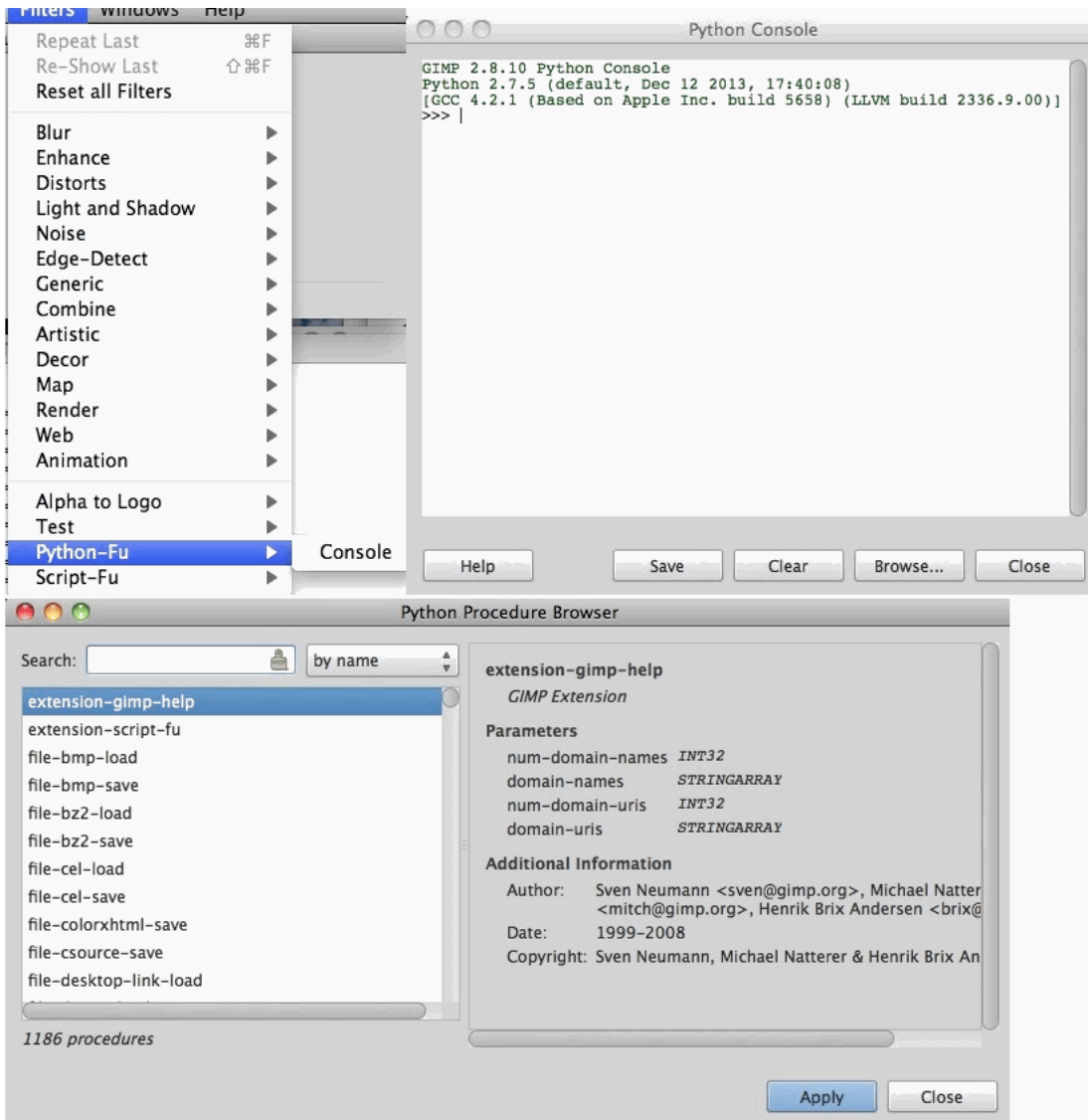
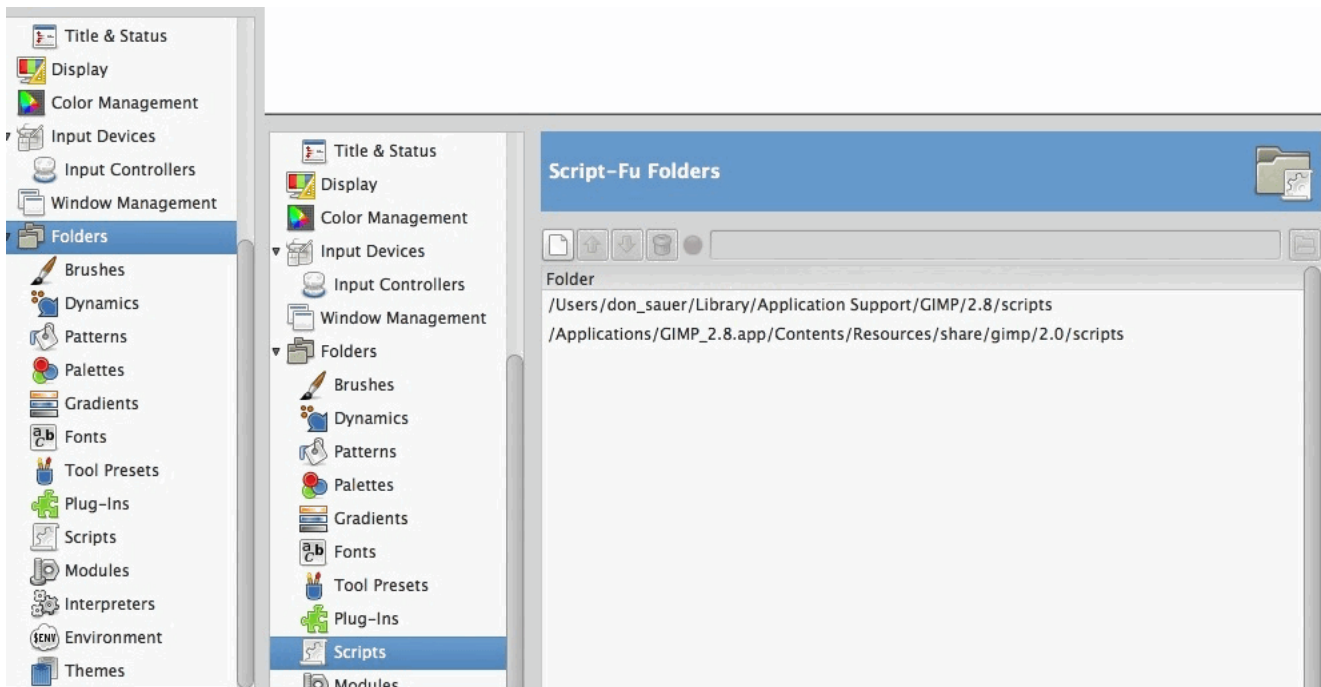


# GimpPython\_CopyPaste\_Templates

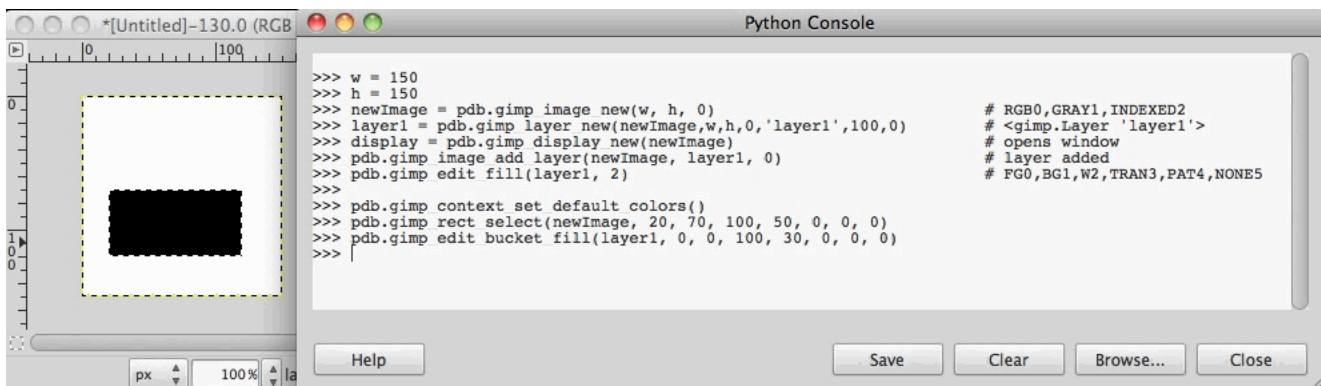
## =====CONSOLE\_AND\_BROWSER=====



## =====FOLDERS=====



## =====COPY\_AND\_PASTE INTO\_CONSOLE\_EXAMPLE=====



```

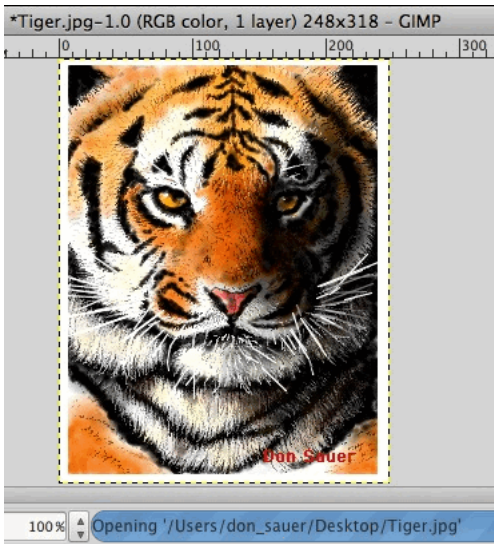
w = 150
h = 150
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_edit_fill(layer1, 2)

pdb.gimp_context_set_default_colors()
pdb.gimp_rect_select(newImage, 20, 70, 100, 50, 0, 0, 0)
pdb.gimp_edit_bucket_fill(layer1, 0, 0, 100, 30, 0, 0, 0)
  
```

## =====IMAGE\_PROCESSING=====

```

pdb.gimp_display_delete(display)      # Optional delete previous image
  
```



```
filename = "/Users/don_sauer/Desktop/Tiger.jpg"
raw_filename = "Tiger.jpg"
new_image = pdb.gimp_file_load(filename, raw_filename)
new_layer = pdb.gimp_file_load_layer(new_image, filename)
display = pdb.gimp_display_new(new_image)

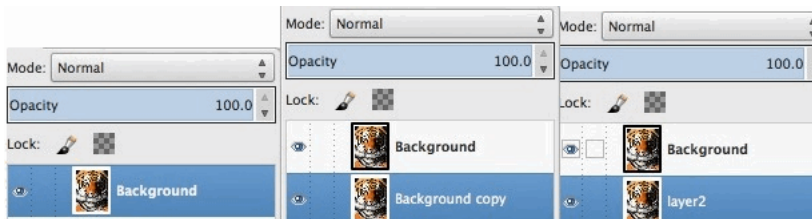
base_type = pdb.gimp_image_base_type(new_image)
drawable = pdb.gimp_image_get_active_drawable(new_image)
active_layer = pdb.gimp_image_get_active_layer(new_image)
this_filename = pdb.gimp_image_get_filename(new_image)
height = pdb.gimp_image_height(new_image)
width = pdb.gimp_image_width(new_image)
xresolution, yresolution = pdb.gimp_image_get_resolution(new_image)
unit = pdb.gimp_image_get_unit(new_image)
filename = pdb.gimp_image_get_filename(new_image)
name = pdb.gimp_image_get_name(new_image)
num_images, image_ids = pdb.gimp_image_list()
```

```
# 0 {RGB(0),GRAY(1,INDEXED(2))}
# <gimp.Layer 'Background'>
# <gimp.Layer 'Background'>
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 318
# 248
# (72.0, 72.0)
# 1
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 'Tiger.jpg'
# (1, (5,))
```



```
pdb.gimp_image_select_rectangle(new_image, 0, 10,10, 200, 300)
this_selection = pdb.gimp_image_get_selection(new_image)
is_selection = pdb.gimp_item_is_selection(this_selection)
pdb.gimp_selection_all(new_image)
pdb.gimp_selection_none(new_image)
```

```
#
# <gimp.Channel 'Selection Mask'>
# 1 means true
#
```



```
layer1 = pdb.gimp_image_get_layer_by_name(new_image, 'Background')
layer2 = pdb.gimp_layer_copy(layer1, TRUE)
pdb.gimp_image_add_layer(new_image, layer2, 1)
active_layer = pdb.gimp_image_get_active_layer(new_image)
pdb.gimp_layer_set_name(active_layer, 'layer2')
layer2
layer1
name = pdb.gimp_layer_get_name(layer2)
pdb.gimp_layer_add_alpha(layer2)
mode = pdb.gimp_layer_get_mode(layer2)
```

```
# <gimp.Layer 'Background'>
# layer2 copy of layer + alpha
# adds layer below
# <gimp.Layer 'Background copy'>
# rename
# <gimp.Layer 'layer2'>
# <gimp.Layer 'Background'>
# 'layer2'
# add alpha if has none
# 0 Normal
```

```

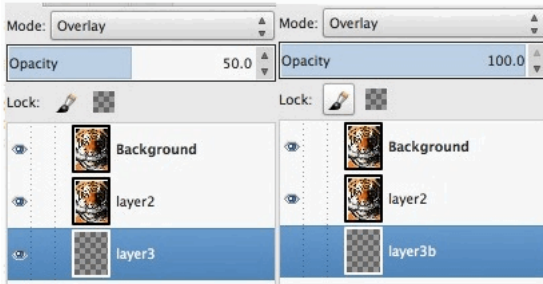
opacity = pdb.gimp_layer_get_opacity(layer2)
visible = pdb.gimp_layer_get_visible(layer2)
is_layer = pdb.gimp_drawable_is_layer(layer2)
is_selection = pdb.gimp_item_is_selection(layer2)
position = pdb.gimp_image_get_layer_position(new_image, layer1)
position = pdb.gimp_image_get_layer_position(new_image, layer2)
num_layers, layer_ids = pdb.gimp_image_get_layers(new_image)

```

```

# 100.0 100%
# 1 true
# 1
# 0
# 0
# 1
# 2, (2, 6)

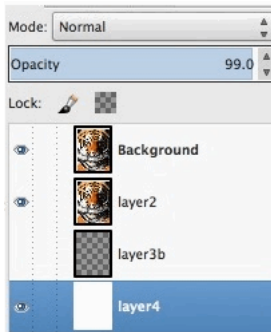
```



```

layer3 = pdb.gimp_layer_new(new_image,width,height,1,'layer3',50,5) # 1 = RGBA <gimp.Layer 'layer4'>
pdb.gimp_image_add_layer(new_image, layer3, 2) # add below Transparent
opacity = pdb.gimp_layer_get_opacity(layer3) # 50.0 50%
mode = pdb.gimp_layer_get_mode(layer3) # 5 Overlay
pdb.gimp_layer_set_name(layer3, 'layer3b') # rename
pdb.gimp_layer_set_opacity(layer3, 100)
pdb.gimp_layer_set_visible(layer3, 0)

```



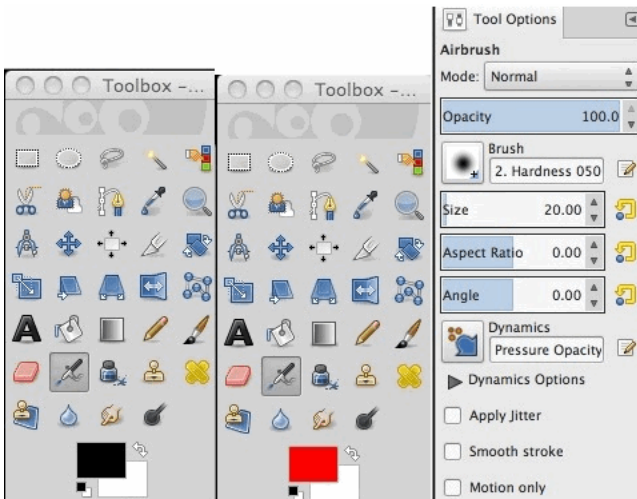
```

layer4 = pdb.gimp_layer_new(new_image,width,height,0,'layer4',99,0) # 0 = RGBA <gimp.Layer 'layer4'>
pdb.gimp_image_add_layer(new_image, layer4, 3) # adds black layer
pdb.gimp_invert(layer4) # makes from black to white
name = pdb.gimp_image_get_name(new_image) # 'Tiger.jpg'
image = pdb.gimp_item_get_image(layer4) # <gimp.Image 'Tiger.jpg'>

name = pdb.gimp_item_get_name(layer2) # 'layer2'

is_selection = pdb.gimp_item_is_selection(layer4) # 0
is_drawable = pdb.gimp_item_is_drawable(layer4) # 1
is_layer = pdb.gimp_item_is_layer(layer4) # 1
is_layer_mask = pdb.gimp_item_is_layer_mask(layer4) # 0
is_text_layer = pdb.gimp_item_is_text_layer(layer4) # 0
is_valid = pdb.gimp_item_is_valid(layer4) # 1
is_dirty = pdb.gimp_image_is_dirty(new_image) # 1
is_valid = pdb.gimp_image_is_valid(new_image) # 1

```



```

background_color = pdb.gimp_context_get_background()
foreground_color = pdb.gimp_context_get_foreground()
pdb.gimp_context_set_foreground((1.0, 0.0, 0.0, 1.0))
name_active_brush = pdb.gimp_context_get_brush()

```

```

# gimpcolor.RGB(1.0, 1.0, 1.0, 1.0)
# gimpcolor.RGB(0.0, 0.0, 0.0, 1.0)
# set foreground to red
# '2. Hardness 050'

```



```

size = pdb.gimp_context_get_brush_size() # 20.0
name, width, height, spacing = pdb.gimp_brushes_get_brush() # ('2. Hardness 050', 51, 51, 10)
nam,op,sp,mod,w,h,len,mask = pdb.gimp_brushes_get_brush_data(name) # ('2. Hardness 050',1.0,10,0,51,51,2601,(0, 0,...
opacity = pdb.gimp_brushes_get_opacity() # 100.0
paint_mode = pdb.gimp_brushes_get_paint_mode() # 0
# NORMAL0,DISSOLVE1,BEHIND2,MULTIPLY3,SCREEN4,OVERLAY5,DIFFERENCE6,ADDITION7,SUBTRACT8,DARKEN-ONLY9,
# LIGHTEN-ONLY10,HUE11,SATURATION12,COLOR13,VALUE14,DIVIDE15,DODGE16,BURN17,HARDLIGHT18,SOFTLIGHT19,
# GRAIN-EXTRACT20,GRAIN-MERGE21,COLOR-ERASE22,ERASE23,REPLACE24,ANTI-ERASE25
spacing = pdb.gimp_brushes_get_spacing() # 10

```



```

pdb.gimp_image_set_active_layer(new_image, layer1) #
color = pdb.gimp_image_pick_color(new_image, layer1,120,150,0,1,3) # gimpcolor.RGB(0.91,0.42,0.059,1.0)
has_alpha = pdb.gimp_drawable_has_alpha(layer1) # 0
pdb.gimp_layer_add_alpha(layer1)
has_alpha = pdb.gimp_drawable_has_alpha(layer1) # 1
color = pdb.gimp_color_picker(new_image, layer1, 120,150,0,1,3) # gimpcolor.RGB(0.94,0.42 0.059,1.0)
num_channels, pixel = pdb.gimp_drawable_get_pixel(layer1, 120, 150) # (3, (227, 108, 8))
is_gray = pdb.gimp_drawable_is_gray(layer1) # 0
is_indexed = pdb.gimp_drawable_is_indexed(layer1) # 0
is_rgb = pdb.gimp_drawable_is_rgb(layer1) # 1
type = pdb.gimp_drawable_type(layer1) # 0 RGB0,RGBA1,GRY2,GRYA3,INDX4,INDXA5
width = pdb.gimp_drawable_width(layer1) # 248
pdb.gimp_image_select_color(new_image, 0, layer1, color) # exact color
pdb.gimp_by_color_select(layer1, color, 50, 0, 0, 0, 3, 0) # 50 threshold radius 3
pdb.gimp_selection_grow(new_image, 10) # grow select by 10
pdb.gimp_selection_shrink(new_image, 5) # shrink select by 5
pdb.gimp_selection_invert(new_image)
non_empty, x1, y1, x2, y2 = pdb.gimp_selection_bounds(new_image) # (1, 0, 0, 248, 318)
this_selection = pdb.gimp_image_get_selection(new_image) # <gimp.Channel 'Selection Mask'>
is_selection = pdb.gimp_item_is_selection(this_selection) # 1 means true
is_empty = pdb.gimp_selection_is_empty(new_image) # 0

```



```

pdb.gimp_selection_sharpen(new_image)
pdb.gimp_selection_translate(new_image, 50, 70)
pdb.gimp_selection_clear(new_image) # unselect

```



```

pdb.gimp_layer_set_visible(layer1, 0)
pdb.gimp_layer_set_visible(layer2, 1)
pdb.gimp_image_set_active_layer(new_image, layer1)
pdb.gimp_image_set_active_layer(new_image, layer2)
layer2 = pdb.gimp_image_get_active_layer(new_image)
pdb.gimp_image_select_rectangle(new_image, 0, 30,30, 100, 50)
pdb.gimp_edit_bucket_fill(layer2, 0, 0, 100, 0, 0, 0,0)

pdb.gimp_selection_clear(new_image)
pdb.gimp_image_select_rectangle(new_image, 0, 30,100, 100, 50)
pdb.gimp_edit_fill(layer2, 2)
pdb.gimp_selection_clear(new_image)
pdb.gimp_drawable_fill(layer2, 3)

pdb.gimp_layer_set_visible(layer1, 1)
pdb.gimp_image_set_active_layer(new_image, layer1)
pdb.gimp_invert(layer1)

```

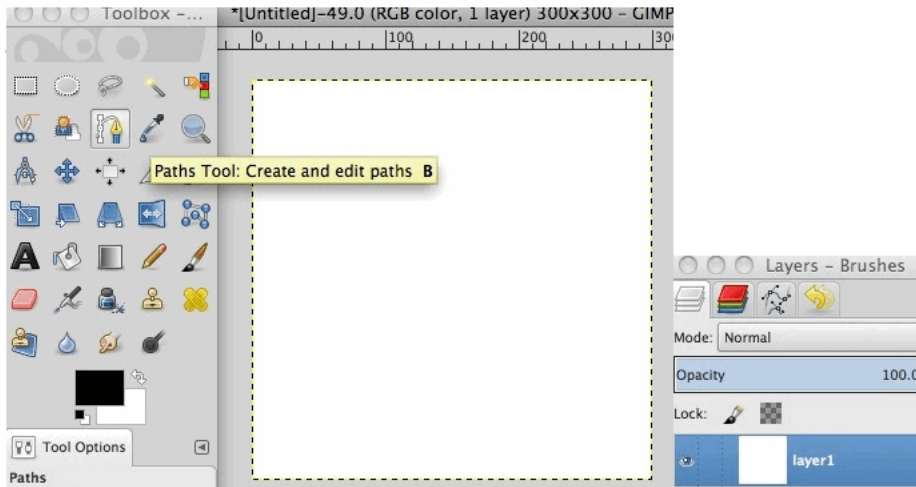
# <gimp.Layer 'layer2'>  
#  
# unslect  
#  
# FOR0,BACK01,WHI-2,TRANS3,PAT4,NONE5  
# unslect  
# FOR0,BACK01,WHI-2,TRANS3,PAT4,NONE5

## =====Vectors=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



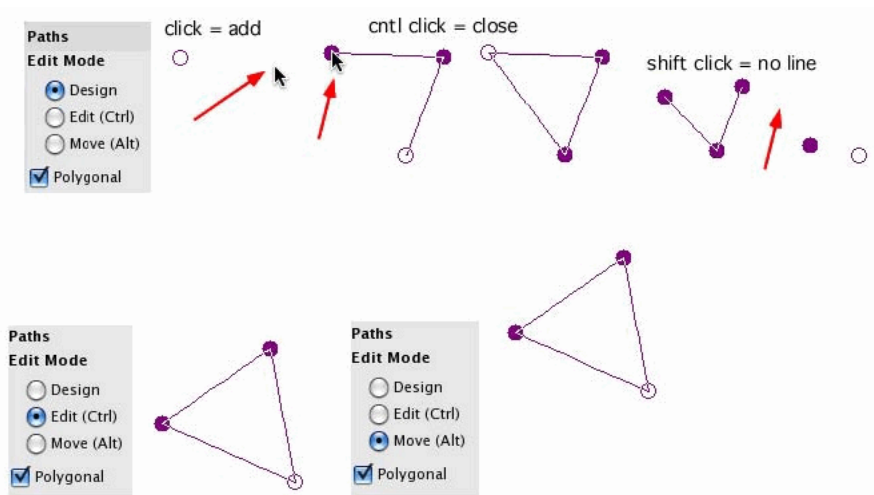
## # USE CODE BELOW TO CREATE A IMAGE AND LAYER, THEN USE PATH TOOL

```

w = 300
h = 300
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_edit_fill(layer1, 2)
pdb.gimp_context_set_default_colors()

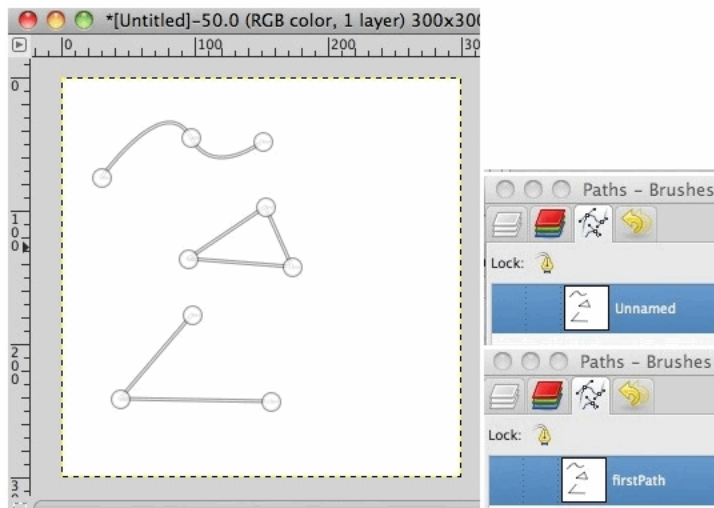
```

