r*math.sin(mins_arc)
ym=yc-0.85*clock_r*math.cos(mins_arc)
cairo_move_to(cr,xc,yc)
cairo_line_to(cr,xm,ym)

cairo_set_line_width(cr,3)
cairo_stroke(cr)

-- Draw seconds hand

if show_seconds then
    xs=xc+clock_r*math.sin(secs_arc)
    ys=yc-clock_r*math.cos(secs_arc)
    cairo_move_to(cr,xc,yc)
    cairo_line_to(cr,xs,ys)

    cairo_set_line_width(cr,1)
    cairo_stroke(cr)
end
end

function conky_clock_rings()
    local function setup_rings(cr,pt)
        local secs, mins, hours, mins_secs, hours_mins
        local str=''
        local value=0

        if pt['name']=='clock' then
            secs=os.date("%S")
            mins=os.date("%M")
            hours=os.date("%I")

            mins_secs=mins+secs/60
            hours_mins=hours+mins/60
            if hours_mins >= 12 then hours_mins=hours_mins-12 end
            if pt['arg']=="heure" then
                str=hours_mins
            elseif pt['arg']=="minutes" then
                str=mins_secs
            else
                str=secs
            end
        else
            str=string.format('${%s %s}',pt['name'],pt['arg'])
            str=conky_parse(str)
        end
    end
end
```

```
end

value=tonumber(str)

if value==nil then -- Gestion du problème de séparateur
décimale
    str=conky_parse(str):gsub("%.",".")
    value=tonumber(str)
end

if value == nil then value = 0 end
pct=value/pt['max']

draw_ring(cr,pct,pt)
end

-- Check that Conky has been running for at least 5s

if conky_window==nil then return end
local
cs=cairo_xlib_surface_create(conky_window.display,conky_window.drawable
,conky_window.visual, conky_window.width,conky_window.height)

local cr=cairo_create(cs)

local updates=conky_parse('${updates}')
update_num=tonumber(updates)

if update_num>5 then
    for i in pairs(settings_table) do
        setup_rings(cr,settings_table[i])
    end
end

draw_clock_hands(cr,clock_x,clock_y)
end
```

Fichier caps\_num.sh :

caps\_num.sh

```
#!/bin/bash
caps=$(xset -q | grep -c "Caps Lock:  on")
num=$(xset -q | grep -c "Num Lock:  on")

function bottom_align(){
    if [ $caps -gt 0 ]; then
        echo "\${color red} Maj\${color}"
    else
```

```
        echo " min"
    fi
    if [ $num -gt 0 ]; then
        echo "\${color 6495ee} Num${color}"
    else
        echo " nv"
    fi
}

function top_align(){
    if [ $caps -gt 0 ]; then
        echo -n "\${color red} Maj\${color}"
    else
        echo -n " min "
    fi
    if [ $num -gt 0 ]; then
        echo "\${color 6495ee} Num${color}"
    else
        echo " nv"
    fi
}

#check alignment
alignbl=$(grep "alignment bottom_left" ~/.conkyrc|grep -c "#")
alignbr=$(grep "alignment bottom_right" ~/.conkyrc|grep -c "#")
if [ $alignbl -gt 0 ] && [ $alignbr -gt 0 ]; then
    top_align
else
    bottom_align
fi
```

Fichier météo :

Fichier email :

Fichier Clementine :

Icones :

Logo Debian au centre de la pendule (debian-logo.png):



From:

<http://debian-facile.org/> - Documentation - Wiki

Permanent link:

<http://debian-facile.org/utilisateurs:jeremix:config:conky-1.10-lua-horloge-caps-num-lock-mail-clementine>

Last update: 01/11/2020 09:09