# RGB0,GRAY1,INDEXED2  
# <gimp.Layer 'layer1'>  
# opens window  
# layer added  
# FG0,BG1,W2,TRAN3,PAT4,NONE5



# DRAW SOME PATHS (cntl=command on Mac) drag creates curves

```
activeVectors = pdb.gimp_image_get_active_vectors(newImage) # <gimp.Vectors 'Unnamed'>
name = pdb.gimp_vectors_get_name(activeVectors) # 'Unnamed'
pdb.gimp_vectors_set_name(activeVectors, "firstPath")
```



```
num_strokes, stroke_ids = pdb.gimp_vectors_get_strokes(activeVectors) # (3, (1, 2, 3)) THREE strokes 1 TO 3
type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[0])
#(0,18,(18.0,91.0,30.0,75.0,42.0,59.0,80.0,13.0,97.0,45.0,114.0,77.0,151.0,48.0,151.0,48.0,151.0,48.0),0)

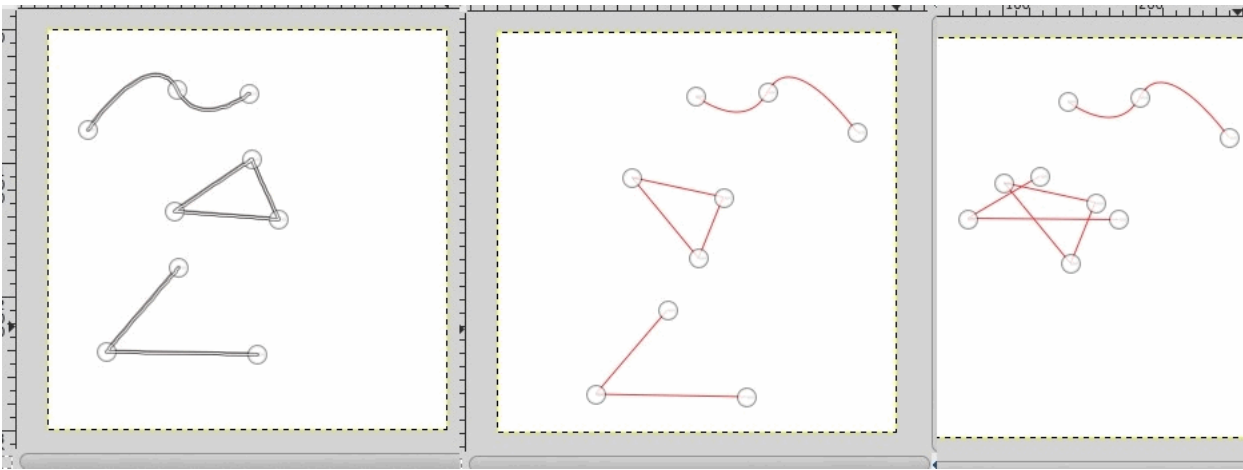
type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[1])
#(0,18,(95.0,136.0,95.0,136.0,95.0,136.0,153.0,97.0,153.0,97.0,153.0,97.0,173.0,142.0,173.0,142.0,173.0,142.0),1)

type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[2])
#(0,18,(98.0,178.0,98.0,178.0,98.0,178.0,44.0,241.0,44.0,241.0,44.0,241.0,157.0,243.0,157.0,243.0,157.0,243.0),0)

controlpoints[3] # 178.0

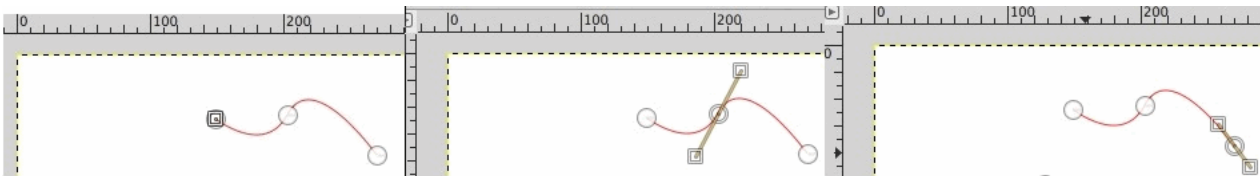
position = pdb.gimp_image_get_vectors_position(newImage, activeVectors) #0
pdb.gimp_vectors_set_visible(activeVectors, 1)
```





```
length = pdb.gimp_vectors_stroke_get_length(activeVectors, stroke_ids[0], 3) # 151.6713715341169
pdb.gimp_vectors_stroke_flip(activeVectors, stroke_ids[0], 0, 150) # flip_HO,V1, axis
pdb.gimp_vectors_stroke_translate(activeVectors, stroke_ids[2], 30, 30)
pdb.gimp_vectors_stroke_rotate(activeVectors, stroke_ids[1], 130, 130, 45.0)
pdb.gimp_vectors_stroke_scale(activeVectors, stroke_ids[2], 1.0, 0.5)

x,y,slope,valid = pdb.gimp_vectors_stroke_get_point_at_dist(activeVectors, stroke_ids[0],50,3) #(235.01,39.61,0.721,1)
```

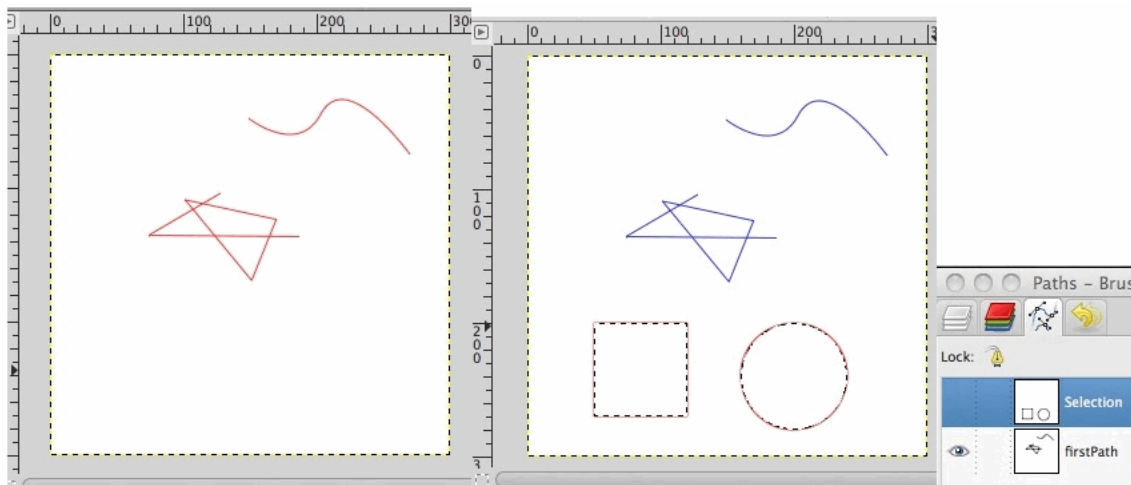


```
type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[0])
# (0,18,(282,91,270,75,258,59,220,13,203,45,186,77,149,48,149,48,149,48),0)
```

```
fpath = "/Users/don_sauer/Downloads/firstPath"
pdb.gimp_vectors_export_to_file(newImage, fpath, activeVectors)
vstring = pdb.gimp_vectors_export_to_string(newImage, activeVectors) # svg file format

pdb.gimp_image_remove_vectors(newImage, activeVectors)
merge = 0
scale = 0
num_vectors, vectors_ids = pdb.gimp_vectors_import_from_file(newImage, fpath, merge, scale)

activeVectors = pdb.gimp_image_get_active_vectors(newImage) # <gimp.Vectors 'firstPath'>
pdb.gimp_vectors_set_visible(activeVectors, 1)
```

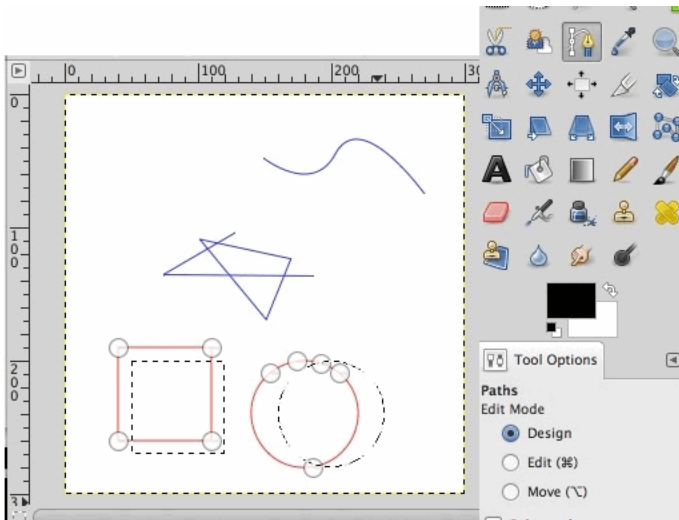


```
pdb.gimp_image_select_rectangle(newImage, 0, 50,200, 70, 70)
pdb.gimp_image_select_ellipse(newImage, 0, 160, 200, 80, 80)
pdb.plugin_sel2path(newImage, layer1)
activeVectors = pdb.gimp_image_get_active_vectors(newImage) # <gimp.Vectors 'Selection'>
pdb.gimp_vectors_set_visible(activeVectors, 1)

num_strokes, stroke_ids = pdb.gimp_vectors_get_strokes(activeVectors) # (2, (1, 2)) Two strokes number 1 TO 2

type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[0])
# (0,24,(120,200,120,200,120,200,120,270,120,270,50,270,50,270,50,200,50,200,50,200),1)

type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[1])
# (0,30,(206,200,212,202,216,203,222,206,226,209,252,231,240,274,206,280,166,286,144,235,174,209,180,204,187,
# 202,194,200,200,200),1)
```



```

pdb.gimp_selection_none(newImage)
pdb.gimp_vectors_to_selection(activeVectors, 0, 0, 1, 0, 0)
pdb.gimp_vectors_stroke_translate(activeVectors, stroke_ids[0], -10, -10)
pdb.gimp_vectors_stroke_translate(activeVectors, stroke_ids[1], -20, 0)

```

```

=====

?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
    "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">

<svg xmlns="http://www.w3.org/2000/svg"
    width="4.16667in" height="4.16667in"
    viewBox="0 0 300 300">
  <path id="firstPath"
    fill="none" stroke="black" stroke-width="1"
    d="M 270.00,75.00
      C 258.00,59.00 220.00,13.00 203.00,45.00
      186.00,77.00 149.00,48.00 149.00,48.00M 101.37,108.99
      C 101.37,108.99 169.72,123.62 169.72,123.62
      169.72,123.62 151.24,169.27 151.24,169.27
      151.24,169.27 101.37,108.99 101.37,108.99 Z
      M 128.00,104.00
      C 128.00,104.00 74.00,135.50 74.00,135.50
      74.00,135.50 187.00,136.50 187.00,136.50" />
</svg>

```

## =====**CutPaste**=====

```

pdb.gimp_display_delete(display)          # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)

drawable = pdb.gimp_image_get_active_drawable(newImage)
activeLayer = pdb.gimp_image_get_active_layer(newImage)
this_filename = pdb.gimp_image_get_filename(newImage)
height = pdb.gimp_image_height(newImage)
width = pdb.gimp_image_width(newImage)
xresolution, yresolution = pdb.gimp_image_get_resolution(newImage)
unit = pdb.gimp_image_get_unit(newImage)
filename = pdb.gimp_image_get_filename(newImage)
name = pdb.gimp_image_get_name(newImage)

# <gimp.Layer 'Background'>
# <gimp.Layer 'Background'>
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 318
# 248
# (72.0, 72.0)
# 1
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 'Tiger.jpg'

```



```

layer1 = pdb.gimp_image_get_layer_by_name(newImage, 'Background') # <gimp.Layer 'Background'>

layer2 = pdb.gimp_layer_new(newImage,width,height,0,'layer2',100,0) # 0 = RGBA <gimp.Layer 'layer4'>
pdb.gimp_image_add_layer(newImage, layer2, 3) # adds black layer

pdb.gimp_image_select_rectangle(newImage, 0, 50,50, 100, 100) #
non_empty = pdb.gimp_edit_copy(layer1)

floatingSel = pdb.gimp_selection_float(layer1, 50, 50)
pdb.gimp_floating_sel_attach(floatingSel, layer1)

```

## =====PasteOtherLayers=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)

drawable = pdb.gimp_image_get_active_drawable(newImage) # <gimp.Layer 'Background'>
activeLayer = pdb.gimp_image_get_active_layer(newImage) # <gimp.Layer 'Background'>
this_filename = pdb.gimp_image_get_filename(newImage) # '/Users/don_sauer/Desktop/Tiger.jpg'
height = pdb.gimp_image_height(newImage) # 318
width = pdb.gimp_image_width(newImage) # 248
xresolution, yresolution = pdb.gimp_image_get_resolution(newImage) # (72.0, 72.0)
unit = pdb.gimp_image_get_unit(newImage) # 1
filename = pdb.gimp_image_get_filename(newImage) # '/Users/don_sauer/Desktop/Tiger.jpg'
name = pdb.gimp_image_get_name(newImage) # 'Tiger.jpg'

layer1 = pdb.gimp_image_get_layer_by_name(newImage, 'Background') # <gimp.Layer 'Background'>
layer2 = pdb.gimp_layer_new(newImage,width,height,1,'layer2',100,0) # 1 = RGBA
pdb.gimp_image_add_layer(newImage, layer2, 1) # adds trans layer

pdb.gimp_image_select_rectangle(newImage, 0, 50,50, 100, 100) #
non_empty = pdb.gimp_edit_copy(layer1) # 1

floatingSel = pdb.gimp_edit_paste(layer2, 0) # <gimp.Layer 'Pasted Layer'>
pdb.gimp_drawable_offset(floatingSel, 0, 1, 30, 30)
pdb.gimp_floating_sel_anchor(floatingSel)
#floatingSel # <gimp.Layer '(null)'>

pdb.gimp_image_lower_layer(newImage, layer1)

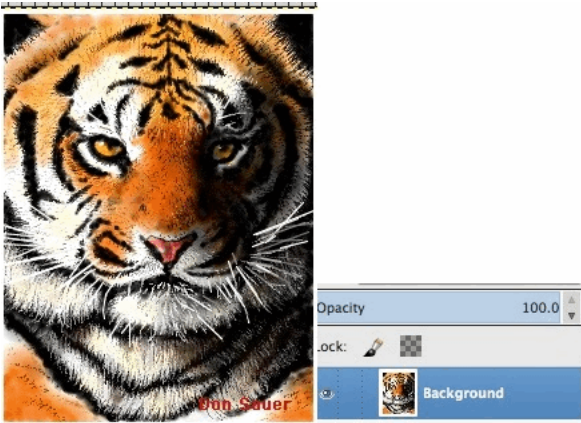
```

## =====SELECTIONS=====

```

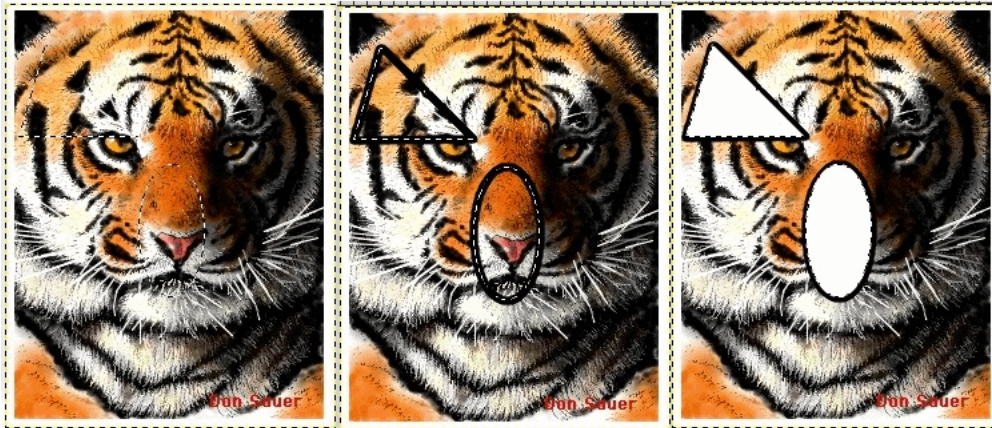
pdb.gimp_display_delete(display) # Optional delete previous image

```



```
filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
pdb.gimp_context_set_default_colors() # foreground and background colors to black and white
pdb.gimp_context_set_defaults()

drawable = pdb.gimp_image_get_active_drawable(newImage) # <gimp.Layer 'Background'>
activeLayer = pdb.gimp_image_get_active_layer(newImage) # <gimp.Layer 'Background'>
this_filename = pdb.gimp_image_get_filename(newImage) # '/Users/don_sauer/Desktop/Tiger.jpg'
height = pdb.gimp_image_height(newImage) # 318
width = pdb.gimp_image_width(newImage) # 248
layer1 = pdb.gimp_image_get_layer_by_name(newImage, 'Background') # <gimp.Layer 'Background'>
```



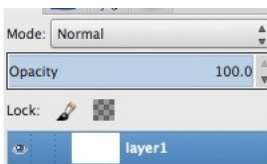
```
pdb.gimp_image_select_ellipse(newImage, 0, 100, 120, 50, 100)
pts2 = [30,30, 10,100, 100,100, 40,40]
pdb.gimp_image_select_polygon(newImage, 0,8, pts2)

NewBrush = pdb.gimp_brush_new("nextbrush")
radius_out = pdb.gimp_brush_set_radius(NewBrush,3)
hardness_out = pdb.gimp_brush_set_hardness(NewBrush, 1.0)
pdb.gimp_context_set_brush(NewBrush)
pdb.gimp_edit_stroke(layer1)
pdb.gimp_brush_delete(NewBrush)

pdb.gimp_edit_clear(layer1) # Clear selected area of drawable.
```

## =====BRUSH\_SETTINGS=====

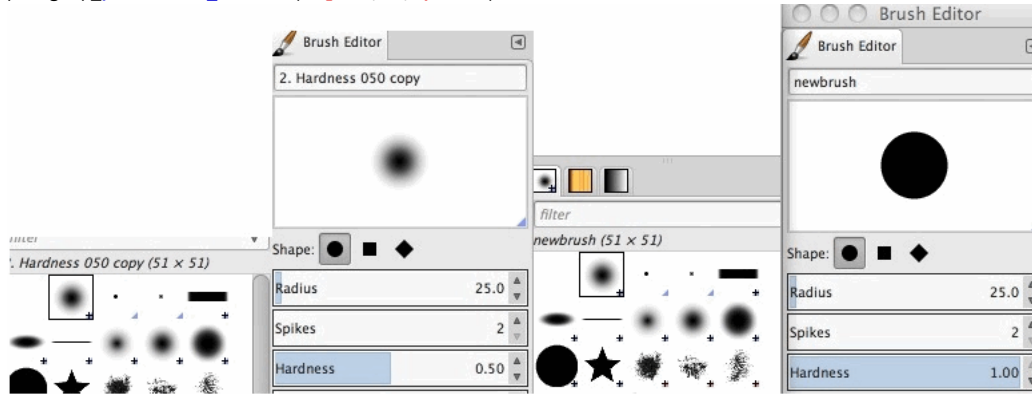
```
pdb.gimp_display_delete(display) # Optional delete previous image
```



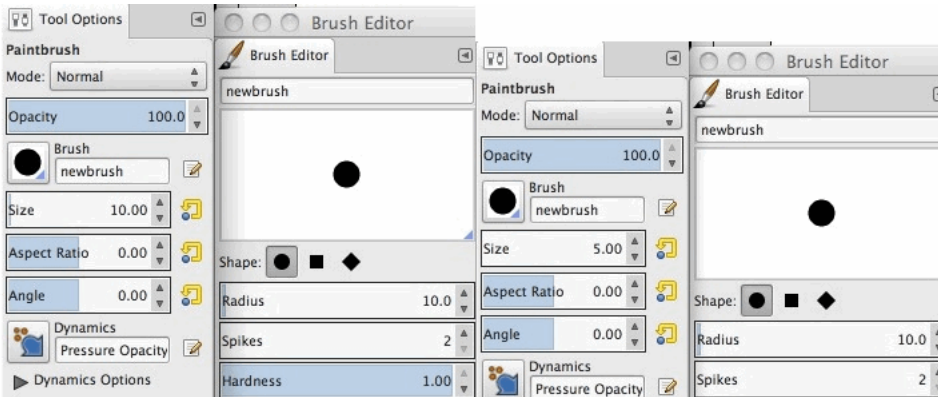
```
newImage = pdb.gimp_image_new(500, 300, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage, 500, 300, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_invert(layer1)
pdb.gimp_context_set_defaults()
```



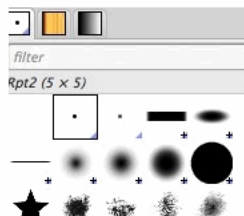
```
brushname = pdb.gimp_context_get_brush() # '2. Hardness 050'
pdb.gimp_context_set_brush('2. Hardness 050')
points = [50,50, 300,50]
pdb.gimp_paintbrush_default(layer1, 4, points)
```



```
newbrush = pdb.gimp_brush_duplicate('2. Hardness 050') # '2. Hardness 050' is not editable
newbrush = pdb.gimp_brush_rename(newbrush, "newbrush")
hardness_out = pdb.gimp_brush_set_hardness(newbrush, 1.0)
pdb.gimp_context_set_brush(newbrush)
points = [50,120, 300,120]
pdb.gimp_paintbrush_default(layer1, 4, points)
```

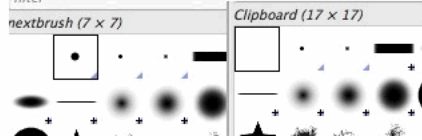


```
radius_out = pdb.gimp_brush_set_radius(newbrush, 10)
points = [50,180, 300,180]
pdb.gimp_paintbrush_default(layer1,4, points)
pdb.gimp_context_set_brush_size(5)
points = [50,230, 300,230]
pdb.gimp_paintbrush_default(layer1, 4, points)
pdb.gimp_brush_delete(newbrush)
```



```
nextbrush = pdb.gimp_brush_new("nextbrush")
radius_out = pdb.gimp_brush_set_radius(nextbrush,3)
hardness_out = pdb.gimp_brush_set_hardness(nextbrush, 1.0)
pdb.gimp_context_set_brush(nextbrush)
```

```
points = [50,280, 300,280]
pdb.gimp_paintbrush_default(layer1, 4, points)
```



```
pdb.gimp_brush_delete(nextbrush)
```

## =====PAINT\_METHODS=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
pdb.gimp_context_set_default_colors()
pdb.gimp_context_set_defaults()

# fore and back black and white

drawable = pdb.gimp_image_get_active_drawable(newImage)
activeLayer = pdb.gimp_image_get_active_layer(newImage)
this_filename = pdb.gimp_image_get_filename(newImage)
height = pdb.gimp_image_height(newImage)
width = pdb.gimp_image_width(newImage)
layer1 = pdb.gimp_image_get_layer_by_name(newImage, 'Background')
pdb.gimp_context_set_default_colors()
pdb.gimp_context_set_defaults()

# <gimp.Layer 'Background'>
# <gimp.Layer 'Background'>
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 318
# 248
# <gimp.Layer 'Background'>
# fore and back black and white

NewBrush = pdb.gimp_brush_new("nextbrush")
radius_out = pdb.gimp_brush_set_radius(NewBrush,10)
hardness_out = pdb.gimp_brush_set_hardness(NewBrush, 1.0)
pdb.gimp_brushes_set_spacing(1)
pdb.gimp_context_set_brush(NewBrush)

pdb.gimp_context_set_paint_method('gimp-smudge')
pdb.gimp_context_get_paint_method() #'gimp-smudge'
points = [80,30, 140,30]
pdb.gimp_smudge_default(activeLayer,len(points), points)

pdb.gimp_context_set_paint_method('gimp-eraser')
points = [80,60, 140,60]
pdb.gimp_eraser_default(activeLayer,len(points), points)

pdb.gimp_context_set_paint_method('gimp-clone')
points = [80,100, 140,100]
pdb.gimp_clone(activeLayer, activeLayer, 0, 80, 80, len(points), points)

pdb.gimp_context_set_paint_method('gimp-heal')
points = [100,125, 150,125]
pdb.gimp_heal(activeLayer, activeLayer, 55, 160, len(points), points)

pdb.gimp_context_set_paint_method('gimp-dodge-burn')
```



```

points = [80,150, 130,150]
exposure= 100
dodgeburn_type= 0 # DODGE (0), BURN (1)
dodgeburn_mode= 1 # SHADOWS (0), MIDTONES (1), HIGHLIGHTS
pdb.gimp_dodgeburn(activeLayer, exposure, dodgeburn_type, dodgeburn_mode, len(points), points)

pdb.gimp_context_set_paint_method('gimp-dodge-burn')
points = [80,180, 130,180]
dodgeburn_type= 1 # DODGE (0), BURN (1)
dodgeburn_mode= 2 # SHADOWS (0), MIDTONES (1), HIGHLIGHTS
pdb.gimp_dodgeburn(activeLayer, exposure, dodgeburn_type, dodgeburn_mode, len(points), points)

pdb.gimp_context_set_paint_method('gimp-airbrush')
pressure = 50 # 100 mx
points = [80,210, 130,210]
pdb.gimp_airbrush(activeLayer, pressure, len(points), points)

pdb.gimp_context_set_paint_method('gimp-paintbrush')
points = [80,250, 130,250]
pdb.gimp_paintbrush_default(activeLayer, len(points), points)

pdb.gimp_context_set_paint_method('gimp-convolve')
points = [80,280, 130,280]
type =0 # BLUR-CONVOLVE (0), SHARPEN-CON
pdb.gimp_convolve(drawable, 100,type, len(points), points)

pdb.gimp_brush_delete(NewBrush)

num_paint_methods, paint_methods = pdb.gimp_context_list_paint_methods()
#(11, ('gimp-pencil', 'gimp-paintbrush', 'gimp-eraser', 'gimp-airbrush',
# 'gimp-ink', 'gimp-clone', 'gimp-heal', 'gimp-perspective-clone',
# 'gimp-convolve', 'gimp-smudge', 'gimp-dodge-burn'))

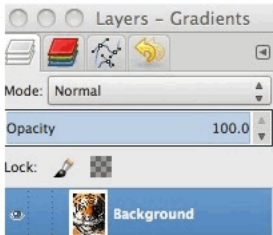
```

## =====TRANSFORMS=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)

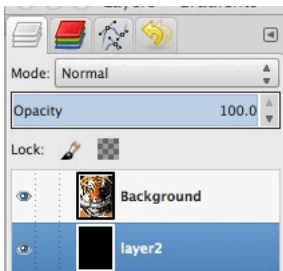
```

```

drawable = pdb.gimp_image_get_active_drawable(newImage)
activeLayer = pdb.gimp_image_get_active_layer(newImage)
this_filename = pdb.gimp_image_get_filename(newImage)
height = pdb.gimp_image_height(newImage)
width = pdb.gimp_image_width(newImage)
xresolution, yresolution = pdb.gimp_image_get_resolution(newImage)
unit = pdb.gimp_image_get_unit(newImage)
filename = pdb.gimp_image_get_filename(newImage)
name = pdb.gimp_image_get_name(newImage)

# <gimp.Layer 'Background'>
# <gimp.Layer 'Background'>
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 318
# 248
# (72.0, 72.0)
# 1
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 'Tiger.jpg'

```



```

layer1 = pdb.gimp_image_get_layer_by_name(newImage, 'Background') # <gimp.Layer 'Background'>
layer2 = pdb.gimp_layer_new(newImage,width,height,0,'layer2',100,0) # 0 = RGBA <gimp.Layer 'layer4'>

```



```

pdb.gimp_image_add_layer(newImage, layer2, 3) # adds black layer

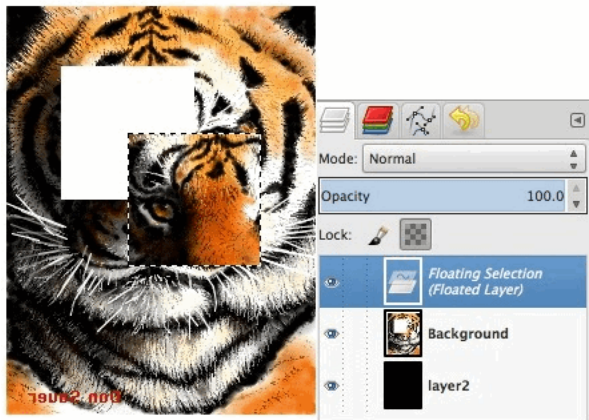
```



```

pdb.gimp_image_flip(newImage, 0) # HORIZONTAL (0), VERTICAL (1)

```



```

pdb.gimp_image_select_rectangle(newImage, 0, 50,50, 100, 100) #
non_empty = pdb.gimp_edit_copy(layer1)
floatingSel = pdb.gimp_selection_float(layer1, 50, 50)

```



```

rotate_type = 0 #ROTATE-90 (0), ROTATE-180 (1), ROTATE-270 (2) }
auto_center = 1
center_x = 0
center_y = 0
item = pdb.gimp_item_transform_rotate_simple(floatingSel,rotate_type,auto_center,center_x,center_y)

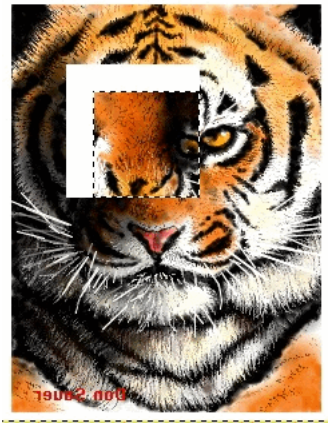
```



```

auto_center = 0
center_x = 150
center_y = 100
item = pdb.gimp_item_transform_rotate_simple(floatingSel, rotate_type, auto_center, center_x, center_y)

```



```

wrap_around = 0
fill_type = 1 # OFFSET-BACKGROUND (0), OFFSET-TRANSPARENT (1) }
offset_x = 20
offset_y = 20
pdb.gimp_drawable_offset(floatingSel, wrap_around, fill_type, offset_x, offset_y)

```



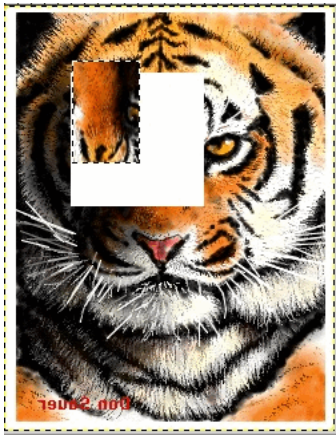
```

item = pdb.gimp_item_transform_scale(floatingSel, 60, 60, 140, 140)

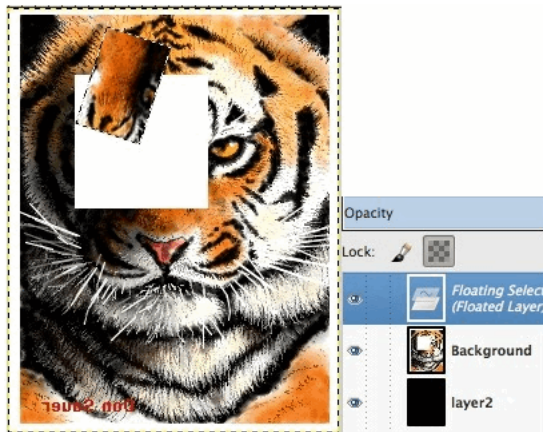
offset_x, offset_y = pdb.gimp_drawable_offsets(floatingSel)
transform_direction = pdb.gimp_context_get_transform_direction()
transform_resize = pdb.gimp_context_get_transform_resize()

# (60, 60)
# 0 FORWARD0, BACKWARD1
# 0 ADJUST0, CLIP1, CROP2, CASPECT3

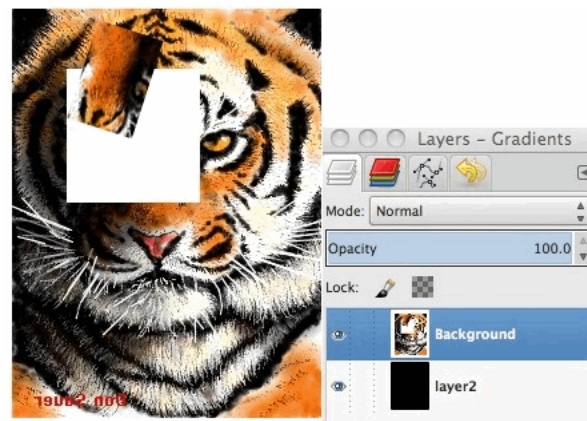
```



```
InCenterx = 100
InCentery = 100
OutCenterx = 70
OutCentery = 70
scale_x = .8
scale_y = 1.2
angle = 0
item = pdb.gimp_item_transform_2d(floatingSel, InCenterx, InCentery, scale_x, scale_y, angle, OutCenterx, OutCentery)
```



```
angle = 3.14/10
auto_center = 0
center_x = 150
center_y = 100
item = pdb.gimp_item_transform_rotate(floatingSel, angle, auto_center, center_x, center_y)
```

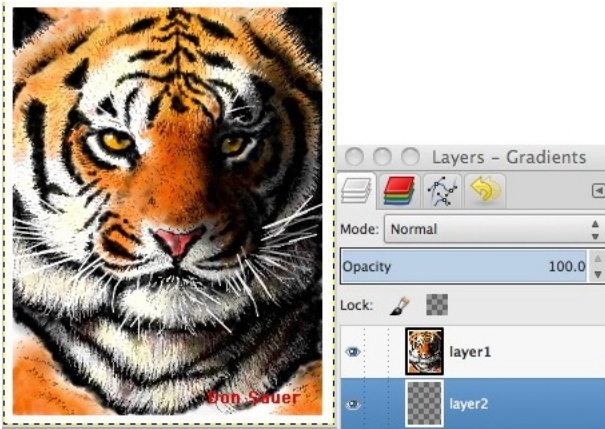


```
pdb.gimp_floating_sel_attach(floatingSel, layer1)
```

## =====MASKING=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



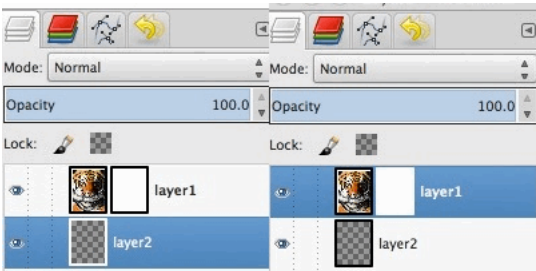


```
filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)

drawable = pdb.gimp_image_get_active_drawable(newImage)
activeLayer = pdb.gimp_image_get_active_layer(newImage)
this_filename = pdb.gimp_image_get_filename(newImage)
height = pdb.gimp_image_height(newImage)
width = pdb.gimp_image_width(newImage)
xresolution, yresolution = pdb.gimp_image_get_resolution(newImage)
unit = pdb.gimp_image_get_unit(newImage)
filename = pdb.gimp_image_get_filename(newImage)
name = pdb.gimp_image_get_name(newImage)

# <gimp.Layer 'Background'>
# <gimp.Layer 'Background'>
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 318
# 248
# (72.0, 72.0)
# 1
# '/Users/don_sauer/Desktop/Tiger.jpg'
# 'Tiger.jpg'

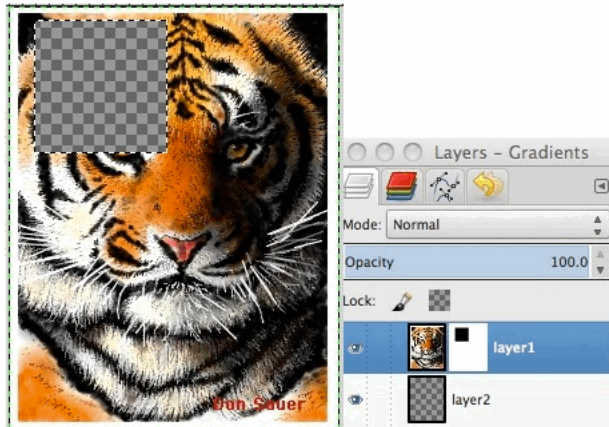
layer1 = pdb.gimp_image_get_layer_by_name(newImage, 'Background') # <gimp.Layer 'Background'>
pdb.gimp_layer_set_name(layer1, 'layer1')
layer2 = pdb.gimp_layer_new(newImage, width, height, 1, 'layer2', 100, 0) # 1 = RGBA
pdb.gimp_image_add_layer(newImage, layer2, 1) # adds trans layer
```



```
mask_type = 0 # WHITE0, BLACK1, ALPHA2, ALPHATRANSFER3, SELECTION4, COPY5, CHANNEL6
mask1 = pdb.gimp_layer_create_mask(layer1, mask_type)
pdb.gimp_layer_add_mask(layer1, mask1)

pdb.gimp_image_set_active_layer(newImage, layer1)
edit_mask = pdb.gimp_layer_get_edit_mask(layer1)
mask = pdb.gimp_layer_get_mask(layer1)
is_layer_mask = pdb.gimp_item_is_layer_mask(mask1)
is_layer_mask = pdb.gimp_drawable_is_layer_mask(mask1)

# 1
# <gimp.Channel 'layer1 mask'>
# 1
```

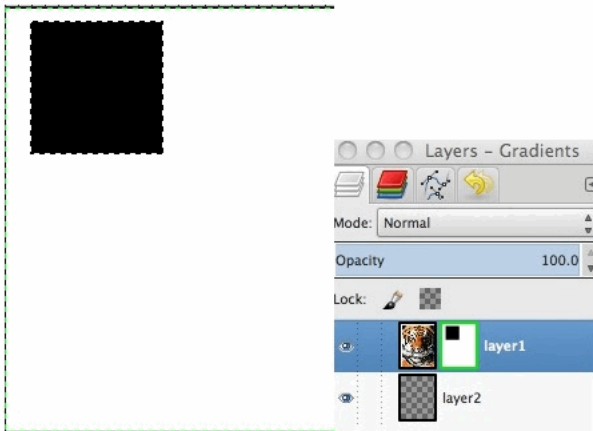


```
pdb.gimp_rect_select(newImage, 19, 10, 100, 100, 0, 0, 1)
fill_mode = 0 # FG0, BG1, PATTERN2
paint_mode = 0
opacity = 100
threshold = 100
```

```

sample_merged = 0
x = 0
y = 0
pdb.gimp_edit_bucket_fill(mask1, fill_mode, paint_mode, opacity, threshold, sample_merged, x, y)
non_empty, x1, y1, x2, y2 = pdb.gimp_drawable_mask_bounds(mask1) #(1, 19, 10, 119, 110)
apply_mask = pdb.gimp_layer_get_apply_mask(layer1) #1
edit_mask = pdb.gimp_layer_get_edit_mask(layer1) #1
mask = pdb.gimp_layer_get_mask(layer1) # <gimp.Channel 'layer1 mask'>
show_mask = pdb.gimp_layer_get_show_mask(layer1) #0

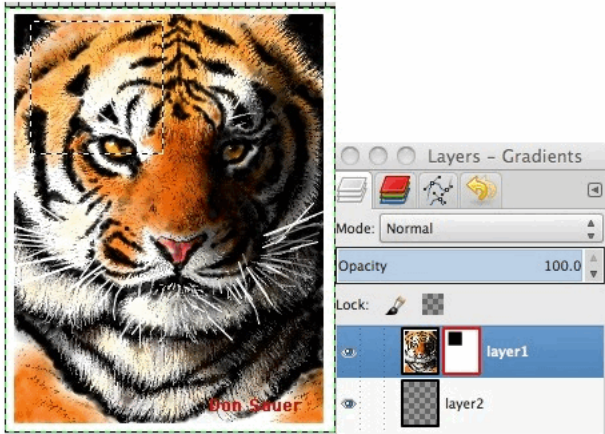
```



```

pdb.gimp_layer_set_show_mask(layer1, 1)
apply_mask = pdb.gimp_layer_get_apply_mask(layer1) #1

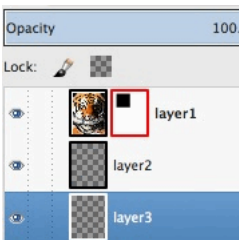
```



```

pdb.gimp_layer_set_show_mask(layer1, 0)
pdb.gimp_layer_set_apply_mask(layer1, 0)

```

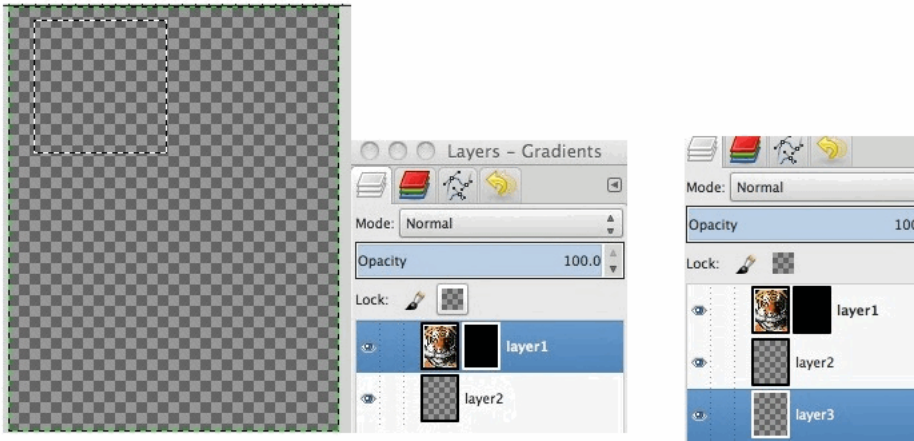


```

layer3 = pdb.gimp_layer_copy(layer2, 1)
pdb.gimp_image_add_layer(newimage, layer3, 2) # adds trans layer
pdb.gimp_layer_set_name(layer3, "layer3")

```

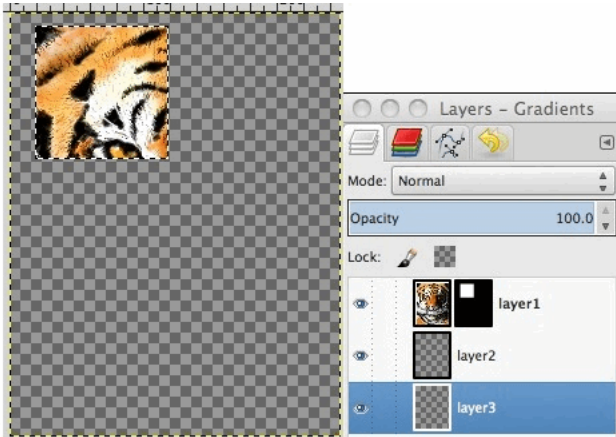




```

pdb.gimp_drawable_fill(mask1, 0)
pdb.gimp_layer_set_apply_mask(layer1, 1)

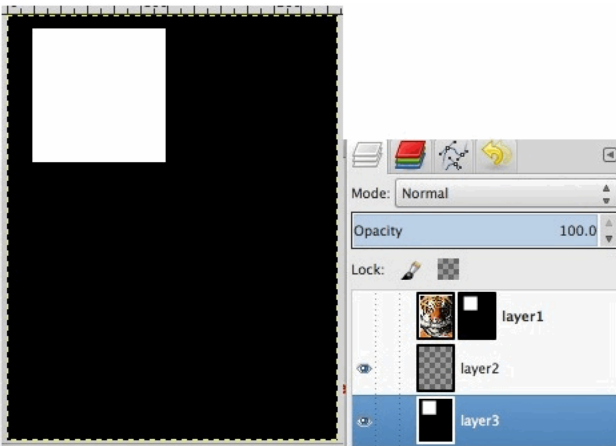
```



```

pdb.gimp_context_swap_colors()
pdb.gimp_edit_bucket_fill(mask1, 0, 0, 100, 5, 0, 0, 0)

```



```

pdb.gimp_selection_all(newImage)
non_empty = pdb.gimp_edit_copy(mask1)
floating_sel = pdb.gimp_edit_paste(layer3, 0)
pdb.gimp_floating_sel_anchor(floating_sel)
pdb.gimp_item_set_visible(layer1, 0)

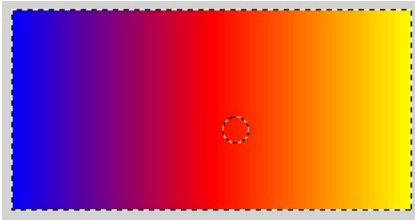
```

## =====GRADIENTS=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



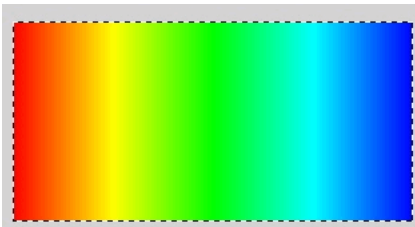
```
w = 300
h = 150
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_edit_fill(layer1, 2)
pdb.gimp_context_set_default_colors()

pdb.gimp_selection_all(newImage)

thermalgrad = pdb.gimp_gradient_new("thermalgrad") # 'thermalgrad'
pdb.gimp_gradient_segment_range_split_uniform(thermalgrad, 0, 1, 2) # start, end, numb parts
pdb.gimp_gradient_segment_set_left_color(thermalgrad, 0, (0,0,255), 100)
pdb.gimp_gradient_segment_set_right_color(thermalgrad, 0, (255,0,0), 100)
pdb.gimp_gradient_segment_set_left_color(thermalgrad, 1, (255,0,0), 100)
pdb.gimp_gradient_segment_set_right_color(thermalgrad, 1, (255,255,0), 100)

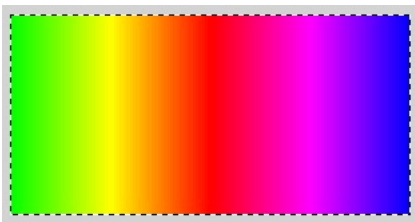
pdb.gimp_context_set_gradient(thermalgrad)

blend = 3 # FG-BG-RGB0, FG-BG-HSV1, FG-TRANS2, CUSTOM3
paint = 0 # NOR0, DIS1, BHIN2, MUL3, SCR4, OVER5, DIFF6, ADD7, SUBTRACT-MOD
grad = 0 # LIN0, BILIN1, RAD), SQR3, CONSYM), CONASYM5, BURSTANG6, BURSTSPHER7, BURSTDIMP8, CLOCK9, CCLOCK10
opac = 100
off = 0
repeat = 0
rev = 0
ssamp = 0
dep = 0
thres = 0
dith = 0
x1 = 0
y1 = h/2
x2 = w
y2 = h/2
pdb.gimp_edit_blend(layer1, blend, paint, grad, opac, off, repeat, rev, ssamp, dep, thres, dith, x1, y1, x2, y2)
pdb.gimp_gradient_delete(thermalgrad)
```



```
grad2 = pdb.gimp_gradient_new("grad2") # 'grad2'
pdb.gimp_gradient_segment_range_split_uniform(grad2, 0, 3, 4)
pdb.gimp_gradient_segment_set_left_color(grad2, 0, (255,0,0), 100)
pdb.gimp_gradient_segment_set_right_color(grad2, 0, (255,255,0), 100)
pdb.gimp_gradient_segment_set_left_color(grad2, 1, (255,255,0), 100)
pdb.gimp_gradient_segment_set_right_color(grad2, 1, (0,255,0), 100)
pdb.gimp_gradient_segment_set_left_color(grad2, 2, (0,255,0), 100)
pdb.gimp_gradient_segment_set_right_color(grad2, 2, (0,255,255), 100)
pdb.gimp_gradient_segment_set_left_color(grad2, 3, (0,255,255), 100)
pdb.gimp_gradient_segment_set_right_color(grad2, 3, (0,0,255), 100)

pdb.gimp_context_set_gradient(grad2)
pdb.gimp_edit_blend(layer1, blend, paint, grad, opac, off, repeat, rev, ssamp, dep, thres, dith, x1, y1, x2, y2)
pdb.gimp_gradient_delete(grad2)
```



```
grad3 = pdb.gimp_gradient_new("grad3") # 'grad3'
pdb.gimp_gradient_segment_range_split_uniform(grad3, 0, 3, 4)
pdb.gimp_gradient_segment_set_left_color(grad3, 0, (0,255,0), 100)
pdb.gimp_gradient_segment_set_right_color(grad3, 0, (255,255,0), 100)
pdb.gimp_gradient_segment_set_left_color(grad3, 1, (255,255,0), 100)
pdb.gimp_gradient_segment_set_right_color(grad3, 1, (255,0,0), 100)
pdb.gimp_gradient_segment_set_left_color(grad3, 2, (255,0,0), 100)
```

```

pdb.gimp_gradient_segment_set_right_color(grad3, 2, (255,0,255), 100)
pdb.gimp_gradient_segment_set_left_color(grad3, 3, (255,0,255), 100)
pdb.gimp_gradient_segment_set_right_color(grad3, 3, (0,0,255), 100)

pdb.gimp_context_set_gradient(grad3)
pdb.gimp_edit_blend(layer1,blend,paint,grad,opac,off,repeat,rev,ssamp,dep,thres,dith,x1,y1,x2,y2)

pdb.gimp_gradient_delete(grad3)

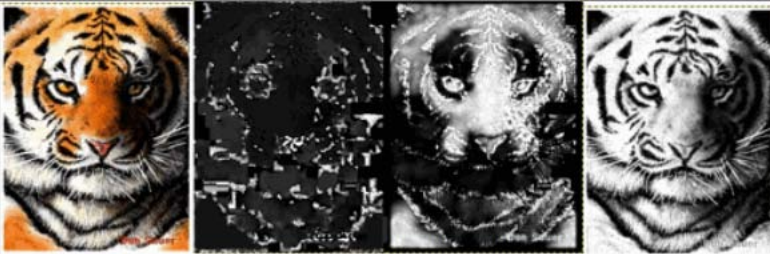
```

## =====DeCompose/Compose/HSV=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")
pdb.gimp_context_set_default_colors()
width = pdb.gimp_image_width(newImage)
height = pdb.gimp_image_height(newImage)
pdb.gimp_layer_add_alpha(layer1)

imgH, imgS, imgV, imgA = pdb.plug_in_decompose(newImage, layer1, "HSV", 0)
#(<gimp.Image 'Tiger-hue.jpg'>, <gimp.Image 'Tiger-saturation.jpg'>, <gimp.Image 'Tiger-value.jpg'>, None)
display = pdb.gimp_display_new(imgH)
display = pdb.gimp_display_new(imgS)
display = pdb.gimp_display_new(imgV)

newHSVImage = pdb.plug_in_compose(imgH, layer1, imgS, imgV, None, "HSV")
display2 = pdb.gimp_display_new(newHSVImage)

```

## =====DeCompose/Compose\_RGB=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")
pdb.gimp_context_set_default_colors()
width = pdb.gimp_image_width(newImage)
height = pdb.gimp_image_height(newImage)
pdb.gimp_layer_add_alpha(layer1)

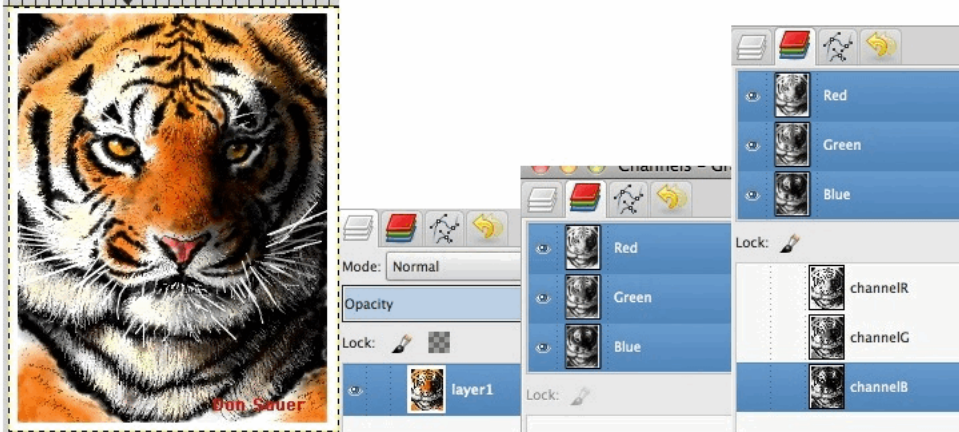
imgR, imgG, imgB, imgA = pdb.plug_in_decompose(newImage, layer1, "RGBA", 0)
#(<gimp.Image 'Tiger-red.jpg'>, <gimp.Image 'Tiger-green.jpg'>, <gimp.Image 'Tiger-blue.jpg'>, <gimp.Image 'Tiger-alpha.jpg'>)
display = pdb.gimp_display_new(imgR)
display = pdb.gimp_display_new(imgG)
display = pdb.gimp_display_new(imgB)
display = pdb.gimp_display_new(imgA)

newRGBAImage = pdb.plug_in_compose(imgR, layer1, imgG, imgB, imgA, "RGBA")
display2 = pdb.gimp_display_new(newRGBAImage)

```

## =====Channels\_RGB=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")
pdb.gimp_context_set_default_colors()
width = pdb.gimp_image_width(newImage)
height = pdb.gimp_image_height(newImage)

component = 0 # { RED0, GREEN1, BLUE2, GRAY3, INDEXED4, ALPHA5 }
channelR = pdb.gimp_channel_new_from_component(newImage, 0, "channelR")
pdb.gimp_image_add_channel(newImage, channelR, 0)

channelG = pdb.gimp_channel_new_from_component(newImage, 1, "channelG")
pdb.gimp_image_add_channel(newImage, channelG, 1)

channelB = pdb.gimp_channel_new_from_component(newImage, 2, "channelB")
pdb.gimp_image_add_channel(newImage, channelB, 2)

num_channels, channel_ids = pdb.gimp_image_get_channels(newImage) # (3, (125, 126, 127))
color = pdb.gimp_channel_get_color(channelR) # gimpcolor.RGB(0.0, 0.0, 0.0, 1.0)
color = pdb.gimp_channel_get_color(channelG) # gimpcolor.RGB(0.0, 0.0, 0.0, 1.0)
color = pdb.gimp_channel_get_color(channelB) # gimpcolor.RGB(0.0, 0.0, 0.0, 1.0)

pdb.gimp_channel_set_color(channelR, (255,0,0,0) )
pdb.gimp_channel_set_color(channelG, (0,255,0,0) )
pdb.gimp_channel_set_color(channelB, (0,0,255,0) )

color = pdb.gimp_channel_get_color(channelR) # gimpcolor.RGB(1.0, 0.0, 0.0, 1.0)
color = pdb.gimp_channel_get_color(channelG) # gimpcolor.RGB(0.0, 1.0, 0.0, 1.0)
color = pdb.gimp_channel_get_color(channelB) # gimpcolor.RGB(0.0, 0.0, 1.0, 1.0)
```



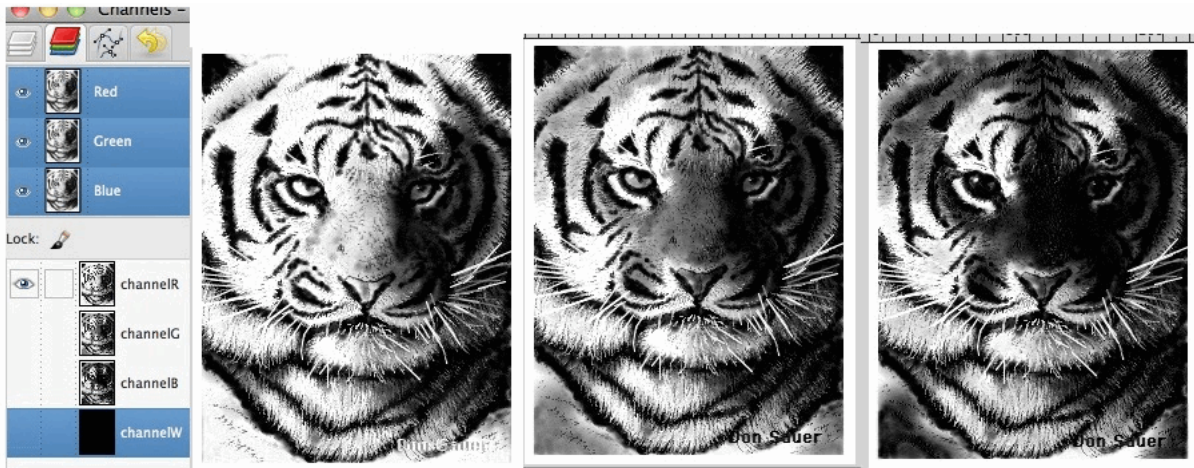
```
channelW = pdb.gimp_channel_new(newImage, width, height, "channelW", 100, (255,255,255))
pdb.gimp_image_add_channel(newImage, channelW, 3)
color = pdb.gimp_channel_get_color(channelW) # gimpcolor.RGB(0.0, 0.0, 0.0, 1.0)

pdb.gimp_item_set_visible(channelR, 1)
pdb.gimp_item_set_visible(channelR, 0)

pdb.gimp_item_set_visible(channelG, 1)
pdb.gimp_item_set_visible(channelG, 0)

pdb.gimp_item_set_visible(channelB, 1)
pdb.gimp_item_set_visible(channelB, 0)
```





```

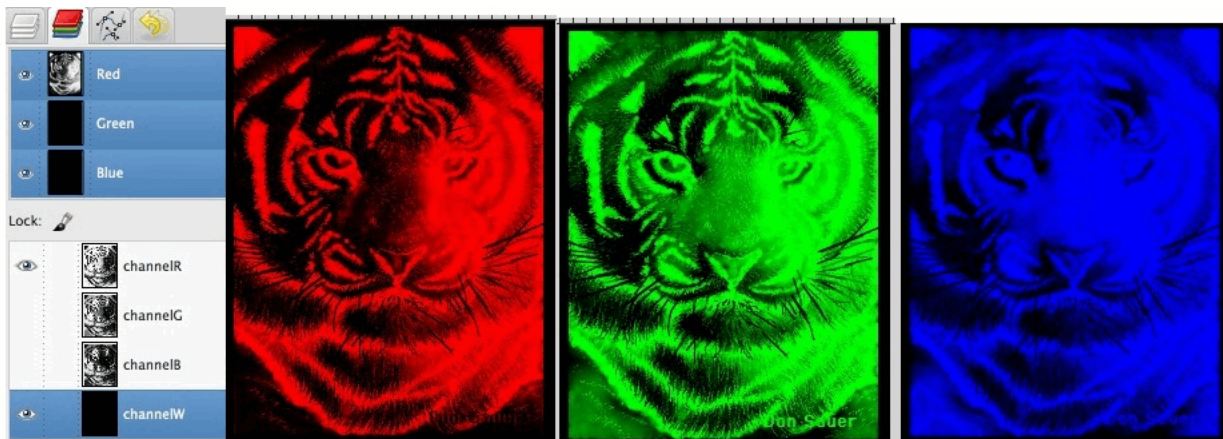
pdb.gimp_item_set_visible(layer1, 0)
pdb.gimp_item_set_visible(channelW, 0)

pdb.gimp_item_set_visible(channelR, 1)
pdb.gimp_item_set_visible(channelR, 0)

pdb.gimp_item_set_visible(channelG, 1)
pdb.gimp_item_set_visible(channelG, 0)

pdb.gimp_item_set_visible(channelB, 1)
pdb.gimp_item_set_visible(channelB, 0)

```



```

pdb.gimp_item_set_visible(channelW, 1)

```

## ===== PALETTES =====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)

```

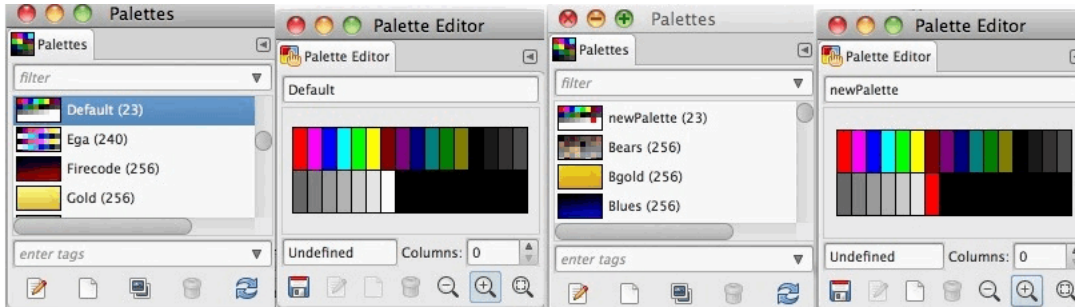


```

display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")

palname = pdb.gimp_context_get_palette() #'Default'
pdb.plugin_palettemap(newImage, layer1)
palette_gradient = pdb.python_fu_palette_to_gradient(palname)
actual_name = pdb.gimp_gradient_rename(palette_gradient, "palette_gradient")
pdb.gimp_gradient_delete("palette_gradient")

```



```

palname = pdb.gimp_context_get_palette() #'Default'
color = pdb.gimp_palette_entry_get_color(palname, 0) #gimpcolor.RGB(1.0, 0.0, 0.0, 1.0)
color = pdb.gimp_palette_entry_get_color(palname, 1) #gimpcolor.RGB(1.0, 0.0, 1.0, 1.0)
num_colors, colors = pdb.gimp_palette_get_colors(palname) #(23,(gimpcolor.RGB(1.0, 0.0, 0.0, 1.0).....
Iseditable = pdb.gimp_palette_is_editable(palname) #0

newPalette = pdb.gimp_palette_duplicate(palname)
newPalette = pdb.gimp_palette_rename(newPalette, "newPalette")
pdb.gimp_palette_entry_set_color(newPalette, 22, (255,0,0))

pdb.gimp_palette_delete(newPalette)
pdb.gimp_context_set_palette(palname)

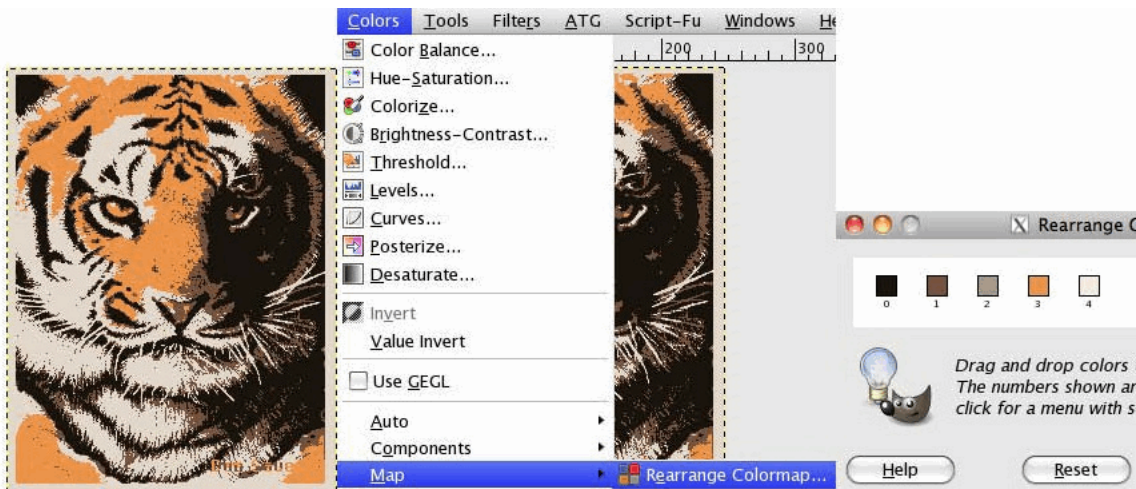
```

## =====COLOR\_MAPS=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

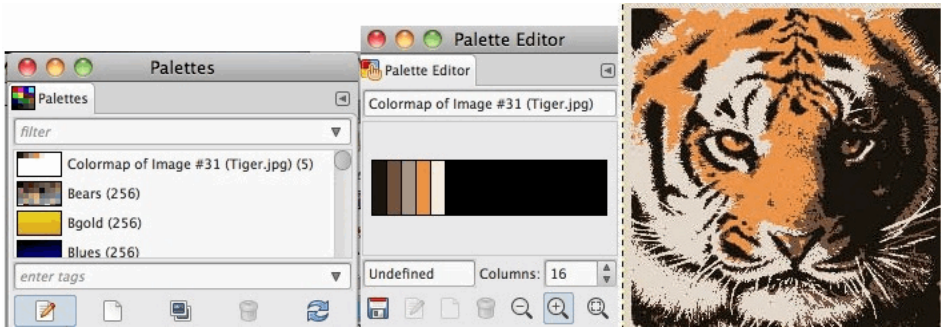
```



```

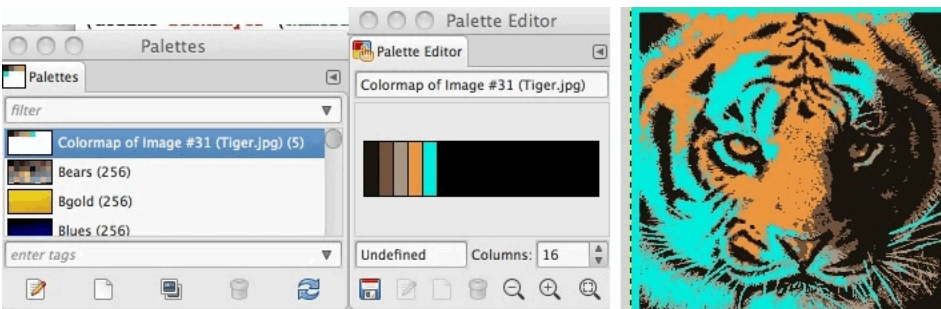
filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")

```



```
dith = 0 # NO-DITHER (0), FS-DITHER (1), FLOWBLEED-DITHER (2), FIXED-DITHER (3) }
palet = 0 # MAKE-PALETTE (0), WEB-PALETTE (2), MONO-PALETTE (3), CUSTOM-PALETTE (4) }
numcols = 5
alpha = 0
unused = 0
usepalette = "null"
pdb.gimp_image_convert_indexed(newImage, dith, palet, numcols, alpha, unused, usepalette)

num_bytes, colormap = pdb.gimp_image_get_colormap(newImage)
#(15,(27,21,15,115,83,61,168,150,134,237,150,64,245,237,224))
colormap[3] #115
```



```
Ncolormap = [27,21,15,115,83,61,168,150,134,237,150,64,0,237,224]
pdb.gimp_image_set_colormap(newImage, 15, Ncolormap)
```

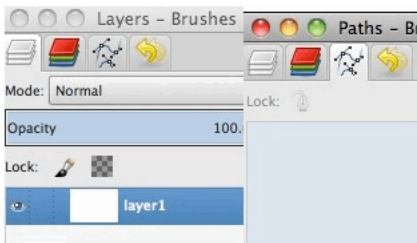


```
pdb.script_fu_cmap_to_palette(newImage, layer1)
pdb.gimp_context_set_palette("Tiger.jpg")

ascending = 1
pdb.python_fu_palette_sort("Tiger.jpg", "RGB", "Red", ascending)
```

## =====Paint\_and\_Vectors\_and\_Select=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```

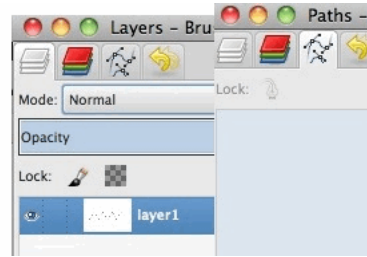
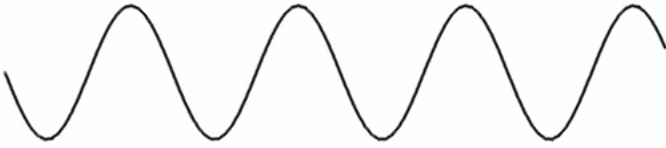


```
import math, string

w = 600
h = 400

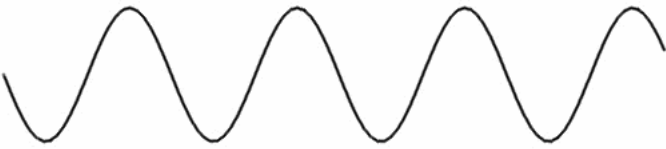
pdb.gimp_display_delete(display) # optional

newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()
```

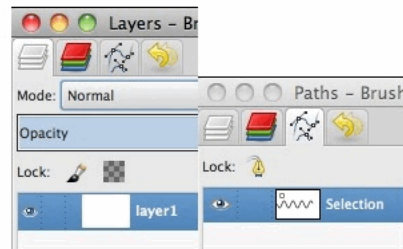


```
pt = []
for i in range(100):
    pt.append(10+i*5)
    pt.append(200 + 50*math.sin(float(i)/4))

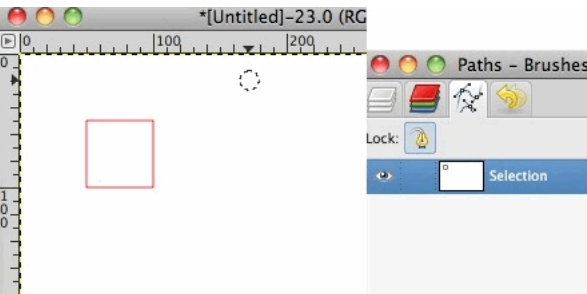
pdb.gimp_context_set_paint_method('gimp-paintbrush')
name, width, height, spacing = pdb.gimp_brushes_get_brush()
pdb.gimp_brushes_set_brush('Circle (02)')
pdb.gimp_paintbrush_default(layer1, len(pt), pt)
```



```
pdb.gimp_rect_select(newImage, 50, 50, 50, 50, 0, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3
pdb.gimp_edit_bucket_fill(layer1,0,0,100,0,0,0,0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL...)op,thres,merg, x, y
```



```
pdb.gimp_by_color_select(layer1, (0,0,0), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
vect2 = pdb.plug_in_sel2path(newImage, layer1)
pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(layer1)
pdb.gimp_selection_none(newImage)
activeVectors = pdb.gimp_image_get_active_vectors(newImage)
pdb.gimp_vectors_set_visible(activeVectors, 1)
num_strokes, stroke_ids = pdb.gimp_vectors_get_strokes(activeVectors) # (2, (1, 2))
```



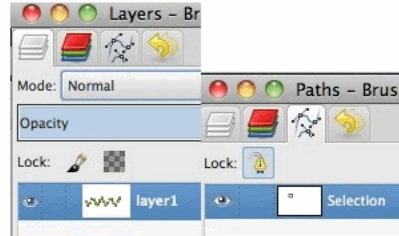
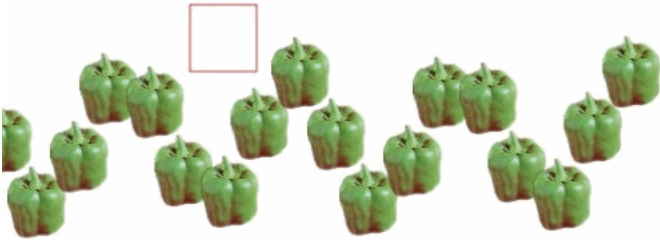
```
pdb.gimp_vectors_remove_stroke(activeVectors, stroke_ids[1])
```



```

pdb.gimp_vectors_to_selection(activeVectors, 0, 0, 0, 0, 0)
pdb.gimp_vectors_stroke_translate(activeVectors, stroke_ids[0], 100, 50)

```



```

pdb.gimp_selection_all(newImage)
pdb.gimp_brushes_set_brush('z Pepper')
pdb.gimp_context_set_paint_method('gimp-paintbrush')
pdb.gimp_paintbrush_default(layer1, len(pt), pt)

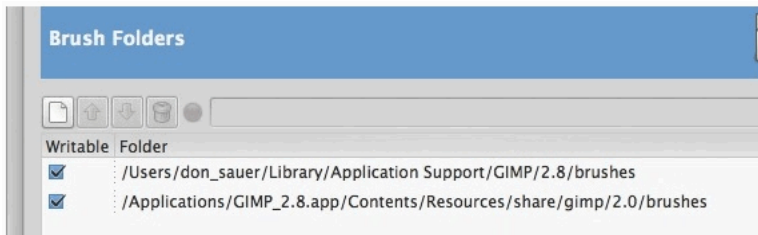
```

## =====CREATE\_PaintBrush=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

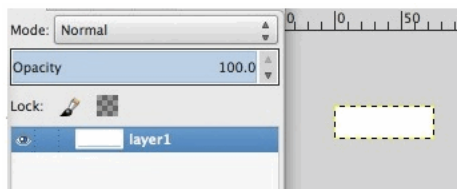
```



```

w = 75
h = 25
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

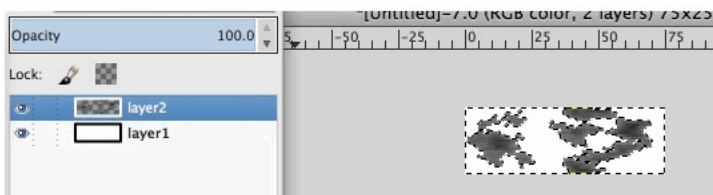
```



```

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2') # adds layer on top
pdb.gimp_image_add_layer(newImage, layer2, 0)
pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(layer2)
pdb.gimp_selection_none(newImage)

```



```

pdb.gimp_ellipse_select(newImage, 0, 0, 75, 25, 0, 0, 0, 0) # x,y,w,h,add?,alias,feath, rad
pdb.plug_in_solid_noise(newImage, layer2,1,0,233,15,4,4) # tile turb detail radx rady
pdb.gimp_selection_none(newImage)
pdb.gimp_by_color_select(layer2, (0,0,0), 120, 0, 0, 0, 3, 0) # 60 threshold radius 3

```



```

pdb.gimp_selection_invert(newImage)
pdb.gimp_edit_clear(layer2)
pdb.gimp_image_remove_layer(newImage, layer1)
pdb.gimp_selection_none(newImage)

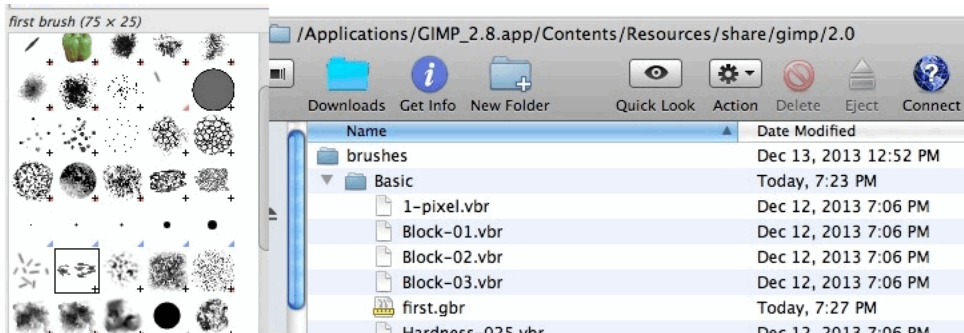
```



```

pdb.gimp_convert_grayscale(newImage)
fpath = "/Applications/GIMP_2.8.app/Contents/Resources/share/gimp/2.0/brushes/Basic/first.gbr"
pdb.file_gbr_save(newImage, layer2, fpath, "first", 3, "first brush")

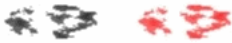
```



```

pdb.gimp_display_delete(display) # optional
pdb.gimp_brushes_refresh()

```



```

w = 300
h = 300
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

pdb.gimp_context_set_paint_method('gimp-paintbrush')
name, width, height, spacing = pdb.gimp_brushes_get_brush() # ('first brush', 75, 25, 3)
pdb.gimp_brushes_set_brush('first brush')
pt = [ 100,100,100,100]
pdb.gimp_paintbrush_default(layer1, len(pt), pt)

pdb.gimp_context_set_foreground((255,0,0))
pt = [ 200,100,200,100]
pdb.gimp_paintbrush_default(layer1, len(pt), pt)

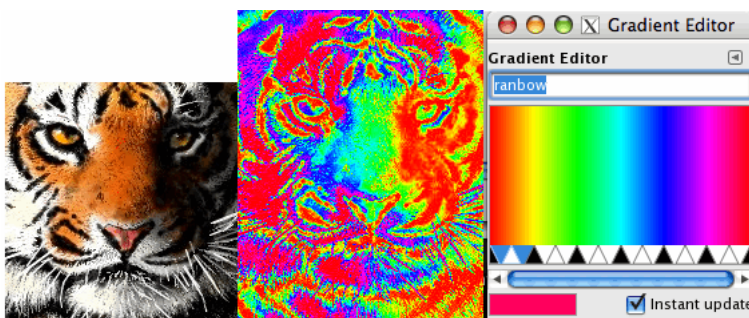
```

## ====Color\_Map\_With\_Gradient=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)

```

```

display = pdb.gimp_display_new(newImage)
Background = pdb.gimp_image_get_active_layer(newImage) # <gimp.Layer 'Background'>
height = pdb.gimp_image_height(newImage) # 318
width = pdb.gimp_image_width(newImage) # 248

layer2 = pdb.gimp_layer_new(newImage,width,height,1,'layer2',100,0) # 1 = RGBA
pdb.gimp_image_add_layer(newImage, layer2, 1) # adds trans layer

pdb.gimp_desaturate(Background)
pdb.gimp_gradients_set_gradient("Full saturation spectrum CCW")
pdb.plug_in_gradmap(newImage, Background)

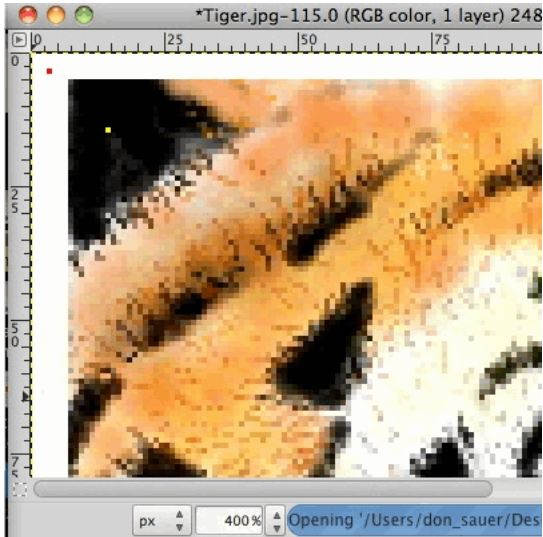
```

## =====Get\_Set\_Pixels=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
activeLayer = pdb.gimp_image_get_active_layer(newImage) # <gimp.Layer 'Background'>
height = pdb.gimp_image_height(newImage) # 318
width = pdb.gimp_image_width(newImage) # 248

num_channels, pixel = pdb.gimp_drawable_get_pixel(activeLayer, 73, 61) # (3, (254, 253, 221))
num_channels, pixel = pdb.gimp_drawable_get_pixel(activeLayer, 130, 130) # (3, (226, 92, 3))
merge = 0
average = 1
radius = 5
color = pdb.gimp_image_pick_color(newImage, activeLayer, 73, 40, merge, average, radius)
color # gimpcolor.RGB(0.9686274509803922, 0.6823529411764706, 0.34901960784313724, 1.0)
pdb.gimp_drawable_get_pixel(activeLayer, 3, 3)
pdb.gimp_drawable_set_pixel(activeLayer, 3, 3, 3, (255,0,0))
pdb.gimp_drawable_get_pixel(activeLayer, 3, 3)

pixel2 = activeLayer.get_pixel(3,10) # (253, 0, 0)

activeLayer.set_pixel(14,14, (255, 255, 0))
# layer.set_pixel(x,y, newColor)

activeLayer.name # 'Background'

pdb.gimp_drawable_update(activeLayer, 0, 0, width, height)
#activeLayer.update(0, 0, width, height)

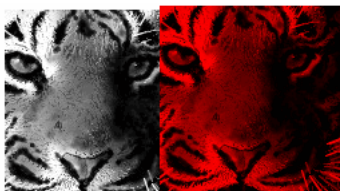
```

## =====Colorify=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"

```

```

rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
activeLayer = pdb.gimp_image_get_active_layer(newImage)           # <gimp.Layer 'Background'>
height = pdb.gimp_image_height(newImage)                          # 318
width = pdb.gimp_image_width(newImage)                             # 248

pdb.gimp_desaturate(activeLayer)
pdb.plug_in_colorify(newImage, activeLayer, (255, 0, 0))

```

## =====Alpha\_From\_Color=====

```

pdb.gimp_display_delete(display)           # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
activeLayer = pdb.gimp_image_get_active_layer(newImage)           # <gimp.Layer 'Background'>
height = pdb.gimp_image_height(newImage)                          # 318
width = pdb.gimp_image_width(newImage)                             # 248

pdb.plug_in_colortoalpha(newImage, activeLayer, (0,0,0))
pdb.plug_in_threshold_alpha(newImage, activeLayer, 100)

```

## =====REMOVE\_BACKGROUND\_COLOR=====

```

pdb.gimp_display_delete(display)           # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/flower.gif"
rawFilename = "flower.gif"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
Background = pdb.gimp_image_get_active_layer(newImage)           # <gimp.Layer 'Background'>
height = pdb.gimp_image_height(newImage)                          # 318
width = pdb.gimp_image_width(newImage)                             # 248
pdb.gimp_convert_rgb(newImage)

pdb.gimp_by_color_select(Background, (0, 59, 155), 70, 0, 1, 0, 0, 0)
pdb.gimp_bucket_fill(Background, 1, 0, 100, 100, 0, 0, 0)
pdb.gimp_selection_none(newImage)

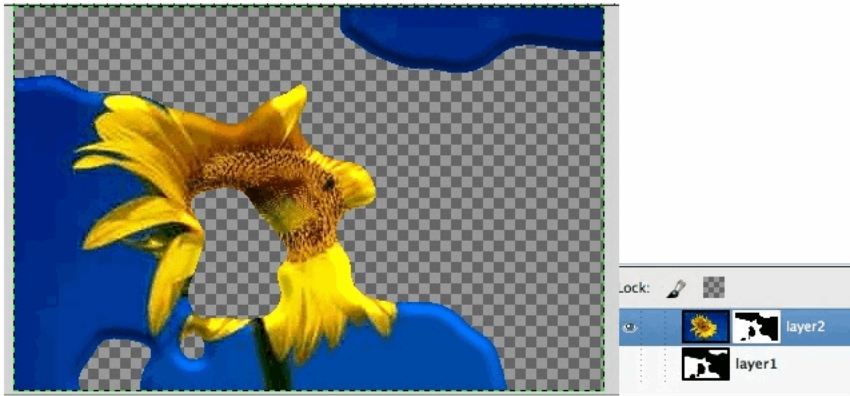
```

## =====BUMP\_FOR\_WET\_EFFECT=====

```

pdb.gimp_display_delete(display)           # Optional delete previous image

```



```
import random
filename = "/Users/don_sauer/Desktop/flower.gif"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
backgnd = pdb.gimp_file_load_layer(newImage, filename)
layer1 = pdb.gimp_image_get_active_layer(newImage)
display = pdb.gimp_display_new(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")
pdb.gimp_convert_rgb(newImage)
pdb.gimp_context_set_default_colors()

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0)

pdb.gimp_layer_add_alpha(layer2)
mask2 = pdb.gimp_layer_create_mask(layer2, 0) # W0,B1,A2,ATRANS3,SEL4,COPY5,CHAN6
pdb.gimp_layer_add_mask(layer2, mask2) #
pdb.gimp_item_set_visible(layer2, 0)

seed = random.randrange(65535)
pdb.plug_in_plasma(newImage, layer1, seed, 1) # "seed turb" (rand 65535) .2 ) )
pdb.gimp_desaturate(layer1)
pdb.gimp_brightness_contrast(layer1, 2, 127)
pdb.plug_in_gauss(newImage, layer1, 35, 35, 0) #
pdb.gimp_threshold(layer1, 127, 255) # "minlev maxlev" 130 255 ))

pdb.gimp_selection_all(newImage)
non_empty = pdb.gimp_edit_copy(layer1)
floating_sel = pdb.gimp_edit_paste(mask2, 1)
pdb.gimp_floating_sel_anchor(floating_sel)

pdb.gimp_selection_none(newImage)
pdb.gimp_by_color_select(layer1, (0,0,0), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_selection_grow(newImage, 5)
pdb.gimp_edit_bucket_fill(layer1, 0, 0, 100, 0, 0, 0, 0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL..)op,thres,merg, x, y
pdb.gimp_selection_none(newImage)
pdb.plug_in_gauss(newImage, layer1, 8, 8, 0) #

pdb.plug_in_bump_map(newImage, layer2, layer1, 74, 36, 5, 0, 0, 0, 129, 1, 0, 0)

pdb.gimp_item_set_visible(layer2, 1)
pdb.gimp_item_set_visible(layer1, 0)
```

## =====Color\_BlackWhite\_Image=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
```



```

newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")
pdb.gimp_desaturate(layer1)

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top

pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(layer2)
pdb.gimp_selection_none(newImage)
pdb.gimp_layer_set_mode(layer2, 13) # ADD7,SUB8,DK9,LI10,HUE11,SAT12,COL13,VAL14,DIV15

layer3 = pdb.gimp_layer_copy(layer2, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, 'layer3')
pdb.gimp_image_add_layer(newImage, layer3, 0) # adds layer on top

pdb.gimp_context_set_brush("Circle (11)")
pdb.gimp_context_set_foreground((0,255,255))
points = [82,106, 92,104]
pdb.gimp_paintbrush_default(layer2, len(points), points)

pdb.gimp_context_set_foreground((255,100,0))
points = [130,130,130,110]
pdb.gimp_paintbrush_default(layer3, len(points), points)

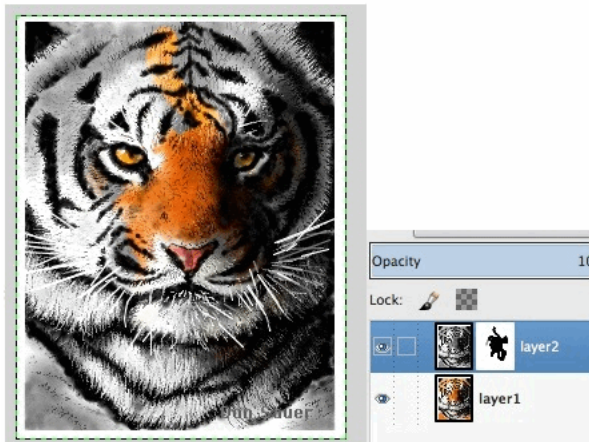
```

## =====PAINT\_BW\_ACCESS\_TO\_COLOR=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")
pdb.gimp_context_set_default_colors()

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top
pdb.gimp_desaturate(layer2)

mask2 = pdb.gimp_layer_create_mask(layer2,0) # W0,B1,A2,ATRANS3,SEL4,COPY5,CHAN6
pdb.gimp_layer_add_mask(layer2, mask2) #

activelayer = pdb.gimp_image_get_active_layer(newImage) # <gimp.Layer 'layer2'>

pdb.gimp_context_set_brush("Circle (19)")
points = [82,145, 200,90]
pdb.gimp_paintbrush_default(mask2, len(points), points)

```

## =====COPY\_A\_COLOR\_TO\_A\_LAYER=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```
filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")
pdb.gimp_context_set_default_colors()
pdb.gimp_convert_indexed(newImage, 0, 0, 4, 0, 0, "none")
pdb.gimp_convert_rgb(newImage)

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top

pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(layer2)
pdb.gimp_selection_none(newImage)

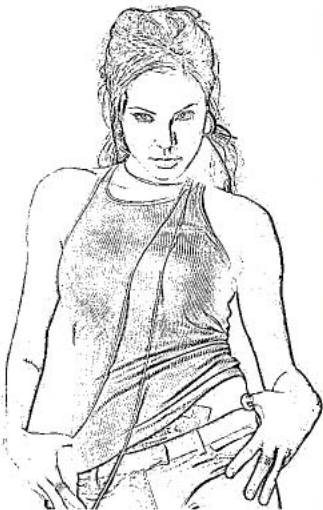
pdb.gimp_by_color_select(layer1, (233,152,70), 40, 0, 0, 0, 3, 0) # 40 threshold radius 3

non_empty = pdb.gimp_edit_copy(layer1)
floating_sel = pdb.gimp_edit_paste(layer2, 0)
pdb.gimp_floating_sel_anchor(floating_sel)

pdb.gimp_item_set_visible(layer1, 0)
```

## =====PENCIL\_DRAW=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
filename = "/Users/don_sauer/Desktop/AngelinaJolie.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top
pdb.gimp_desaturate(layer2)

layer3 = pdb.gimp_layer_copy(layer2, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, 'layer2')
pdb.gimp_image_add_layer(newImage, layer3, 0) # adds layer on top
pdb.gimp_invert(layer3)
```

```

pdb.plugin_gauss_iir(newImage, layer3, 5, 5, 5) #rad blurx blurry"
pdb.gimp_layer_set_opacity(layer3, 50)
layer2 = pdb.gimp_image_merge_down(newImage, layer3, 0)

layer3 = pdb.gimp_layer_copy(layer2, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, 'layer2')
pdb.gimp_image_add_layer(newImage, layer3, 0) # adds layer on top
pdb.gimp_layer_set_mode(layer2, 16) # DODG16,BURN17,HARD18,SOFT19,EXTR20,MERG21
layer2 = pdb.gimp_image_merge_down(newImage, layer3, 0)

pdb.plugin_unsharp_mask(newImage, layer2, 60, 10, 0) # "rad val thres" 60 10 0 )
pdb.gimp_brightness_contrast(layer2, 127, 57)

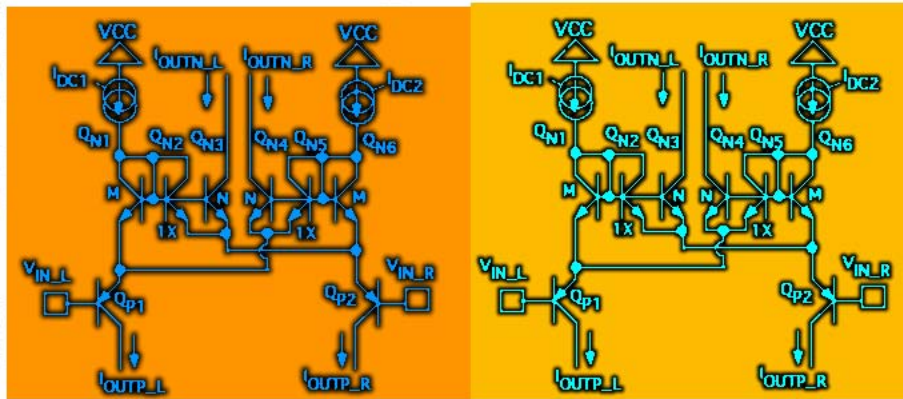
```

## =====OUTLINE\_WITH\_SHADE=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/AB_input.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")

pdb.gimp_context_set_default_colors()

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top
pdb.plugin_gauss(newImage, layer2, 4, 4, 0) #
pdb.gimp_brightness_contrast(layer2, -121, 127)
pdb.plugin_colortoalpha(newImage, layer2, (255,255,255))
pdb.plugin_gauss(newImage, layer2, 8, 8, 0) #

layer3 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, 'layer3')
pdb.gimp_image_add_layer(newImage, layer3, 0) # adds layer on top
pdb.plugin_gauss(newImage, layer3, 1, 1, 0) #
pdb.gimp_brightness_contrast(layer3, -94, 127)

pdb.gimp_context_set_foreground((0,150,255))
pdb.gimp_context_set_background((255,150,0))

pdb.gimp_by_color_select(layer3, (255,255,255), 20, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_edit_bucket_fill(layer1,1,0,100,0,0,0,0) # (FG,BG,PAT)(NOR)
pdb.gimp_edit_clear(layer3)
pdb.gimp_selection_invert(newImage)

pdb.gimp_edit_bucket_fill(layer3,0,0,100,0,0,0,0) # (FG,BG,PAT)(NOR)
pdb.gimp_selection_none(newImage)

```

## =====Fractalize\_Vector=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



w = 600

```

h = 600
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

vectorData = [24,179,33,173,42,167,102,85,138,110,174,135,183,207,216,202,249,197,272,146,311,152,350,
158,453,178,466,165,479,152,495,110,523,111,551,112]
numpts = len(vectorData)

vect1 = pdb.gimp_vectors_new(newImage, "vect1")
pdb.gimp_image_add_vectors(newImage, vect1, 0)
close = 0
stroke_id = pdb.gimp_vectors_stroke_new_from_points(vect1, 0,numpts, vectorData, close)
pdb.gimp_vectors_set_visible(vect1, 1)

```

```

pdb.fractalize_path(newImage, vect1, 5, 1, 1, 1, 50, 1)

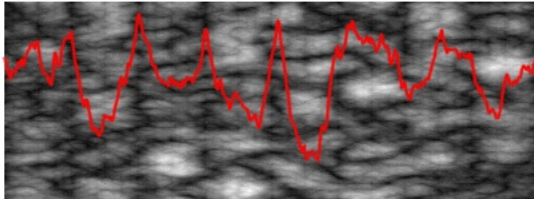
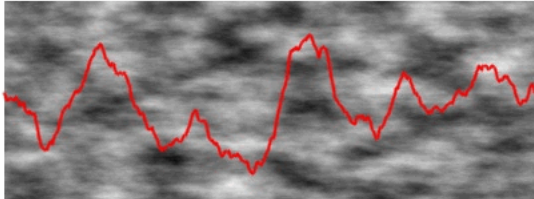
```

## =====PAINT\_SOLID\_NOISE\_WAVEFORMS=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

w = 500
h = 400

newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

pdb.gimp_rect_select(newImage, 50, 20, 400, 150, 0, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3
pdb.plug_in_solid_noise(newImage, layer1,1,0,233,15,8,8) # tile turb detail radx rady
pdb.gimp_selection_none(newImage)

pdb.gimp_rect_select(newImage, 50, 220, 400, 150, 0, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3
pdb.plug_in_solid_noise(newImage, layer1,1,1,233,15,8,8) # tile turb detail radx rady
pdb.gimp_selection_none(newImage)

pt = []
for i in range(50, 450):
    pt.append(i)
    num_channels, pixel = pdb.gimp_drawable_get_pixel(layer1,i, 100)
    pt.append(20+pixel[0]/1.5)

pdb.gimp_context_set_foreground((255,0,0))
pdb.gimp_context_set_paint_method('gimp-paintbrush')
name, width, height, spacing = pdb.gimp_brushes_get_brush()
pdb.gimp_brushes_set_brush('Circle (02)')
pdb.gimp_paintbrush_default(layer1, len(pt), pt)

pt = []
for i in range(50, 450):
    pt.append(i)
    num_channels, pixel = pdb.gimp_drawable_get_pixel(layer1,i, 300)
    pt.append(220+pixel[0]/1.5)

pdb.gimp_context_set_foreground((255,0,0))
pdb.gimp_context_set_paint_method('gimp-paintbrush')
name, width, height, spacing = pdb.gimp_brushes_get_brush()
pdb.gimp_brushes_set_brush('Circle (02)')
pdb.gimp_paintbrush_default(layer1, len(pt), pt)

```

## =====Gradients\_in\_Text=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```





```
newImage = pdb.gimp_image_new(300, 150, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage, 300, 150, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
w = pdb.gimp_drawable_width(layer1) # 300
h = pdb.gimp_drawable_height(layer1) # 150
pdb.gimp_invert(layer1) # have white layer

Txlayer = pdb.gimp_text_layer_new(newImage, "Effects", "Gill Sans Bold", 25, 2)
pdb.gimp_image_add_layer(newImage, Txlayer, 1) # Txlayer at position 1
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_lower_layer(newImage, layer1) # how to lower layer1
tw = pdb.gimp_drawable_width(Txlayer) # 243
th = pdb.gimp_drawable_height(Txlayer) # 83
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2) # center text
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)
pdb.gimp_by_color_select(Txlayer, (0,0,0), 10, 0, 0, 0, 1, 0) # select black

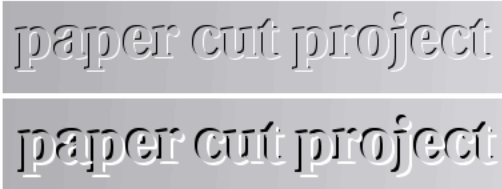
pdb.gimp_context_set_foreground((0,0,0))
pdb.gimp_context_set_background((255,0,0))
mode = 0 # FG-BG-RGB-MODE (0), FG-BG-HSV-MODE (1), FG-TRANSPARENT-MODE (2), CUSTOM-MODE (3) }
paint = 0 # NORMAL-MODE (0), DISSOLVE-MODE (1), BEHIND-MODE (2), M
type = 0 # GRADIENT-LINEAR (0), GRADIENT-BILINEAR (1), GRADIENT-RADIAL (2), GRA
opac = 100
offs = 0
repeat = 0 # REPEAT-NONE (0), REPEAT-SAWTOOTH (1), REPEAT-TRIANGULAR (2)
rev = 0
supersample = 0
depth = 1
thresh = 0
dith = 0
x1 = 20
y1 = 20
x2 = 150
y2 = 150
pdb.gimp_edit_blend(Txlayer, mode, paint, type, opac, offs, repeat, rev, supersample, depth, thresh, dith, x1, y1, x2, y2)

pdb.gimp_context_set_brush('Circle (03)')
pdb.gimp_context_set_brush_size(5)
name = pdb.gimp_context_get_paint_method()
pdb.gimp_context_set_paint_method('gimp-paintbrush')
pdb.gimp_edit_stroke(Txlayer)
pdb.gimp_selection_none(newImage)

pdb.gimp_image_set_active_layer(newImage, layer1)
pdb.gimp_selection_all(newImage)
pdb.gimp_layer_add_alpha(layer1)
pdb.plug_in_colortoalpha(newImage, layer1, (255,255,255))
```

## =====Paper\_Cut\_Text=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
w = 450
h = 250
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
w = pdb.gimp_drawable_width(layer1) # 300
h = pdb.gimp_drawable_height(layer1) # 150
pdb.gimp_invert(layer1) # have white layer
pdb.gimp_context_set_foreground((0, 0, 0))
pdb.gimp_context_set_background((255, 255, 255))

Txlayer = pdb.gimp_text_layer_new(newImage, "paper cut project", "Times Bold", 18, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0) # Txlayer at top
tw = pdb.gimp_drawable_width(Txlayer) # 243
th = pdb.gimp_drawable_height(Txlayer) # 83
```

```

pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2) # center text
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (35, 99)
pdb.gimp_selection_none(newImage)

Txlayer2 = pdb.gimp_layer_copy(Txlayer, 1)
pdb.gimp_layer_set_name(Txlayer2, "Txlayer2")
pdb.gimp_image_add_layer(newImage, Txlayer2, 0) # Txlayer2 at top
Txlayer3 = pdb.gimp_layer_copy(Txlayer, 1)
pdb.gimp_layer_set_name(Txlayer3, "Txlayer3")
pdb.gimp_image_add_layer(newImage, Txlayer3, 0) # Txlayer2 at top
pdb.gimp_selection_none(newImage)

pdb.gimp_context_set_foreground((175, 174, 178))
pdb.gimp_context_set_background((214, 213, 218))
mode = 0 # FG-BG-RGB-MODE (0), FG-BG-HSV-MODE (1), FG-TRANSPARENT-MODE (2), CUSTOM-MODE (3) }
paint = 0 # NORMAL-MODE (0), DISSOLVE-MODE (1), BEHIND-MODE (2), M
type = 0 # GRADIENT-LINEAR (0), GRADIENT-BILINEAR (1), GRADIENT-RADIAL (2), GRA
opac = 100
offs = 0
repeat = 0 # REPEAT-NONE (0), REPEAT-SAWTOOTH (1), REPEAT-TRIANGULAR (2)
rev = 0
supersample = 0
depth = 1
thresh = 0
dith = 0
x1 = 20
y1 = 125
x2 = 400
y2 = 150
pdb.gimp_edit_blend(layer1, mode, paint, type, opac, offs, repeat, rev, supersample, depth, thresh, dith, x1, y1, x2, y2)

pdb.gimp_by_color_select(Txlayer, (0,0,0), 10, 0, 0, 0, 0, 1, 0) # select black
pdb.gimp_edit_blend(Txlayer3, mode, paint, type, opac, offs, repeat, rev, supersample, depth, thresh, dith, x1, y1, x2, y2)

pdb.gimp_context_set_foreground((100, 100, 100))
fill_mode = 0 # FG0, BGL, PATTERN-2
pdb.gimp_edit_bucket_fill(Txlayer2, fill_mode, 0, 100, 0, 0, 0, 0, 0)
pdb.gimp_layer_translate(Txlayer2, -2, -2)
pdb.gimp_context_set_foreground((255, 255, 255))
fill_mode = 0 # FG0, BGL, PATTERN-2
pdb.gimp_edit_bucket_fill(Txlayer, fill_mode, 0, 100, 0, 0, 0, 0, 0)
pdb.gimp_layer_translate(Txlayer, 2, 2)
pdb.gimp_selection_none(newImage)

layer1 = pdb.gimp_image_merge_visible_layers(newImage, 0)

```

## =====Perspective\_Text=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

newImage = pdb.gimp_image_new(500, 300, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage, 500, 300, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_invert(layer1) # have white layer
pdb.gimp_context_set_foreground((0,0,0))
pdb.gimp_context_set_background((255,255,255))

Txlayer = pdb.gimp_text_layer_new(newImage, "Fencepost", "Gill Sans Bold", 30, 2)
pdb.gimp_image_add_layer(newImage, Txlayer, 1) # layer2 at position 1

pdb.gimp_image_lower_layer(newImage, layer1) # need layer1 lower than layer2
tw = pdb.gimp_drawable_width(Txlayer) # 441
th = pdb.gimp_drawable_height(Txlayer) # 98
pdb.gimp_layer_translate(Txlayer, (500-tw)/2, (300-th)/2)
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)

offp = 50
Txlayer = pdb.gimp_perspective(Txlayer, 1, offx+offp, offy, offx+tw-offp, offy, offx, offy+th, offx+tw, offy+th)
layer1 = pdb.gimp_image_merge_down(newImage, Txlayer, 0) # <gimp.Layer 'layer1'>

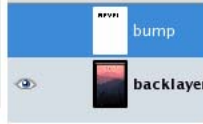
```

## =====BUMP\_Applications=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```
filename = "/Users/don_sauer/Downloads/2D0/GRAPHIC1/gimp_imageProcess/pict/colortest.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
Background = pdb.gimp_image_get_active_layer(newImage) # <gimp.Layer 'Background'>
height = pdb.gimp_image_height(newImage) # 318
width = pdb.gimp_image_width(newImage) # 248

wlayer = pdb.gimp_layer_new(newImage, width, height, 0, "wlayer", 100, 0)
pdb.gimp_image_add_layer(newImage, wlayer, 0) # layer added
pdb.gimp_invert(wlayer) # have White layer

Txlayer = pdb.gimp_text_layer_new(newImage, "BEVEL", "Gill Sans Bold", 30, 2)
pdb.gimp_image_add_layer(newImage, Txlayer, 0) # layer2 at position 1
pdb.gimp_layer_translate(Txlayer, 20, 70)

pdb.gimp_layer_set_visible(Background, 0) #
wlayer = pdb.gimp_image_merge_visible_layers(newImage, 0)
pdb.gimp_layer_set_visible(Background, 1) #

pdb.gimp_image_lower_layer(newImage, wlayer) # need layer1 lower than layer2

pdb.plugin_bump_map(newImage, Background, wlayer, 74, 36, 5, 0, 0, 0, 70, 1, 0, 0)
```

## =====Bevel\_Text=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
w = 300
h = 150
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_context_set_foreground((210,210,210))
pdb.gimp_context_set_background((255,255,255))
pdb.gimp_edit_fill(layer1, 0) # FG0,BG1,W2,TRAN3,PAT4,NONE5

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top
pdb.plugin_hsv_noise(newImage, layer2, 4, 81, 62, 62) # "hold hue sat val"

pdb.plugin_bump_map(newImage, layer1, layer2, 182, 52, 3, 0, 0, 0, 129, 0, 0, 0) # "azi elev dept x y wl amb"
pdb.gimp_item_set_visible(layer2, 0)

pdb.gimp_context_set_foreground((0,0,0))
Txlayer = pdb.gimp_text_layer_new(newImage, "BEVEL", "Gill Sans Bold", 20, 2)
pdb.gimp_image_add_layer(newImage, Txlayer, 0) # Txlayer at position 1
tw = pdb.gimp_drawable_width(Txlayer) # 243
th = pdb.gimp_drawable_height(Txlayer) # 83
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2) # center text
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)

Txlayer2 = pdb.gimp_layer_copy(Txlayer, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(Txlayer2, 'Txlayer2')
pdb.gimp_image_add_layer(newImage, Txlayer2, 0) # adds layer on top

pdb.gimp_by_color_select(Txlayer, (0,0,0), 10, 0, 0, 0, 0, 1, 0) # select black
pdb.gimp_selection_invert(newImage)
pdb.gimp_selection_grow(newImage, 1)
non_empty = pdb.gimp_edit_cut(Txlayer)
pdb.gimp_selection_none(newImage)

pdb.gimp_by_color_select(Txlayer, (0,0,0), 10, 0, 0, 0, 0, 1, 0) # select black
pdb.gimp_edit_bucket_fill(Txlayer, 1, 0, 100, 0, 0, 0, 0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL..)op,thres,merg, x, y
pdb.plugin_gauss_iir(newImage, Txlayer, 5, 5, 5) #rad blurx blurry"
pdb.gimp_layer_translate(Txlayer, 2, 2)
```

```
layer1 = pdb.gimp_image_merge_visible_layers(newImage,0)
pdb.gimp_selection_none(newImage)
```

## =====Western\_Text=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
w = 530
h = 300
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_context_set_foreground((167,62,67))
pdb.gimp_context_set_background((192,115,37))
pdb.gimp_edit_fill(layer1, 1)

# RGB0,GRAY1,INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added

# FG0,BG1,W2,TRAN3,PAT4,NONE5

Txlayer = pdb.gimp_text_layer_new(newImage, "SUGAR", "Times Bold", 45, 2)
pdb.gimp_image_add_layer(newImage, Txlayer, 0)
# Txlayer at position 1
# 243
# 83
# center text
tw = pdb.gimp_drawable_width(Txlayer)
th = pdb.gimp_drawable_height(Txlayer)
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2)
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)

pdb.gimp_by_color_select(Txlayer, (167,62,67), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_context_set_foreground((92,55,57))
pdb.gimp_context_set_brush('Circle (03)')
pdb.gimp_edit_stroke(Txlayer)

Txlayer2 = pdb.gimp_layer_copy(Txlayer, 1)
pdb.gimp_layer_set_name(Txlayer2, 'Txlayer2')
pdb.gimp_image_add_layer(newImage, Txlayer2, 0)
# layer2 copy of layer + alpha
# adds layer on top

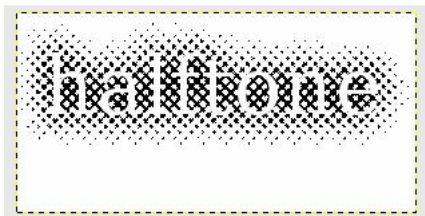
pdb.gimp_context_set_foreground((255,255,255))
pdb.gimp_selection_grow(newImage, 2)
pdb.gimp_edit_fill(Txlayer2, 0)
pdb.gimp_layer_translate(Txlayer2, 3, 3)
# FG0,BG1,W2,TRAN3,PAT4,NONE5

Txlayer3 = pdb.gimp_layer_copy(Txlayer2, 1)
pdb.gimp_layer_set_name(Txlayer3, 'Txlayer3')
pdb.gimp_image_add_layer(newImage, Txlayer3, 0)
# layer2 copy of layer + alpha
# adds layer on top
pdb.gimp_edit_clear(Txlayer3)

pdb.gimp_selection_shrink(newImage, 8)
pdb.gimp_context_set_foreground((92,55,57))
pdb.gimp_edit_fill(Txlayer3, 0)
# FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_selection_none(newImage)
layer1 = pdb.gimp_image_merge_visible_layers(newImage,0)
```

## =====HalfTone\_Text=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
w = 300
h = 150
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0)

# RGB0,GRAY1,INDEXED2
# <gimp.Layer 'layer1'>
```



```

display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_context_set_default_colors()
pdb.gimp_edit_fill(layer1, 1)

# opens window
# layer added
# FG0,BG1,W2,TRAN3,PAT4,NONE5

Txlayer = pdb.gimp_text_layer_new(newImage, "halftone", "Serif", 20, 2)
pdb.gimp_image_add_layer(newImage, Txlayer, 0)
tw = pdb.gimp_drawable_width(Txlayer)
th = pdb.gimp_drawable_height(Txlayer)
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2)
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)
# center text

Txlayer2 = pdb.gimp_layer_copy(Txlayer, 1)
pdb.gimp_layer_set_name(Txlayer2, 'Txlayer2')
pdb.gimp_image_add_layer(newImage, Txlayer2, 0)
# layer2 copy of layer + alpha
# adds layer on top

layer1 = pdb.gimp_image_merge_down(newImage, Txlayer, 0)
pdb.gimp_image_set_active_layer(newImage, layer1)
pdb.plugin_in_gauss(newImage, layer1, 35, 35, 0) #
pdb.gimp_brightness_contrast(layer1, -91, 83)
pdb.plugin_in_newsprint(newImage, layer1, 6, 2, 100, 45, 0, 45, 0, 45, 0, 45, 0, 1)

pdb.gimp_by_color_select(Txlayer2, (0,0,0), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_edit_fill(Txlayer2, 1) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_selection_none(newImage)

```

## =====CHISELED\_Text=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

w = 450
h = 150
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0) # RGB0,GRAY1,INDEXED2
display = pdb.gimp_display_new(newImage) # <gimp.Layer 'layer1'>
# opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_context_set_pattern("Pink Marble")
pdb.gimp_edit_fill(layer1, 4) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

Txlayer = pdb.gimp_text_layer_new(newImage, "CHISELED", "Baskerville Bold", 25, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0)
# Txlayer at position 1
tw = pdb.gimp_drawable_width(Txlayer)
th = pdb.gimp_drawable_height(Txlayer)
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2)
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)
# 243
# 83
# center text

Txlayer2 = pdb.gimp_layer_copy(Txlayer, 1)
pdb.gimp_layer_set_name(Txlayer2, 'Txlayer2')
pdb.gimp_image_add_layer(newImage, Txlayer2, 0)
# layer2 copy of layer + alpha
# adds layer on top

pdb.plugin_in_blur(newImage, Txlayer2)
pdb.gimp_layer_translate(Txlayer2, 4, 4)

Txlayer3 = pdb.gimp_layer_copy(Txlayer, 1)
pdb.gimp_layer_set_name(Txlayer3, 'Txlayer3')
pdb.gimp_image_add_layer(newImage, Txlayer3, 0)
# layer2 copy of layer + alpha
# adds layer on top

pdb.gimp_by_color_select(Txlayer, (0,0,0), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_edit_blend(Txlayer3, 0, 0, 7, 100, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0)
pdb.gimp_brightness_contrast(Txlayer3, 80, 68)

Txlayer4 = pdb.gimp_layer_copy(Txlayer, 1)
pdb.gimp_layer_set_name(Txlayer4, 'Txlayer4')
pdb.gimp_image_add_layer(newImage, Txlayer4, 0)
# layer2 copy of layer + alpha
# adds layer on top
pdb.gimp_edit_fill(Txlayer4, 4) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.plugin_in_bump_map(newImage, Txlayer4, Txlayer3, 47, 40, 30, 0, 0, 0, 100, 1, 0, 0)

layer1 = pdb.gimp_image_merge_visible_layers(newImage, 0)
pdb.gimp_selection_none(newImage)

```

## =====OUTLINED\_Text=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```
w = 300
h = 150
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_context_set_foreground((71, 185, 243))
pdb.gimp_context_set_background((13, 88, 207))
pdb.gimp_edit_fill(layer1, 1)

# RGB0, GRAY1, INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added

# FG0, BG1, W2, TRAN3, PAT4, NONES5

Translayer = pdb.gimp_layer_copy(layer1, 1)
pdb.gimp_layer_set_name(Translayer, 'Translayer')
pdb.gimp_image_add_layer(newImage, Translayer, 0)
pdb.gimp_layer_add_alpha(Translayer)

# layer2 copy of layer + alpha
# adds layer on top

pdb.gimp_edit_fill(Translayer, 3)
pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(Translayer)
pdb.gimp_selection_none(newImage)

# FG0, BG1, W2, TRAN3, PAT4, NONES5

Txlayer = pdb.gimp_text_layer_new(newImage, "OutLine", "Sans Bold", 20, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0)
tw = pdb.gimp_drawable_width(Txlayer)
th = pdb.gimp_drawable_height(Txlayer)
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2)
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)
Txlayer = pdb.gimp_image_merge_down(newImage, Txlayer, 0)

# Txlayer at position 1
# 243
# 83
# center text

Txlayer2 = pdb.gimp_layer_copy(Txlayer, 1)
pdb.gimp_layer_set_name(Txlayer2, 'Txlayer2')
pdb.gimp_image_add_layer(newImage, Txlayer2, 0)

# layer2 copy of layer + alpha
# adds layer on top

pdb.gimp_context_set_brush("Circle (03)")
thisbrush = pdb.gimp_context_get_brush("Circle (03)")
pdb.gimp_brush_set_hardness(thisbrush, 0.8)

pdb.gimp_context_set_default_colors()
pdb.gimp_by_color_select(Txlayer2, (71, 185, 243), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_image_set_active_layer(newImage, Txlayer2)
pdb.gimp_edit_stroke(Txlayer2)

pdb.gimp_selection_grow(newImage, 3)
pdb.gimp_edit_fill(Txlayer, 1)
pdb.plugin_gauss_iir(newImage, Txlayer, 7, 7, 7)

# FG0, BG1, W2, TRAN3, PAT4, NONES5
# rad blurx blurry"

layer1 = pdb.gimp_image_merge_visible_layers(newImage, 0)
pdb.gimp_selection_none(newImage)
```

```
=====BEVEL_Text=====
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
w = 730
h = 250
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_context_set_foreground((255, 0, 0))
pdb.gimp_context_set_background((0, 0, 0))
pdb.gimp_edit_blend(layer1, 0, 0, 7, 100, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0)

# RGB0, GRAY1, INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added

# BURSTANG6

pdb.gimp_context_set_default_colors()
Txlayer = pdb.gimp_text_layer_new(newImage, "BEVEL", "Gill Sans Bold", 50, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0)

# Txlayer at position 1
```

```

tw = pdb.gimp_drawable_width(Txlayer)          # 243
th = pdb.gimp_drawable_height(Txlayer)         # 83
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2) # center text
offx, offy = pdb.gimp_drawable_offsets(Txlayer)  # (20, 100)

pdb.gimp_by_color_select(Txlayer, (0,0,0), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3

pdb.gimp_edit_blend(Txlayer,0,0,7,100,0,0,0,0,1,0,1,10, 10, 20, 20) # BURSTANG6
pdb.gimp_invert(Txlayer)
pdb.gimp_brightness_contrast(Txlayer, -61, 36)
layer1 = pdb.gimp_image_merge_visible_layers(newImage,0)
pdb.gimp_selection_none(newImage)

```

## =====MIRROR\_Text=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

w = 600
h = 300
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

Txlayer = pdb.gimp_text_layer_new(newImage, "MIRROR TEXT", "Liberation Sans Bold", 25, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0) # Txlayer at position 1
tw = pdb.gimp_drawable_width(Txlayer) # 243
th = pdb.gimp_drawable_height(Txlayer) # 83
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2) # center text
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)

layer1 = pdb.gimp_image_merge_down(newImage, Txlayer, 0)

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top
pdb.gimp_by_color_select(layer2, (255,255,255), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_edit_clear(layer2)

layer3 = pdb.gimp_layer_copy(layer2, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, 'layer3')
pdb.gimp_image_add_layer(newImage, layer3, 0) # adds layer on top

layer4 = pdb.gimp_layer_copy(layer2, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer4, 'layer4')
pdb.gimp_image_add_layer(newImage, layer4, 0) # adds layer on top

pdb.gimp_selection_invert(newImage)
pdb.gimp_selection_grow(newImage, 2)
pdb.gimp_edit_bucket_fill(layer3,1,0,100,0,0,0,0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL..)op,thres,merg, x, y

pdb.gimp_selection_grow(newImage, 2)
pdb.gimp_edit_bucket_fill(layer2,0,0,100,0,0,0,0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL..)op,thres,merg, x, y
pdb.gimp_selection_none(newImage)

layer3 = pdb.gimp_image_merge_down(newImage, layer4, 0)
layer2 = pdb.gimp_image_merge_down(newImage, layer3, 0)

layer3 = pdb.gimp_layer_copy(layer2, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, 'layer3')
pdb.gimp_image_add_layer(newImage, layer3, 0) # adds layer on top

flip = 0 # HORIZONTAL (0), VERTICAL (1) }
autocenter = 1 #
axis = 500 #
clip = 0 #
layer2 = pdb.gimp_drawable_transform_flip_simple(layer2, 1, autocenter, axis, clip)
pdb.gimp_layer_translate(layer2, 0, 50)

pdb.gimp_context_swap_colors()
pdb.gimp_edit_blend(layer2,2,0,0,100,0,0,0,0,1,0,1,280,200,280,0) # LIN0,BILIN1,RAD),SQR3,CONSYM)

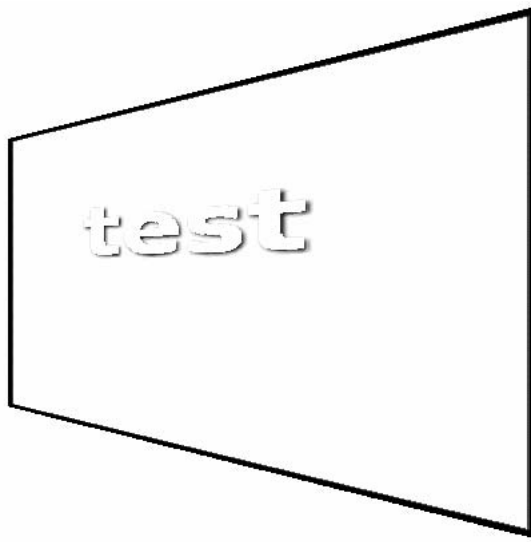
```

## =====PERSPECTIVE\_CUTOUT\_Text=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

w = 600
h = 600
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_edit_fill(layer1, 2)
pdb.gimp_context_set_default_colors()

layer2 = pdb.gimp_layer_copy(layer1, 1)
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0)

Txlayer = pdb.gimp_text_layer_new(newImage, "test", "Sans Bold", 50, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0)
tw = pdb.gimp_drawable_width(Txlayer)
th = pdb.gimp_drawable_height(Txlayer)
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2)
offx, offy = pdb.gimp_drawable_offsets(Txlayer)

layer2 = pdb.gimp_image_merge_down(newImage, Txlayer, 0)

layer3 = pdb.gimp_layer_copy(layer2, 1)
pdb.gimp_layer_set_name(layer3, 'layer3')
pdb.gimp_image_add_layer(newImage, layer3, 0)

pdb.gimp_by_color_select(layer3, (255,255,255), 60, 0, 0, 0, 3, 0)
pdb.gimp_edit_clear(layer3)
pdb.gimp_selection_invert(newImage)
pdb.gimp_edit_bucket_fill(layer3, 1, 0, 100, 0, 0, 0, 0)
pdb.gimp_layer_translate(layer3, -2, -2)

pdb.gimp_selection_all(newImage)
pdb.plugin_gauss_iir(newImage, layer2, 5, 5, 52)
layer2 = pdb.gimp_image_merge_down(newImage, layer3, 0)
pdb.gimp_selection_none(newImage)

pdb.gimp_rect_select(newImage, 100, 100, 400, 400, 0, 0, 0)
pdb.gimp_context_get_brush()
pdb.gimp_context_set_brush('Circle (05)')
name = pdb.gimp_context_get_paint_method()
name = pdb.gimp_context_set_paint_method('gimp-paintbrush')

pdb.gimp_edit_stroke(layer2)
pdb.gimp_selection_none(newImage)

offset = 50
x0 = 100
y0 = 100 + offset
x1 = 500
y1 = 100 - offset
x2 = 100
y2 = 500 - offset
x3 = 500
y3 = 500 + offset
dir = 0
inter = 3
ssamp = 0
lev = 1
clip = 0
layer2 = pdb.gimp_drawable_transform_perspective(layer2, x0, y0, x1, y1, x2, y2, x3, y3, dir, inter, ssamp, lev, clip)
layer1 = pdb.gimp_image_merge_visible_layers(newImage, 0)

# RGB0,GRAY1,INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added
# FG0,BG1,W2,TRAN3,PAT4,NONE5

# layer2 copy of layer + alpha
# adds layer on top

# Txlayer at position 1
# 243
# 83
# center text
# (20, 100)

# layer2 copy of layer + alpha
# adds layer on top

# 60 threshold radius 3

#(FG,BG,PAT)

# rad blurx blurry"

# "x y w h mode"); ADD0,SUB1,REP2,INTE3

# up left
# up right
# dwn left
# dwn right

# TRANSFORM-FORWARD (0), TRANSFORM-BACKWARD (1) }
# NONE (0), LINEAR (1), ICUBIC (2), LANCZOS (3) }

# RESIZE0,CLIP1,CROP2,CROP-WITH-ASPECT3

```

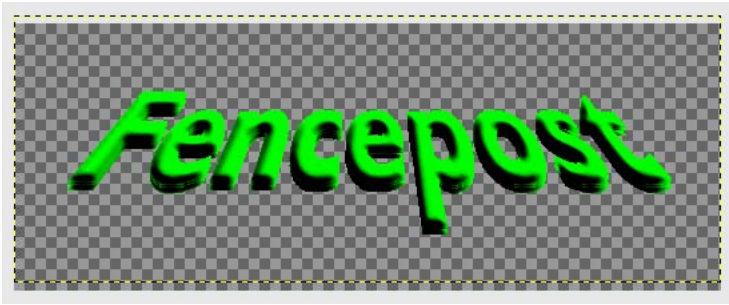
=====3D\_Perspective\_Text=====



```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

newImage = pdb.gimp_image_new(500, 300, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage, 500, 300, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_invert(layer1) # have white layer

Txlayer = pdb.gimp_text_layer_new(newImage, "Fencepost", "Gill Sans Bold", 30, 2)
pdb.gimp_image_add_layer(newImage, Txlayer, 1) # layer2 at position 1
pdb.gimp_context_set_foreground((0,0,0))
pdb.gimp_context_set_background((255,255,255))

pdb.gimp_image_lower_layer(newImage, layer1) # need layer1 lower than layer2
tw = pdb.gimp_drawable_width(Txlayer) # 441
th = pdb.gimp_drawable_height(Txlayer) # 98
pdb.gimp_layer_translate(Txlayer, (500-tw)/2, (300-th)/2)
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)

offp = 50
Txlayer = pdb.gimp_perspective(Txlayer, 1, offx+offp, offy, offx+tw-offp, offy, offx, offy+th, offx+tw, offy+th)
layer1 = pdb.gimp_image_merge_down(newImage, Txlayer, 0) # <gimp.Layer 'layer1'>

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top
pdb.plugin_gauss_iir(newImage, layer2, 7, 7, 7) #rad blurx blurx
pdb.gimp_invert(layer2)

layer3 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, 'layer3')
pdb.gimp_image_add_layer(newImage, layer3, 0) # adds layer on top
pdb.gimp_by_color_select(layer1, (255,255,255), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_edit_clear(layer3)

pdb.gimp_selection_invert(newImage)
pdb.gimp_context_set_foreground((0,255,0))
pdb.gimp_edit_bucket_fill(layer3, 0, 0, 100, 0, 0, 0, 0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL..)op,thres,merg, x, y
pdb.gimp_selection_none(newImage)

pdb.plugin_bump_map(newImage, layer3, layer2, 133, 24, 15, 0, 0, 0, 100, 1, 0, 0)

layer4 = pdb.gimp_layer_copy(layer3, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer4, 'layer4')
pdb.gimp_image_add_layer(newImage, layer4, 0) # adds layer on top
pdb.gimp_layer_translate(layer4, -1, -1)

layer5 = pdb.gimp_layer_copy(layer3, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer5, 'layer5')
pdb.gimp_image_add_layer(newImage, layer5, 0) # adds layer on top
pdb.gimp_layer_translate(layer5, -2, -2)

pdb.gimp_item_set_visible(layer1, 0)
pdb.gimp_item_set_visible(layer2, 0)

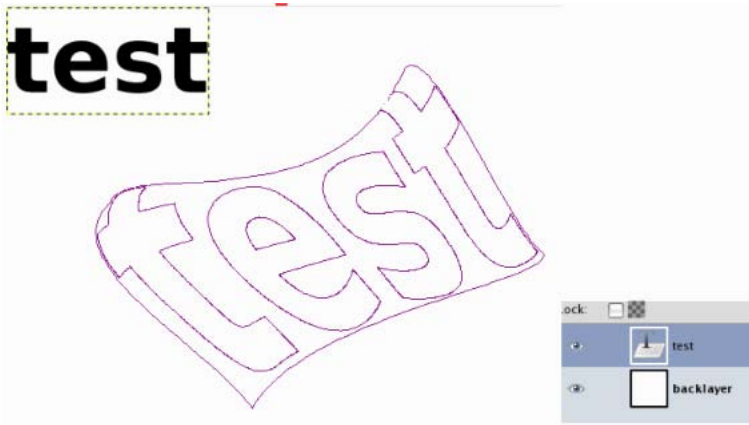
```

=====WARPED\_Text=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```
w = 600
h = 600
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_edit_fill(layer1, 2)
pdb.gimp_context_set_default_colors()

# RGB0, GRAY1, INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added
# FGO, BG1, W2, TRAN3, PAT4, NONE5

Txlayer = pdb.gimp_text_layer_new(newImage, "test", "Sans Bold", 30, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0) # Txlayer at position 1

vectorData = [41, 238, 96, 182, 115, 139, 306, 206, 372, 58, 395, 24, 469, 202, 500, 227, 531, 252, 278, 285, 230, 374, 180, 310]
numpts = len(vectorData)

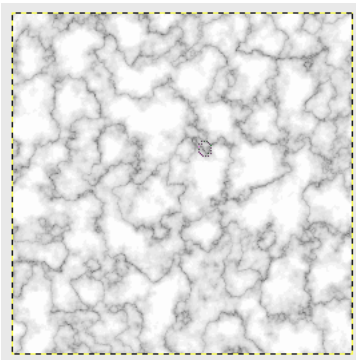
vect1 = pdb.gimp_vectors_new(newImage, "vect1")
pdb.gimp_image_add_vectors(newImage, vect1, 0)
close = 1
stroke_id = pdb.gimp_vectors_stroke_new_from_points(vect1, 0, numpts, vectorData, close)
pdb.gimp_vectors_set_visible(vect1, 1)

active_vectors = pdb.gimp_image_get_active_vectors(newImage) # <gimp.Vectors 'vect1'>
num_strokes, stroke_ids = pdb.gimp_vectors_get_strokes(active_vectors) # (1, (1,))

pdb.script_fu_sg_warp_text(newImage, Txlayer, 0, vect1, 0, 60) # <Image>/Filters/Distorts"
```

## =====MARBLE\_TEXTURE=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
newImage = pdb.gimp_image_new(256, 256, 0)
layer1 = pdb.gimp_layer_new(newImage, 256, 256, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_invert(layer1)

# RGB0, GRAY1, INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added
# have White layer

pdb.plugin_in_solid_noise(newImage, layer1, 1, 0, 233, 15, 8, 8)
layer2 = pdb.gimp_layer_copy(layer1, 1)
pdb.gimp_image_add_layer(newImage, layer2, 1)
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_invert(layer2)
pdb.gimp_layer_set_mode(layer2, 6)

# copy adds alpha
# layer2 at position 1
# 6 = diff

pdb.gimp_image_lower_layer(newImage, layer1)
layer1 = pdb.gimp_image_merge_down(newImage, layer2, 0)

# need layer1 lower than layer2
# <gimp.Layer 'layer1'>

pdb.gimp_levels(layer1, 0, 0, 109, 4.17, 0, 255)

layer2 = pdb.gimp_layer_copy(layer1, 1)
pdb.gimp_image_add_layer(newImage, layer2, 1)
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.plugin_in_rotate(newImage, layer1, 1, 0)
drawable = pdb.gimp_scale(layer1, 1, 0, 0, 256, 256)

# layer2 at position 1
```

```

pdb.gimp_image_lower_layer(newImage, layer1)
pdb.gimp_layer_set_mode(layer2, 4)
pdb.gimp_layer_set_opacity(layer2, 60)
layer1 = pdb.gimp_image_merge_down(newImage, layer2, 0) # <gimp.Layer 'layer1'>

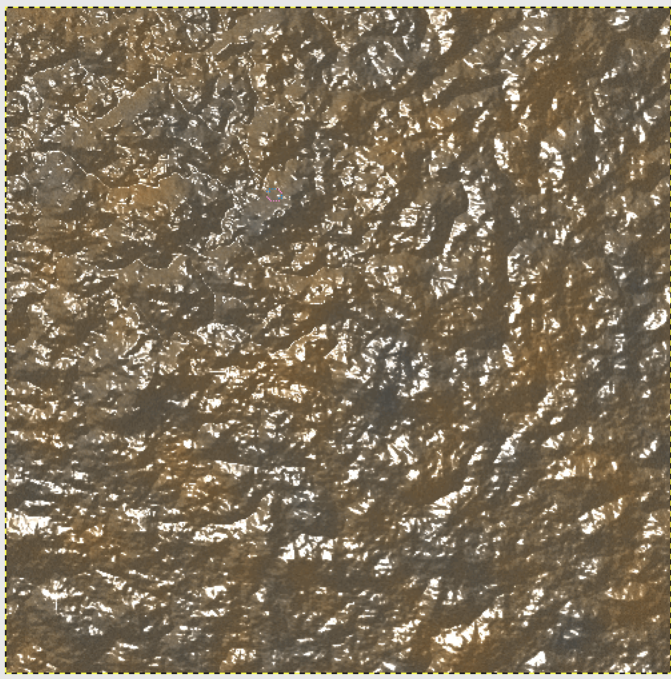
```

## =====WET\_STONE=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

newImage = pdb.gimp_image_new(400, 400, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,400,400,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_invert(layer1) # have White layer

pdb.gimp_context_set_foreground((75,77,77))
pdb.gimp_context_set_background((121,91,50))

pdb.plugin_solid_noise(newImage, layer1,1,0,233,15,8,8)
pdb.gimp_context_set_gradient("FG to BG (RGB)")
pdb.plugin_gradmap(newImage, layer1)

pdb.plugin_rgb_noise(newImage, layer1, 0, 1, 0.5, 0.5, 0.5, 0)

layer2 = pdb.gimp_layer_copy(layer1, 1) # add alpha
pdb.gimp_image_add_layer(newImage, layer2, 1) # layer2 at position 1
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.plugin_solid_noise(newImage, layer2,1,0,233,15,8,8)

layer3 = pdb.gimp_layer_copy(layer2, 1)
pdb.gimp_image_add_layer(newImage, layer3, 2) # layer2 at position 1
pdb.gimp_layer_set_name(layer3, 'layer3')
pdb.gimp_invert(layer3)
pdb.gimp_layer_set_mode(layer3, 6)
pdb.gimp_image_lower_layer(newImage, layer2)
layer2 = pdb.gimp_image_merge_down(newImage, layer3, 0) # <gimp.Layer 'layer1'>

pdb.gimp_image_set_active_layer(newImage, layer1)

num_images, image_ids = pdb.gimp_image_list() # (1, (33,))
num_layers, layer_ids = pdb.gimp_image_get_layers(newImage) # (2, (84, 87))
codeforAbove = "(plug-in-lighting 1 33 84 87 0 1 0 0 0 '(255 255 255) -1 -1 1 -1 -1 1 1 .21 1 .94 22 1 1 0)"

test = "(plug-in-lighting 1 "+ str(image_ids[0])+" " + str(layer_ids[0])+" " +str(layer_ids[1])+" "
test = test +" 0 1 0 0 0 '(255 255 255) -1 -1 1 -1 -1 1 1 .21 1 .94 22 1 1 0)"
test # "(plug-in-lighting 1 33 84 87 0 1 0 0 0 '(255 255 255) -1 -1 1 -1 -1 1 1 .21 1 .94 22 1 1 0)"

pdb.plugin_script_fu_eval(test)

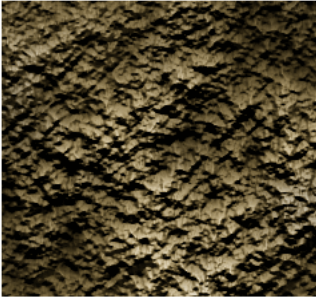
```

## =====ROUGH\_STONE=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```
newImage = pdb.gimp_image_new(400, 400, 0)
layer1 = pdb.gimp_layer_new(newImage,400,400,0,'layer1',100,0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_invert(layer1)

# RGB0,GRAY1,INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added
# have White layer

pdb.gimp_context_set_foreground((0,0,0))
pdb.gimp_context_set_background((255,255,255))

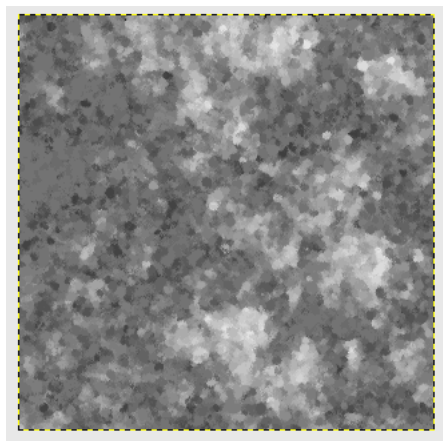
pdb.plug_in_solid_noise(newImage, layer1,1,0,233,15,4,4)
pdb.plug_in_unsharp_mask(newImage, layer1, 5, 3.5, 0)
pdb.plug_in_emboss(newImage, layer1, 90, 40, 20, 0)
pdb.plug_in_make_seamless(newImage, layer1)

colorlay = pdb.gimp_layer_copy(layer1, 1)
pdb.gimp_image_add_layer(newImage, colorlay, 2)
pdb.gimp_layer_set_name(colorlay, 'colorlay')
pdb.gimp_invert(colorlay)
# layer2 at position 1
pdb.plug_in_solid_noise(newImage, colorlay ,1,0,233,2,4,4)
pdb.gimp_layer_set_mode(colorlay, 5)
pdb.gimp_brightness_contrast(colorlay, 0, -50)
pdb.gimp_colorize(colorlay, 42, 50, -8)

pdb.gimp_image_lower_layer(newImage, layer1)
```

## =====ZINC\_PLATED=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
newImage = pdb.gimp_image_new(300, 300, 0)
layer1 = pdb.gimp_layer_new(newImage,300,300,0,'layer1',100,0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_invert(layer1)

# RGB0,GRAY1,INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added
# have White layer

pdb.gimp_context_set_foreground((0,0,0))
pdb.gimp_context_set_background((255,255,255))

pdb.plug_in_plasma(newImage, layer1, 566, 7)
pdb.gimp_desaturate(layer1)

pdb.gimp_by_color_select(layer1, (0,0,0), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3

pdb.gimp_brightness_contrast(layer1, 69, -56)
pdb.gimp_selection_none(newImage)
pdb.plug_in_randomize_slur(newImage, layer1, 1,17, 10, 11)
pdb.plug_in_oilify(newImage, layer1, 9, 1)
```

## =====REFLECTIVE\_GRADIENTS=====



```

pdb.gimp_display_delete(display)          # Optional delete previous image

```



```

import random
from gimpfu import *
import sys,random as r
import math, string

w = 1000
h = 1000
newImage = pdb.gimp_image_new(w, h, 0)          # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0)    # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage)        # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0)    # layer added
pdb.gimp_edit_fill(layer1, 2)                  # FG0,BG1,W2,TRAN3,PAT4,NONE5

pdb.gimp_context_set_default_colors()

blend=0 # FG-BG-RGB-MODE (0), FG-BG-HSV-MODE (1), FG-TRANSPARENT-MODE (2), CUSTOM-MODE (3) }
paint=6 # NORMAL0 DIFFERENCE-MODE (6
grad=0 # LIN0,BILIN1,RAD2,SQ3,SYMME4,ASYMME5,ANGU6,SPHER7,DIMPL8,CLOCK9,CCLOCK10
opac=100 #
off=0 #
repet=0 # NONE (0), REPEAT-SAWTOOTH (1), REPEAT-TRIANGULAR (2) }
rev=0 #
ssamp=0 #
dep=1 #
thres=0 #
dith=0 #

for i in range(30):
    x1 = random.randrange(0, 1000) #
    y1 = random.randrange(0, 1000) #
    x2 = random.randrange(0, 1000) #
    y2 = random.randrange(0, 1000) #
    pdb.gimp_edit_blend(layer1,blend,paint,grad,opac,off,repet,rev,ssamp,dep,thres,dith, x1, y1, x2, y2)

layer2 = pdb.gimp_layer_copy(layer1, 1)          # layer2 copy of layer1 + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0)    # adds layer on top

flip = 0 # HORIZONTAL (0), VERTICAL (1) }
autocenter = 1 #
axis = 500 #
clip = 0 #
layer2 = pdb.gimp_drawable_transform_flip_simple(layer2, flip, autocenter, axis, clip)
pdb.gimp_layer_set_mode(layer2, 10) # NORM0,DISS1,BEHIND2,MULT3,SCRN4,OVERL5,DIFF6,ADD7,SUB8,DRK9,LI10
layer1 = pdb.gimp_image_merge_down(newImage, layer2, 0)

layer2 = pdb.gimp_layer_copy(layer1, 1)          # layer2 copy of layer1 + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0)    # adds layer on top

flip = 1 # HORIZONTAL (0), VERTICAL (1) }
layer2 = pdb.gimp_drawable_transform_flip_simple(layer2, flip, autocenter, axis, clip)
pdb.gimp_layer_set_mode(layer2, 10) # NORM0,DISS1,BEHIND2,MULT3,SCRN4,OVERL5,DIFF6,ADD7,SUB8,DRK9,LI10
layer1 = pdb.gimp_image_merge_down(newImage, layer2, 0)

layer2 = pdb.gimp_layer_copy(layer1, 1)          # layer2 copy of layer1 + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0)    # adds layer on top

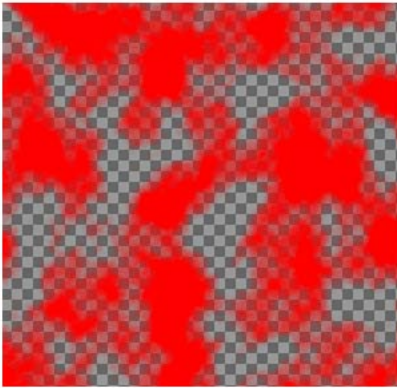
pdb.gimp_context_set_foreground((255,255,255))
pdb.gimp_context_set_background((0,0,255))

blend=0 # FG-BG-RGB-MODE (0), FG-BG-HSV-MODE (1), FG-TRANSPARENT-MODE (2), CUSTOM-MODE (3) }
paint=0 # NORMAL0 DIFFERENCE-MODE (6
grad=2 # LIN0,BILIN1,RAD2,SQ3,SYMME4,ASYMME5,ANGU6,SPHER7,DIMPL8,CLOCK9,CCLOCK10
opac=100 #
off=0 #
repet=0 # NONE (0), REPEAT-SAWTOOTH (1), REPEAT-TRIANGULAR (2) }
rev=0 #
ssamp=0 #
dep=1 #
thres=0 #
dith=1 #
x1 = 500 #
y1 = 500 #
x2 = 0 #
y2 = 0 #
pdb.gimp_edit_blend(layer2,blend,paint,grad,opac,off,repet,rev,ssamp,dep,thres,dith, x1, y1, x2, y2)
pdb.gimp_layer_set_mode(layer2, 17) #;DODG16,BURN17,HARD18,SOFT19,EXTR20,MERG21
layer1 = pdb.gimp_image_merge_down(newImage, layer2, 0)

```

=====PERLIN\_FOG=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
newImage = pdb.gimp_image_new(400, 400, 0)
layer1 = pdb.gimp_layer_new(newImage,400,400,0,'layer1',100,0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_layer_add_alpha(layer1)
has_alpha = pdb.gimp_drawable_has_alpha(layer1)
pdb.gimp_edit_fill(layer1, 3)
pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(layer1)
pdb.gimp_selection_none(newImage)

layer2 = pdb.gimp_layer_copy(layer1, 1)
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0)
pdb.plug_in_solid_noise(newImage, layer2,1,0,233,15,8,8)

pdb.gimp_levels(layer2, 0, 89, 160, 1, 0, 255)
pdb.gimp_item_set_visible(layer2, 0)

layer3 = pdb.gimp_layer_copy(layer1, 1)
pdb.gimp_layer_set_name(layer3, 'layer3')
pdb.gimp_image_add_layer(newImage, layer3, 0)
pdb.gimp_context_set_foreground((255,0,0))
pdb.gimp_edit_fill(layer3, 0)

mask3 = pdb.gimp_layer_create_mask(layer3,0)
pdb.gimp_layer_add_mask(layer3, mask3)

pdb.gimp_selection_all(newImage)
non_empty = pdb.gimp_edit_copy(layer2)
floating_sel = pdb.gimp_edit_paste(mask3, 0)
pdb.gimp_floating_sel_anchor(floating_sel)
```

# RGB0,GRAY1,INDEXED2  
# <gimp.Layer 'layer1'>  
# opens window  
# layer added  
# ??? FG0,BG1,W2,TRAN3,PAT4,NONE5  
# layer2 copy of layer + alpha  
# adds layer on top  
# tile turb detail radx rady  
# RGB= 0 lowin hiin gamma loout hout  
# layer2 copy of layer + alpha  
# adds layer on top  
# FG0,BG1,W2,TRAN3,PAT4,NONE5  
# WHITE0,BLK1,ALP2,ATRANS3,SEL4,COPY5,CHAN6

## =====SIMPSON\_CLOUDS=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
newImage = pdb.gimp_image_new(400, 400, 0)
layer1 = pdb.gimp_layer_new(newImage,400,400,0,'layer1',100,0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_layer_add_alpha(layer1)
has_alpha = pdb.gimp_drawable_has_alpha(layer1)
pdb.gimp_edit_fill(layer1, 3)
pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(layer1)
pdb.gimp_selection_none(newImage)

# =====SolidNoise=====
layer2 = pdb.gimp_layer_copy(layer1, 1)
```

# RGB0,GRAY1,INDEXED2  
# <gimp.Layer 'layer1'>  
# opens window  
# layer added  
# ??? FG0,BG1,W2,TRAN3,PAT4,NONE5  
# layer2 copy of layer + alpha

```

pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0)
pdb.plug_in_solid_noise(newImage, layer2, 1, 0, 233, 15, 8, 8)
pdb.plug_in_mblur(newImage, layer2, 0, 111, 0, 0, 0)

# adds layer on top
# tile turb detail radx rady
# "typ len ang centx centy");lin0,rad1

pdb.gimp_levels(layer2, 0, 89, 160, 1, 0, 255)
pdb.gimp_item_set_visible(layer2, 0)

# RGB= 0 lowin hiin gamma loout hout

layer3 = pdb.gimp_layer_copy(layer1, 1)
# layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, 'layer3')
# adds layer on top
pdb.gimp_image_add_layer(newImage, layer3, 0)
pdb.gimp_context_set_foreground((255,0,0))
# FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_edit_fill(layer3, 0)
# WHIT0,BLK1,ALP2,ATRANS3,SEL4,COPY5,CHAN6

mask3 = pdb.gimp_layer_create_mask(layer3, 0)
pdb.gimp_layer_add_mask(layer3, mask3)

pdb.gimp_selection_all(newImage)
non_empty = pdb.gimp_edit_copy(layer2)
floating_sel = pdb.gimp_edit_paste(mask3, 0)
pdb.gimp_floating_sel_anchor(floating_sel)

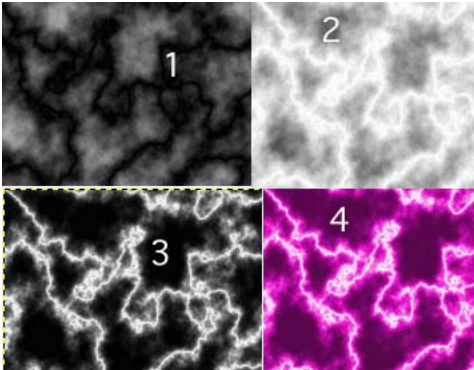
```

## =====ELECTRIC=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

w = 400
h = 400
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
# RGB0,GRAY1,INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_context_set_default_colors()
# FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_edit_fill(layer1, 1)

pdb.plug_in_solid_noise(newImage, layer1, 1, 0, 233, 15, 8, 8)
# tile turb detail radx rady
pdb.gimp_layer_add_alpha(layer1)

layer2 = pdb.gimp_layer_copy(layer1, 1)
# layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
# adds layer on top
pdb.gimp_image_add_layer(newImage, layer2, 0)
pdb.gimp_invert(layer2)
# NORM0,DISS1,BEHN2,MULT3,SCREE4,OVLAY5,DIFF6,ADD7,SUB8
pdb.gimp_layer_set_mode(layer2, 6)

layer1 = pdb.gimp_image_merge_down(newImage, layer2, 0)
pdb.gimp_invert(layer1)

pdb.gimp_levels(layer1, 0, 0, 255, .12, 0, 255)
# RGB= 0 lowin hiin gamma loout hout
pdb.gimp_colorize(layer1, 308, 85, 19) # hue sat lit 308 85 19

```

## =====CREATE\_PINE=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

w = 600
h = 600
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_context_set_foreground((156,114,54))
pdb.gimp_context_set_background((200,172,95))
pdb.gimp_edit_fill(layer1, 1)

pdb.gimp_brushes_set_brush("ahair brush")
points = [100,33, 200,568]
pdb.gimp_paintbrush_default(layer1, len(points), points)
points = [180,33,300,568]
pdb.gimp_paintbrush_default(layer1, len(points), points)
points = [280,33,300,568]
pdb.gimp_paintbrush_default(layer1, len(points), points)
points = [400,33,500,568]
pdb.gimp_paintbrush_default(layer1, len(points), points)

pdb.plug_in_ripple(newImage, layer1, 179, 98, 0, 1,1, 0, 0) # "period ampl orient (H0,V1) (sin1)
pdb.plug_in_rgb_noise(newImage, layer1, 0, 1, 0.5, 0.5, 0.5, 0)

pdb.gimp_image_rotate(newImage, 0) # 90 (0), 180 (1), 270 (2) }
pdb.plug_in_mblur(newImage, layer1, 0, 70, 90, 0, 0) # "typ len ang centx centy);lin0,rad1

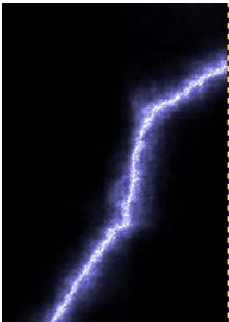
```

## =====Lighting\_Simulation=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

import random
w = 300
h = 300
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_edit_fill(layer1, 2)

pdb.gimp_context_set_foreground((0,0,0))
pdb.gimp_context_set_background((191,191,191))

blend = 0 # FG-BG-RGB0,FG-BG-HSV1,FG-TRANS2,CUSTOM3
paint = 0 # NOR0,DIS1,BHIN2,MUL3,SCRN4,OVER5,DIFF6,ADD7,SUBTRACT-MOD
grad = 0 # LIN0,BILIN1,RAD),SQR3,CONSYM),CONASYM5,BURSTANG6,BURSTSPHER7,BURSTDIMP8,CLOCK9, CCLOCK10
opac = 100
off = 0
repeat = 0
rev = 0
ssamp = 0
dep = 0
thres = 0
dith = 0
x1 = 0

```



```

y1 = 0
x2 = 300
y2 = 300
pdb.gimp_edit_blend(layer1,blend,paint,grad,opac,off,repeat,rev,ssamp,dep,thres,dith,x1,y1,x2,y2)

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top

seed = random.randrange(65535)
pdb.plug_in_plasma(newImage, layer2, seed, .2) # "seed turb" (rand 65535) .2 )
pdb.gimp_desaturate(layer2)

pdb.gimp_layer_set_mode(layer2, 6) # NOR0,DIS1,BHIN2,MUL3,SCRN4,OVER5,DIFF6,ADD7,SUBTRACT-MOD
layer1 = pdb.gimp_image_merge_down(newImage, layer2, 0)

pdb.gimp_invert(layer1)
pdb.gimp_levels(layer1, 0, 184, 255, 0.1, 0, 255) # RGB= 0 lowin hiin gamma loout hout
pdb.gimp_color_balance(layer1,1, 1, 0,0,100) #lev (0->2 Shad=>lit) cya2red mag2grn yel2blu"

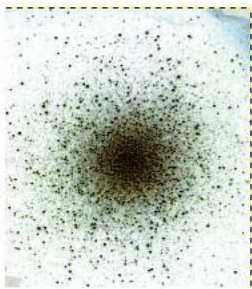
```

## =====SIMULATE\_A\_GLOBULAR=====

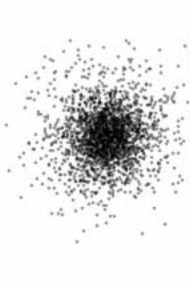
```

pdb.gimp_display_delete(display) # Optional delete previous image

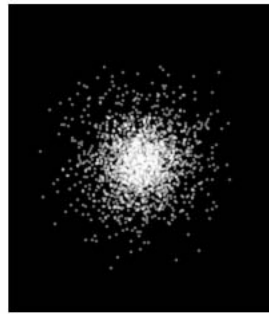
```



Real



Simulated



inverted color

```

import random
w = 300
h = 300
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FGO,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

pdb.gimp.context_set_brush('Circle (02)')
name, width, height, spacing = pdb.gimp_brushes_get_brush() # ('Circle (05)', 3, 3, 10)

points = [w/2,h/2, w/2,h/2]
points2 = [w/2,h/2, w/2,h/2]

for i in range(1000):
    points[0] = random.normalvariate(150, 30)
    points[1] = random.normalvariate(150, 30)
    points[2] = points[0]
    points[3] = points[1]
    points2[0] = random.normalvariate(150, 15)
    points2[1] = random.normalvariate(150, 15)
    points2[2] = points2[0]
    points2[3] = points2[1]
    pdb.gimp_paintbrush_default(layer1, len(points), points)
    pdb.gimp_paintbrush_default(layer1, len(points2), points2)

points = [w/2,h/2, w/2,h/2]
pdb.gimp_paintbrush_default(layer1, len(points), points)

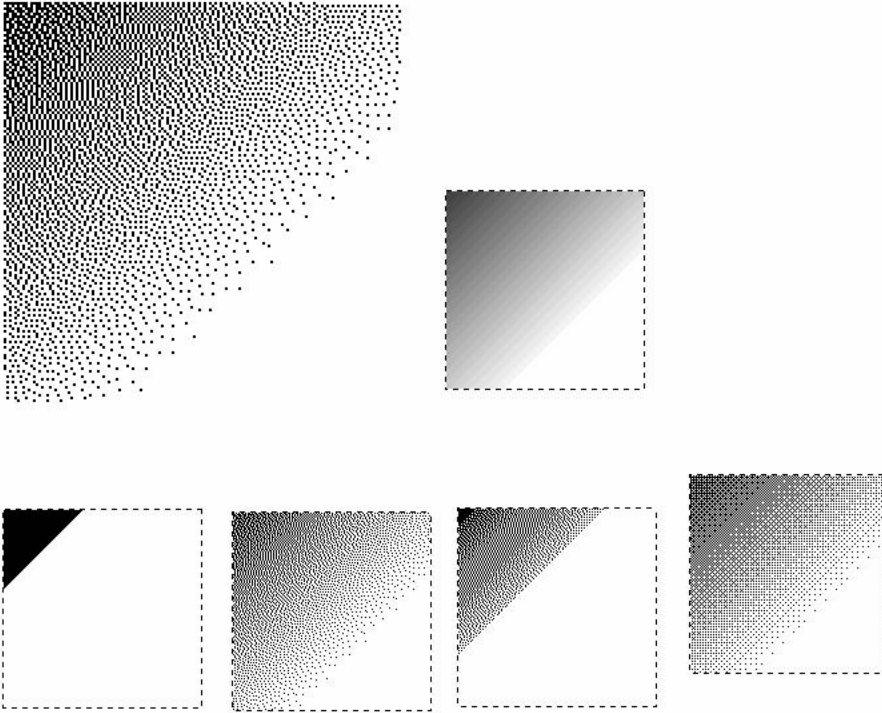
```

## =====DITHER=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



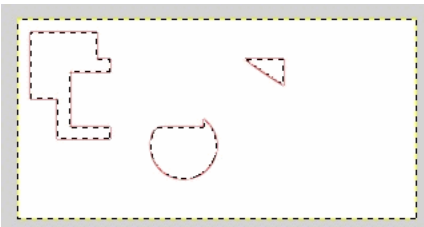
```
w = 300
h = 300
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

pdb.gimp_rect_select(newImage, 50, 50, 150, 150, 0, 0, 0) # "x y w h mode"; ADD0,SUB1,REP2,INTE3
pdb.gimp_edit_blend(layer1,0,0,0,100,0,0,0,0,1,0,1,20,20,150,150) # LIN0,BILIN1,RAD),SQR3,CONSYM)

pdb.gimp_selection_none(newImage)
dithertype = 1 # { NO-DITHER (0), FS-DITHER (1), FLOWBLEED-DITHER (2), FIXED-DITHER (3) }
palettetype = 3 # MAKE-PALETTE (0), WEB-PALETTE (2), MONO-PALETTE (3), CUSTOM-PALETTE (4) }
numcols = 1
alphadither = 0
remo_unused = 0
palette = ""
pdb.gimp_convert_indexed(newImage, dithertype, palettetype,numcols,alphadither, remo_unused, palette)
```

## =====SELECTION\_TYPES=====

```
pdb.gimp_display_delete(display) # Optional delete previous image
```



```
w = 300
h = 150
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

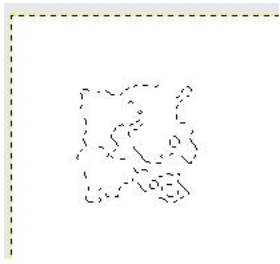
pdb.gimp_rect_select(newImage, 10, 10, 50, 50, 0, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3
pdb.gimp_rect_select(newImage, 30, 30, 40, 60, 0, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3
pdb.gimp_ellipse_select(newImage, 100, 70, 50, 50, 0, 0, 0)
pdb.gimp_rect_select(newImage, 40, 40,100, 41, 1, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3

pts2 = [170,30,200,30,200,50,170,30]
pdb.gimp_image_select_polygon(newImage, 0,len(pts2), pts2)

pdb.plugin_sel2path(newImage, layer1)
active_vectors = pdb.gimp_image_get_active_vectors(newImage)
pdb.gimp_vectors_set_visible(active_vectors, 1)
```

## =====SELECTION\_DISTORTION=====

pdb.gimp\_display\_delete(display) # Optional delete previous image



```
w = 300
h = 300
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top
pdb.gimp_rect_select(newImage, 50, 50, 100, 100, 0, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3

# NOR0,DIS1,BHIN2,MUL3),SCRN4,OVERL5,DIFF6,ADD7,SUB8,Dk9,LI10,HUE11,SAT1,COL13,VAL14,DIV15,
# DGE16,BURN17,HAR18,SOFT19,EXTR20,MERGE21, COLERASE22,ERASE23,REPLACE24,ANTIERASE25
pdb.gimp_edit_bucket_fill(layer2,0,0,100,0,0,0,0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL..)op,thres,merg, x, y
pdb.gimp_selection_none(newImage)

pdb.plug_in_spread(newImage, layer2, 200, 200)
pdb.plug_in_gauss_iir(newImage, layer2, 12, 12, 12) #rad blurx blurry"
pdb.gimp_threshold(layer2, 130, 255) # "minlev maxlev" 130 255 ))
pdb.gimp_by_color_select(layer2, (0,0,0), 10, 0, 0, 0, 3, 0) # 10 threshold radius 3

pdb.gimp_image_remove_layer(newImage, layer2)
```

## =====SELECT\_DISTRESS=====

pdb.gimp\_display\_delete(display) # Optional delete previous image



```
w = 300
h = 300
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

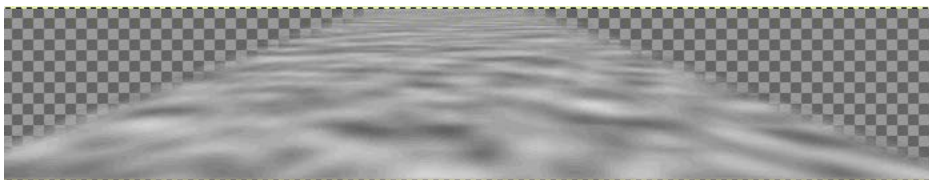
layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top

pdb.gimp_rect_select(newImage, 50, 50, 100, 100, 0, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3

pdb.script_fu_distress_selection(newImage, layer2, 127, 8, 4, 1, 1, 1) # thres,sprd,gran,sh,sv
pdb.gimp_edit_bucket_fill(layer2,0,0,100,0,0,0,0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL..)op,thres,merg, x, y
pdb.gimp_selection_none(newImage)
```

## =====PERSPECTIVE\_WAVES=====

pdb.gimp\_display\_delete(display) # Optional delete previous image



```

w = 700
h = 700
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0)
display = pdb.gimp_display_new(newImage)
pdb.gimp_image_add_layer(newImage, layer1, 0)
pdb.gimp_edit_fill(layer1, 2)
pdb.gimp_context_set_default_colors()

# RGB0,GRAY1,INDEXED2
# <gimp.Layer 'layer1'>
# opens window
# layer added
# FGO,BG1,W2,TRAN3,PAT4,NONE5

pdb.plug_in_solid_noise(newImage, layer1, 1, 0, 233, 15, 8, 8)
pdb.plug_in_gauss_iir(newImage, layer1, 12, 12, 12)
pdb.plug_in_emboss(newImage, layer1, 90, 50, 5, 1)
layer1 = pdb.gimp_item_transform_perspective(layer1, 300, 0, 400, 0, 0, 700, 700, 700)
pdb.gimp_image_scale_full(newImage, 800, 130, 1) # w,h,inter
pdb.plug_in_mblur(newImage, layer1, 0, 35, 0, 0, 0)
# tile turb detail radx rady
# rad blurx blurry"
# "azi elev depth embos" 90 40 5 1) )
# "typ len ang centx centy");lin0,radi

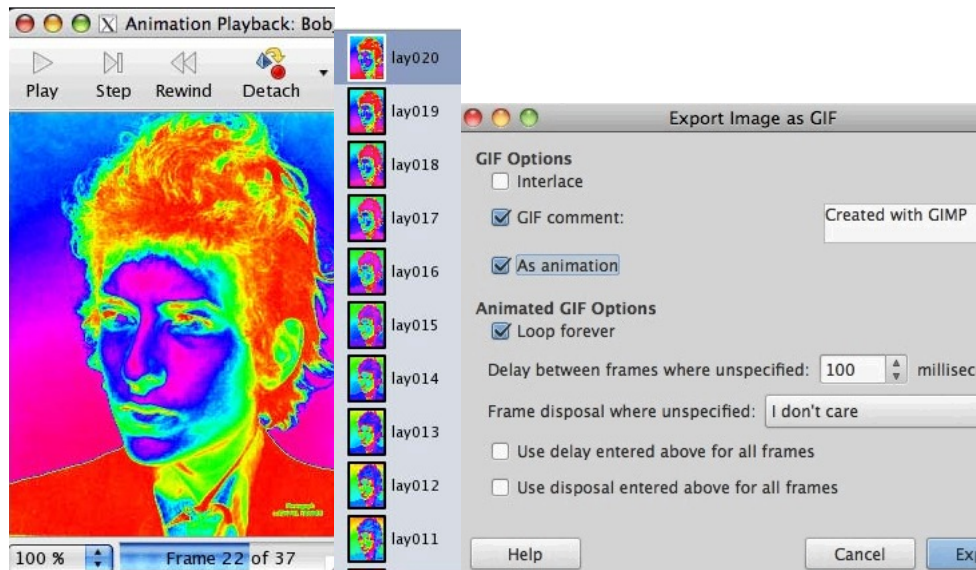
```

## =====Make\_Rainbow\_Gradient\_Movie=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/Bob_Dylan.jpg"
rawFilename = "Dylan.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
backgnd = pdb.gimp_file_load_layer(newImage, filename)
layer1 = pdb.gimp_image_get_active_layer(newImage)
display = pdb.gimp_display_new(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")
pdb.gimp_convert_rgb(newImage)

pdb.gimp_gradients_set_gradient("Full saturation spectrum CCW")
name = pdb.gimp_gradients_get_gradient()
pdb.plug_in_gradmap(newImage, layer1)

str1 = 'now'
str2 = 'then'

for i in range(36):
    str1 = str(i).zfill(2)
    str2 = "lay"+str1
    newlayer = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
    pdb.gimp_layer_set_name(newlayer, str2)
    pdb.gimp_image_add_layer(newImage, newlayer, 0) # adds layer on top
    pdb.gimp_hue_saturation(layer1, 0, 10, 0, 0)

#Now play as a movie
pdb.plug_in_animationplay(newImage, layer1)
#stop movie to continue

#save as gif movie
filename2 = "/Users/don_sauer/Downloads/Bob_Dylan.gif"
pdb.gimp_convert_indexed(newImage, 0, 0, 255, 0, 0, "palette")
pdb.file_gif_save(newImage, layer1, filename2, "test", 0, 1, 100, 0)

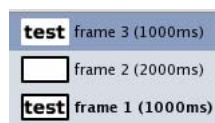
```

## =====CREATE\_PLAY\_AND\_SAVE\_GIF\_MOVIE=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```





```

w = 200
h = 100
newImage = pdb.gimp_image_new(w, h, 0)
layer1 = pdb.gimp_layer_new(newImage, w, h, 0, 'layer1', 100, 0) # RGB0, GRAY1, INDEXED2
display = pdb.gimp_display_new(newImage) # <gimp.Layer 'layer1'>
pdb.gimp_image_add_layer(newImage, layer1, 0) # opens window
pdb.gimp_edit_fill(layer1, 2) # layer added
pdb.gimp_context_set_default_colors() # FG0, BG1, W2, TRAN3, PAT4, NONE5

Txlayer = pdb.gimp_text_layer_new(newImage, "test", "Sans Bold", 24, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0) # Txlayer at position 1
tw = pdb.gimp_drawable_width(Txlayer) # 243
th = pdb.gimp_drawable_height(Txlayer) # 83
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2) # center text
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)

layer1 = pdb.gimp_image_merge_down(newImage, Txlayer, 0)
pdb.gimp_layer_set_name(layer1, "frame 1 (1000ms)")

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, "frame 2 (2000ms)")
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top

layer3 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer3, "frame 3 (1000ms)")
pdb.gimp_image_add_layer(newImage, layer3, 0) # adds layer on top

pdb.gimp_selection_all(newImage)
pdb.gimp_edit_bucket_fill(layer2, 1, 0, 100, 0, 0, 0, 0) # (FG, BG, PAT) (NOR, DIS, BHIN, MUL...) op, thres, merg, x, y

pdb.gimp_image_crop(newImage, w, h, 0, 0)

#Now play as a movie
pdb.plugin_animationplay(newImage, layer1)
#stop movie to continue to Save file

#save as gif movie
pdb.gimp_convert_indexed(newImage, 0, 0, 255, 0, 0, "palette")
fpath = "/Users/don_sauer/Downloads/test.gif"
loop = 1
pdb.file_gif_save(newImage, layer1, fpath, "test", 0, loop, 100, 0)

```

=====LOAD\_ANIMATE\_GIFF=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Desktop/tweety_e0.gif"
rawFilename = "tweety"
newImage = pdb.file_gif_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)

pdb.plugin_animationplay(newImage, layer1)

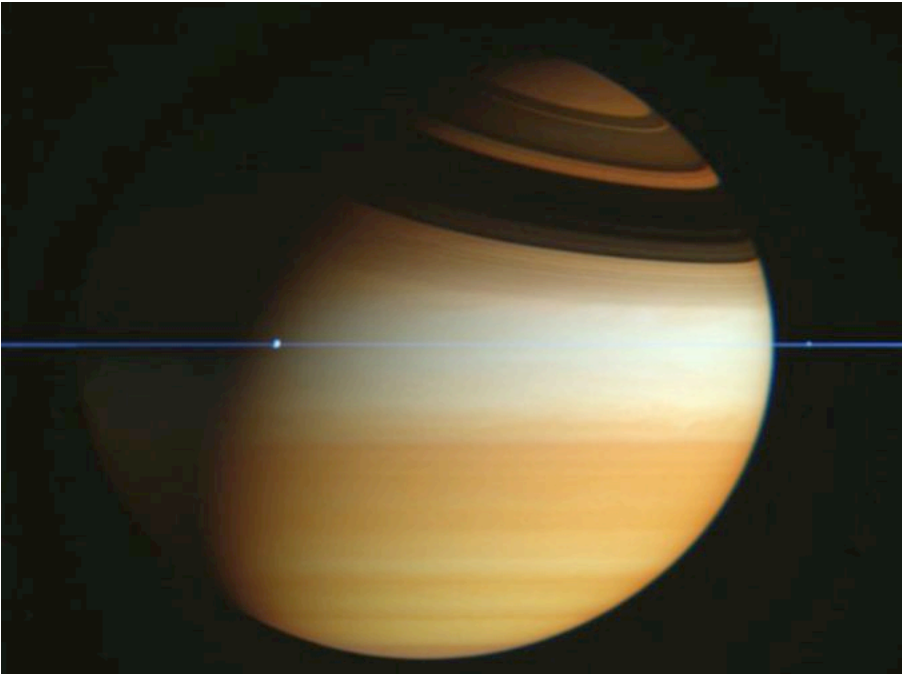
```

=====ResizeImage=====

```

pdb.gimp_display_delete(display) # Optional delete previous image

```



```

filename = "/Users/don_sauer/Downloads/GimpPython/saturnplane_cassini_3_2_300b.jpg"
rawFilename = "Test.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
new_layer = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)

drawable = pdb.gimp_image_get_active_drawable(newImage)          # <gimp.Layer 'Background'>
activeLayer = pdb.gimp_image_get_active_layer(newImage)          # <gimp.Layer 'Background'>
this_filename = pdb.gimp_image_get_filename(newImage)            # '/Users/don_sauer/Desktop/Tiger.jpg'
height = pdb.gimp_image_height(newImage)                          # 665
width = pdb.gimp_image_width(newImage)                            # 900
layer1 = pdb.gimp_image_get_layer_by_name(newImage, 'Background') # <gimp.Layer 'Background'>
xresolution, yresolution = pdb.gimp_image_get_resolution(newImage) # (300.0, 300.0)

pdb.gimp_image_set_resolution(newImage, 72, 72)

aspectR = float(height)/float(width )
aspectR2 = 1/aspectR

if aspectR2 > 1:
    new_height = int( 72*7)
    new_width = int(72*7*aspectR2)

if aspectR > 1:
    new_width = int( 72*7)
    new_height = int(72*7*aspectR)

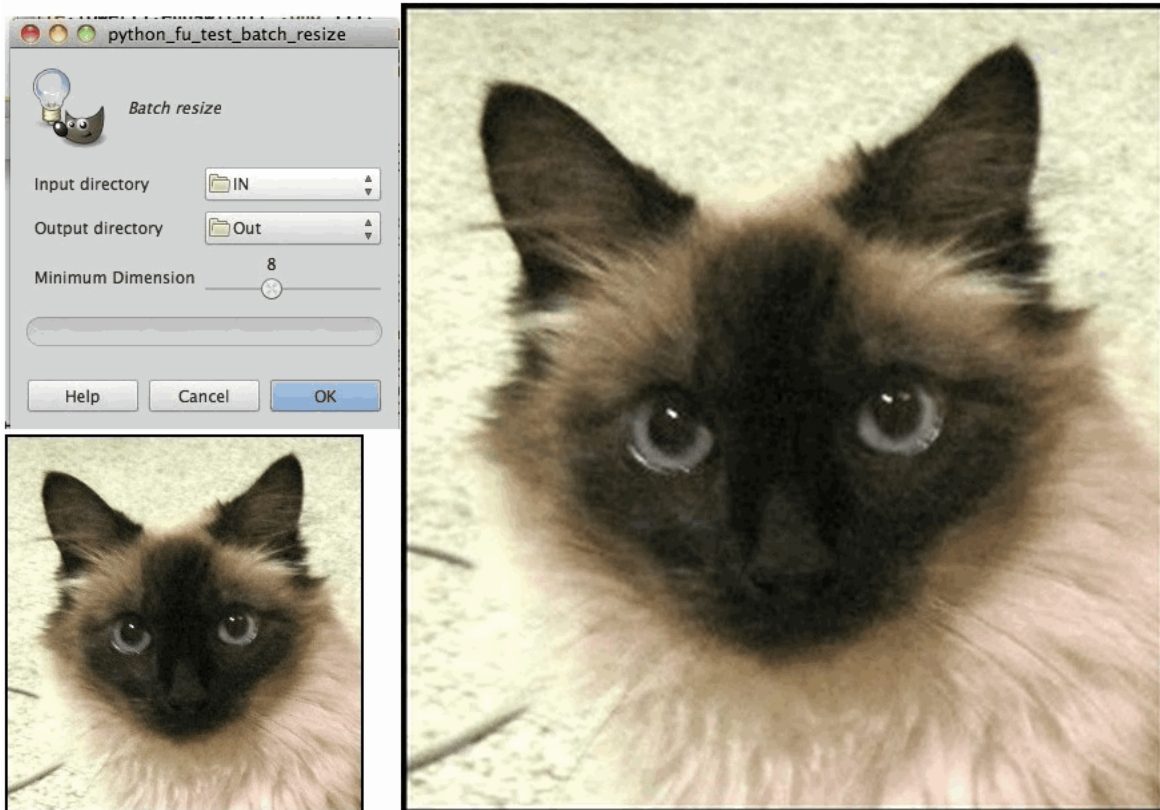
interpolation = 3                                                  # NONE0,LINEAR1,CUBIC2,LANCZO3)
pdb.gimp_image_scale_full(newImage, new_width, new_height, interpolation)

New_filename = "/Users/don_sauer/Downloads/test.jpeg"
raw_filename = "test"
pdb.file_jpeg_save(newImage, layer1, New_filename, raw_filename, 0.8, 0.8, 0, 0, "none", 0, 0, 0, 0)

```

=====test-batch-resize.py=====

Filters/Test/Batch resize



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

```
import os
from gimpfu import *
```

```
def batch_resize(img, layer, inputFolder, outputFolder, minDimension):
    for file in os.listdir(inputFolder):
        # Iterate the folder
        try:
            # Build the full file paths.
            inputPath = inputFolder + "/" + file
            outputPath = outputFolder + "/" + file
            # Open the file if is a JPEG or PNG image.
            if(file.lower().endswith(('.png'))):
                image = pdb.file_png_load(inputPath, inputPath)
            if(file.lower().endswith(('.jpeg', '.jpg'))):
                image = pdb.file_jpeg_load(inputPath, inputPath)
            if(image != None):
                # Verify if the file is an image.
                if(len(image.layers) > 0):
                    layer = image.layers[0]
                    pdb.gimp_image_set_resolution(image, 72, 72)
                    # set resolution of image.
                    aspectR = float(image.height)/float(image.width)
                    aspectR2 = 1/aspectR
                    if aspectR2 > 1:
                        new_height = int( 72*minDimension)
                        new_width = int(72*minDimension*aspectR2)
                    if aspectR > 1:
                        new_width = int( 72*minDimension)
                        new_height = int(72*minDimension*aspectR)
                    interpolation = 3
                    pdb.gimp_image_scale_full(image, new_width, new_height, interpolation) # resize
                    if(file.lower().endswith(('.png'))):
                        # Save the image.
                        pdb.file_png_save(image, image.layers[0], outputPath, outputPath, 0,9,0,0,0,0,0)
                    if(file.lower().endswith(('.jpeg', '.jpg'))):
                        pdb.file_jpeg_save(image, layer, outputPath, outputPath, 0,9,0,0,0,"byGIMP",0,0,0,0)
        except Exception as err:
            gimp.message("Unexpected error: " + str(err))
```

```
register(
    "python_fu_test_batch_resize",
    "Batch resize",
    "Apply resize to the PNG and JPEG images of a folder.",
    "JFM",
    "Open source (BSD 3-clause license)",
    "2013",
    "<Image>/Filters/Test/Batch resize",
    "",
    [
        (PF_DIRNAME, "inputFolder", "Input directory", ""),
        (PF_DIRNAME, "outputFolder", "Output directory", ""),
        (PF_SLIDER, "minDimension", "Minimum Dimension", 7, (1,20,1)),
    ],
    [],
    batch_resize)
```

```
main()
```

## =====FORMAT\_TEMPLATES=====

```
pdb.gimp_display_delete(display) # optional for reload

=====NEW_IMAGE=====
w = 300
h = 300
newImage = pdb.gimp_image_new(w, h, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,w,h,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_edit_fill(layer1, 2) # FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_context_set_default_colors()

=====
newImage = pdb.gimp_image_new(400, 400, 0) # RGB0,GRAY1,INDEXED2
layer1 = pdb.gimp_layer_new(newImage,400,400,0,'layer1',100,0) # <gimp.Layer 'layer1'>
display = pdb.gimp_display_new(newImage) # opens window
pdb.gimp_image_add_layer(newImage, layer1, 0) # layer added
pdb.gimp_layer_add_alpha(layer1)
has_alpha = pdb.gimp_drawable_has_alpha(layer1)
pdb.gimp_edit_fill(layer1, 3) # ??? FG0,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(layer1)
pdb.gimp_selection_none(newImage)

=====LOAD_IMAGE=====

filename = "/Users/don_sauer/Desktop/Tiger.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
layer1 = pdb.gimp_file_load_layer(newImage, filename)
display = pdb.gimp_display_new(newImage)
layer1 = pdb.gimp_image_get_active_layer(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")

filename = "/Users/don_sauer/Desktop/Bob_Dylan.jpg"
rawFilename = "Tiger.jpg"
newImage = pdb.gimp_file_load(filename, rawFilename)
backgnd = pdb.gimp_file_load_layer(newImage, filename)
layer1 = pdb.gimp_image_get_active_layer(newImage)
display = pdb.gimp_display_new(newImage)
pdb.gimp_layer_set_name(layer1, "layer1")

width = pdb.gimp_image_width(newImage)
height = pdb.gimp_image_height(newImage)

pdb.gimp_convert_rgb(newImage)
pdb.gimp_convert_grayscale(newImage)
pdb.gimp_image_convert_grayscale(newImage)

dithtype = 1 # { NO-DITHER (0), FS-DITHER (1), FSLOWBLEED-DITHER (2), FIXED-DITHER (3) }
palettetype = 3 # MAKE-PALETTE (0), WEB-PALETTE (2), MONO-PALETTE (3), CUSTOM-PALETTE (4) }
numcols = 1
alphadither = 0
remo_unused = 0
palette = " "
pdb.gimp_convert_indexed(newImage, dithtype, palettetype,numcols,alphadither, remo_unused, palette)

=====COPY_LAYER=====

layer2 = pdb.gimp_layer_copy(layer1, 1) # layer2 copy of layer + alpha
pdb.gimp_layer_set_name(layer2, 'layer2')
pdb.gimp_image_add_layer(newImage, layer2, 0) # adds layer on top

=====SET_COLORS=====

pdb.gimp_context_set_default_colors()
pdb.gimp_context_set_foreground((255,255,255))
pdb.gimp_context_set_background((0,0,255))

=====SELECT=====

pdb.gimp_by_color_select(layer1, (0,0,0), 60, 0, 0, 0, 3, 0) # 60 threshold radius 3
pdb.gimp_selection_all(newImage)
pdb.gimp_edit_clear(layer1)
pdb.gimp_selection_none(newImage)

pdb.gimp_rect_select(newImage, 10, 10, 50, 50, 0, 0, 0) # "x y w h mode"); ADD0,SUB1,REP2,INTE3
pdb.gimp_ellipse_select(newImage, 100, 70, 50, 50, 0, 0, 0) # x,y,w,h,add?,alias,feath, rad
pts2 = [170,30,200,30,200,50,170,30]
pdb.gimp_image_select_polygon(newImage, 0,len(pts2), pts2)

pdb.gimp_selection_grow(newImage, 4)
pdb.gimp_edit_clear(drawable)
pdb.gimp_selection_invert(newImage)

=====SELECT_FILL=====
# NOR0,DIS1,BHIN2,MUL3),SCRN4,OVERL5,DIFF6,ADD7,SUB8,Dk9,LI10,HUE11,SAT1,COL13,VAL14,DIV15,
```



```

# DGE16,BURN17,HAR18,SOFT19,EXTR20,MERGE21, COLERASE22,ERASE23,REPLACE24,ANTIERASE25
pdb.gimp_edit_bucket_fill(layer2,0,0,100,0,0,0,0) # (FG,BG,PAT)(NOR,DIS,BHIN,MUL..)op,thres,merg, x, y

pdb.script_fu_distress_selection(newImage, layer2, 127, 8, 4, 1, 1, 1) # thres,sprd,gran,sh,sv

=====SELECT_BLEND=====

blend = 0 # FG-BG-RGB0,FG-BG-HSV1,FG-TRANS2,CUSTOM3
paint = 0 # NOR0,DIS1,BHIN2,MUL3,SCRN4,OVER5,DIFF6,ADD7,SUBTRACT-MOD
grad = 0 # LIN0,BILIN1,RAD),SQR3,CONSYM),CONASYM5,BURSTANG6,BURSTSPHER7,BURSTDIMP8,CLOCK9, CCLOCK10
opac = 100
off = 0
repeat = 0
rev = 0
ssamp = 0
dep = 0
thres = 0
dith = 0
x1 = 0
y1 = 0
x2 = 300
y2 = 300
pdb.gimp_edit_blend(layer1,blend,paint,grad,opac,off,repeat,rev,ssamp,dep,thres,dith,x1,y1,x2,y2)
pdb.gimp_edit_blend(layer1,0,0,0,100,0,0,0,0,1,0,1,20,20,150,150) # LIN0,BILIN1,RAD),SQR3,CONSYM)
pdb.gimp_edit_blend(layer1,2,0,0,100,0,0,0,0,1,0,1,20,20,150,150) # LIN0,BILIN1,RAD),SQR3,CONSYM)

=====SELECT_COPY=====

non_empty = pdb.gimp_edit_copy(drawable)
floating_sel = pdb.gimp_edit_paste(drawable, paste_into)
pdb.gimp_floating_sel_anchor(floating_sel)

=====LAYER=====

pdb.gimp_item_set_visible(layer2, 0)
pdb.gimp_layer_add_alpha(layer1)
pdb.gimp_layer_set_mode(layer2, 6)
pdb.gimp_edit_fill(layer3, 0) # FGO,BG1,W2,TRAN3,PAT4,NONE5
pdb.gimp_image_lower_layer(newImage, layer1)
pdb.gimp_layer_set_mode(layer2, 10) # NORM0,DISS1,BEHIND2,MULT3,SCRN4,OVERL5,DIFF6,ADD7,SUB8,DRK9,LI10
pdb.gimp_layer_set_name(item, name)
pdb.gimp_image_set_active_layer(image,layer1)
pdb.gimp_layer_set_opacity(layer, opacity)

layer1 = pdb.gimp_image_merge_down(newImage, layer2, 0)
layer1 = pdb.gimp_image_merge_visible_layers(newImage,0)

=====FILTER_LAYER=====

pdb.gimp_hue_saturation(layer1, 0, 10, 0, 0)
pdb.gimp_levels(layer2, 0, 89, 160, 1, 0, 255) # RGB= 0 lowin hiin gamma loout hout
pdb.gimp_desaturate(layer1)
pdb.gimp_context_set_gradient("FG to BG (RGB)")
pdb.gimp_threshold(layer2, 130, 255) # "minlev maxlev" 130 255 ))
pdb.gimp_invert(layer1)
pdb.gimp_colorize(layer1, 308, 85, 19) # hue sat lit 308 85 19
pdb.gimp_threshold(layer2, 130, 255) # "minlev maxlev" 130 255 ))
pdb.gimp_color_balance(layer1,1, 1, 0,0,100) #lev (0->2 Shad=>lit) cya2red mag2grn yel2blu"
pdb.gimp_brightness_contrast(drawable, brightness, contrast)

=====PLUG_IN=====

pdb.plug_in_blur(image, drawable)
pdb.plug_in_plasma(newImage, layer1, 566, 7)
pdb.plug_in_randomize_slur(newImage, layer1, 1,17, 10, 11)
pdb.plug_in_oilify(newImage, layer1, 9, 1)
pdb.plug_in_solid_noise(newImage, layer2,1,0,233,15,8,8) # tile turb detail radx rady
pdb.plug_in_mblur(newImage, layer2, 0, 111, 0, 0, 0) # "typ len ang centx centy");lin0,rad1
pdb.plug_in_gradmap(newImage, layer1)
pdb.plug_in_rgb_noise(newImage, layer1, 0, 1, 0.5, 0.5, 0.5, 0)
pdb.plug_in_spread(newImage, layer2, 200, 200)
pdb.plug_in_gauss_iir(newImage, layer2, 12, 12, 12) #rad blurx blurry"
pdb.plug_in_spread(newImage, layer2, 200, 200)
pdb.plug_in_gauss_iir(newImage, layer2, 12, 12, 12) #rad blurx blurry"
pdb.plug_in_gauss(newImage, layer2, 12, 12, 0) #
pdb.plug_in_solid_noise(newImage, layer1,1,0,233,15,8,8) # tile turb detail radx rady
pdb.plug_in_gauss_iir(newImage, layer1, 12, 12, 12) #rad blurx blurry"
pdb.plug_in_emboss(newImage, layer1, 90, 50, 5, 1) # "azi elev depth embos" 90 40 5 1 )
pdb.gimp_image_scale_full(newImage,800,130,1) # w,h,inter
pdb.plug_in_mblur(newImage, layer1, 0, 35, 0, 0, 0) # "typ len ang centx centy");lin0,rad1
pdb.plug_in_grid(newImage, layer1, 1, 5, 8, (0,0,0), 255, 1, 5, 8,(0,0,0), 255, 0, 2, 6, (0,0,0), 255)
pdb.plug_in_whirl_pinch(newImage, layer1, 90, 0, 1)
seed = random.randrange(65535)
pdb.plug_in_plasma(newImage, layer2, seed, .2) # "seed turb" (rand 65535) .2 ) )
pdb.plug_in_hsv_noise(newImage, layer2, 4, 81, 62, 62) # "hold hue sat val"
pdb.plug_in_colortoalpha(newImage, activeLayer, (0,0,0))
pdb.plug_in_threshold_alpha(newImage, activeLayer, 100)
pdb.plug_in_unsharp_mask(image, drawable, radius, amount, threshold)

bmap = "bumplayer"
azim = 74
elev = 36
depth = 5
x = 0
y = 0
level = 0
amb = 0
comp = 0 # compenstate brightness

```

```

inv = 0
type = 0
pdb.plugin_bump_map(image, drawable, bmap, azim, elev, depth, x, y, level, amb, comp, inv, type)
pdb.plugin_bump_map(newImage, Background, wlayer, 74, 36, 5, 0, 0, 0, 70, 1, 0, 0)
pdb.plugin_bump_map(newImage, Txlayer4, Txlayer3, 47, 40, 30, 0, 0, 0, 100, 1, 0, 0)

=====MASK=====

mask3 = pdb.gimp_layer_create_mask(layer3, 0) # W0,B1,A2,ATRANS3,SEL4,COPY5,CHAN6
pdb.gimp_layer_add_mask(layer3, mask3) #

pdb.gimp_selection_all(newImage)
non_empty = pdb.gimp_edit_copy(layer2)
floating_sel = pdb.gimp_edit_paste(mask3, 0)
pdb.gimp_floating_sel_anchor(floating_sel)

=====TRANSFORM=====

pdb.gimp_layer_translate(Txlayer, 2, 2)

flip = 0 # HORIZONTAL (0), VERTICAL (1) }
autocenter = 1 #
axis = 500 #
clip = 0 #
layer2 = pdb.gimp_drawable_transform_flip_simple(layer2, flip, autocenter, axis, clip)

offset = 50
x0 = 100 # up left
y0 = 100 + offset
x1 = 500 # up right
y1 = 100
x2 = 100 # dwn left
y2 = 500 - offset
x3 = 500 # dwn right
y3 = 500
dir = 0 # TRANSFORM-FORWARD (0), TRANSFORM-BACKWARD (1) }
inter = 3 # NONE (0), LINEAR (1), ICUBIC (2), LANCZOS (3) }
ssamp = 0
lev = 1
clip = 0 # RESIZE0,CLIP1,CROP2,CROP-WITH-ASPECT3
layer2 = pdb.gimp_drawable_transform_perspective(layer2,x0,y0,x1,y1,x2,y2,x3,y3,dir,inter,ssamp,lev,clip)
layer1 = pdb.gimp_item_transform_perspective(layer1, 300,0,400,0,0,700,700,700)

=====BRUSH=====

pdb.gimp_context_set_brush(NewBrush)

NewBrush = pdb.gimp_brush_new("nextbrush")
radius_out = pdb.gimp_brush_set_radius(NewBrush,3)
hardness_out = pdb.gimp_brush_set_hardness(NewBrush, 1.0)
pdb.gimp_edit_stroke(layer1)
pdb.gimp_brush_delete(NewBrush)
name, width, height, spacing = pdb.gimp_brushes_get_brush()

NewBrush = pdb.gimp_brush_new("nextbrush")
radius_out = pdb.gimp_brush_set_radius(NewBrush,10)
hardness_out = pdb.gimp_brush_set_hardness(NewBrush, 1.0)
pdb.gimp_brushes_set_spacing(1)
pdb.gimp_context_set_brush(NewBrush)

pdb.gimp_context_set_paint_method('gimp-smudge')
pdb.gimp_context_get_paint_method() #'gimp-smudge'
points = [80,30, 140,30]
pdb.gimp_smudge_default(activeLayer,len(points), points)

pdb.gimp_context_set_paint_method('gimp-eraser')
points = [80,60, 140,60]
pdb.gimp_eraser_default(activeLayer,len(points), points)

pdb.gimp_context_set_paint_method('gimp-clone')
points = [80,100, 140,100]
pdb.gimp_clone(activeLayer, activeLayer, 0, 80, 80, len(points), points)

pdb.gimp_context_set_paint_method('gimp-heal')
points = [100,125, 150,125]
pdb.gimp_heal(activeLayer, activeLayer, 55, 160, len(points), points)

pdb.gimp_context_set_paint_method('gimp-dodge-burn')
points = [80,150, 130,150]
exposure= 100
dodgeburn_type= 0 # DODGE (0), BURN (1)
dodgeburn_mode= 1 # SHADOWS (0), MIDTONES (1), HIGHLIGHTS
pdb.gimp_dodgeburn(activeLayer, exposure, dodgeburn_type, dodgeburn_mode, len(points), points)

pdb.gimp_context_set_paint_method('gimp-dodge-burn')
points = [80,180, 130,180]
dodgeburn_type= 1 # DODGE (0), BURN (1)
dodgeburn_mode= 2 # SHADOWS (0), MIDTONES (1), HIGHLIGHTS
pdb.gimp_dodgeburn(activeLayer, exposure, dodgeburn_type, dodgeburn_mode, len(points), points)

pdb.gimp_context_set_paint_method('gimp-airbrush')
pressure = 50 # 100 mx
points = [80,210, 130,210]
pdb.gimp_airbrush(activeLayer, pressure, len(points), points)

pdb.gimp_context_set_paint_method('gimp-paintbrush')
points = [80,250, 130,250]

```

```

pdb.gimp_paintbrush_default(activeLayer, len(points), points)

pdb.gimp_context_set_paint_method('gimp-convolve')
points = [80,280, 130,280]
type = 0 # BLUR-CONVOLVE (0), SHARPEN-CON
pdb.gimp_convolve(drawable, 100,type, len(points), points)

pdb.gimp_brush_delete(NewBrush)

num_paint_methods, paint_methods = pdb.gimp_context_list_paint_methods()
#(11, ('gimp-pencil', 'gimp-paintbrush', 'gimp-eraser', 'gimp-airbrush',
# 'gimp-ink', 'gimp-clone', 'gimp-heal', 'gimp-perspective-clone',
# 'gimp-convolve', 'gimp-smudge', 'gimp-dodge-burn'))

=====TEXT=====
Txlayer = pdb.gimp_text_layer_new(newImage, "BEVEL", "Gill Sans Bold", 50, 2)
pdb.gimp_layer_set_name(Txlayer, "Txlayer")
pdb.gimp_image_add_layer(newImage, Txlayer, 0) # Txlayer at position 1
tw = pdb.gimp_drawable_width(Txlayer) # 243
th = pdb.gimp_drawable_height(Txlayer) # 83
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2) # center text
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)

Txlayer = pdb.gimp_text_layer_new(newImage, "Effects", "Gill Sans Bold", 25, 2)
pdb.gimp_image_add_layer(newImage, Txlayer, 1) # Txlayer at position 1
pdb.gimp_image_lower_layer(newImage, layer1) # how to lower layer1
tw = pdb.gimp_drawable_width(Txlayer) # 243
th = pdb.gimp_drawable_height(Txlayer) # 83
pdb.gimp_layer_translate(Txlayer, (w-tw)/2, (h-th)/2) # center text
offx, offy = pdb.gimp_drawable_offsets(Txlayer) # (20, 100)

pdb.gimp_by_color_select(Txlayer,(0,0,0), 10, 0, 0, 0, 1, 0) # select black

=====VECTORS=====
activeVectors = pdb.gimp_image_get_active_vectors(newImage) # <gimp.Vectors 'Unnamed'>
name = pdb.gimp_vectors_get_name(activeVectors) # 'Unnamed'
pdb.gimp_vectors_set_name(activeVectors, "firstPath")
num_strokes, stroke_ids = pdb.gimp_vectors_get_strokes(activeVectors) # (3, (1, 2, 3)) THREE strokes 1 TO 3

type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[0])
#(0,18,(18.0,91.0,30.0,75.0,42.0,59.0,80.0,13.0,97.0,45.0,114.0,77.0,151.0,48.0,151.0,48.0,151.0,48.0),0)
length = pdb.gimp_vectors_stroke_get_length(activeVectors, stroke_ids[0], 3) # 151.6713715341169
pdb.gimp_vectors_stroke_flip(activeVectors, stroke_ids[0], 0, 150) # flip_HO,V1, axis
pdb.gimp_vectors_stroke_translate(activeVectors, stroke_ids[2], 30, 30)
pdb.gimp_vectors_stroke_rotate(activeVectors, stroke_ids[1], 130, 130, 45.0)
pdb.gimp_vectors_stroke_scale(activeVectors, stroke_ids[2], 1.0, 0.5)
x,y,slope,valid = pdb.gimp_vectors_stroke_get_point_at_dist(activeVectors, stroke_ids[0],50,3) #(235.01,39.61,0.721,1)
fpath = "/Users/don_sauer/Downloads/firstPath"
pdb.gimp_vectors_export_to_file(newImage, fpath, activeVectors)
vstring = pdb.gimp_vectors_export_to_string(newImage, activeVectors) # svg file format

pdb.gimp_image_remove_vectors(newImage, activeVectors)
merge = 0
scale = 0
num_vectors, vectors_ids = pdb.gimp_vectors_import_from_file(newImage, fpath, merge, scale)
activeVectors = pdb.gimp_image_get_active_vectors(newImage) # <gimp.Vectors 'firstPath'>
pdb.gimp_vectors_set_visible(activeVectors, 1)
fpath = "/Users/don_sauer/Downloads/firstPath"
pdb.gimp_vectors_export_to_file(newImage, fpath, activeVectors)
vstring = pdb.gimp_vectors_export_to_string(newImage, activeVectors) # svg file format

pdb.gimp_image_remove_vectors(newImage, activeVectors)
merge = 0
scale = 0
num_vectors, vectors_ids = pdb.gimp_vectors_import_from_file(newImage, fpath, merge, scale)

activeVectors = pdb.gimp_image_get_active_vectors(newImage) # <gimp.Vectors 'firstPath'>
pdb.gimp_vectors_set_visible(activeVectors, 1)

pdb.gimp_image_select_rectangle(newImage, 0, 50,200, 70, 70)
pdb.gimp_image_select_ellipse(newImage, 0, 160, 200, 80, 80)
pdb.plugin_sel2path(newImage, layer1)
activeVectors = pdb.gimp_image_get_active_vectors(newImage) # <gimp.Vectors 'Selection'>
pdb.gimp_vectors_set_visible(activeVectors, 1)

num_strokes, stroke_ids = pdb.gimp_vectors_get_strokes(activeVectors) # (2, (1, 2)) Two strokes number 1 TO 2

type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[0])
#(0,24,(120,200,120,200,120,200,120,270,120,270,120,270,50,270,50,270,50,270,50,200,50,200,50,200),1)

type, num_points, controlpoints, closed = pdb.gimp_vectors_stroke_get_points(activeVectors, stroke_ids[1])
#(0,30,(206,200,212,202,216,203,222,206,226,209,252,231,240,274,206,280,166,286,144,235,174,209,180,204,187,
# 202,194,200,200,200),1)

pdb.gimp_selection_none(newImage)
pdb.gimp_vectors_to_selection(activeVectors, 0, 0, 1, 0, 0)
pdb.gimp_vectors_stroke_translate(activeVectors, stroke_ids[0], -10, -10)
pdb.gimp_vectors_stroke_translate(activeVectors, stroke_ids[1], -20, 0)

pdb.plugin_sel2path(newImage, layer1)
active_vectors = pdb.gimp_image_get_active_vectors(newImage)
pdb.gimp_vectors_set_visible(active_vectors, 1)

=====RANDOM=====
import random
from gimpfu import *
import sys,random as r
import math, string

```

```

from gimpfu import *
import sys, random as r
import math, string

from array import array
import os
from gimpfu import *
import os, sys

randint(2,9) #Inclusive

print random.random()
random.randrange(2, 5)
random.uniform(0, 10)
random.normalvariate(1, 10)    # ave, sd

len(points)

=====FOR_LOOP=====

for i in range(30):
    x1 = random.randrange(0,1000) #
    y1 = random.randrange(0,1000) #
    x2 = random.randrange(0,1000) #
    y2 = random.randrange(0,1000) #
    pdb.gimp_edit_blend(layer1,blend,paint,grad,opac,off,repert,rev,ssamp,dep,thres,dith, x1, y1, x2, y2)

=====MATH=====
import math, string

math.ceil(x)      # Return ceiling as float, smallest integer value greater than or equal to x.
math.copysign(x, y) # Return x with the sign of y.
math.fabs(x)      # Return the absolute value of x.
math.factorial(x) # Return x factorial. Raises ValueError if x is not integral or is negative.
math.floor(x)     # Return floor of x as a float, the largest integer value less than or equal to x.
math.fmod(x, y)   # Return fmod(x, y), as defined by the platform C library.
math.frexp(x)     # Return mantissa and exponent of x as pair (m, e) x == m * 2**e exactly. I
math.fsum(iterable) # Return an accurate floating point sum of values in the iterable. ccurate floating point.
math.isinf(x)     # Check if the float x is positive or negative infinity.
math.isnan(x)     # Check if the float x is a NaN (not a number).
math.ldexp(x, i)  # Return x * (2**i). This is essentially the inverse of function frexp().
math.modf(x)      # Return fractional and integer parts of x. results carry sign of x and are floats.
math.trunc(x)     # Return the Real value x truncated to an Integral (usually a long integer).
math.exp(x)       # Return e**x.
math.expml(x)     # Return e**x - 1.
math.log(x[, base]) # With one argument, return the natural logarithm of x (to base e).
math.loglp(x)     # Return the natural logarithm of 1+x (base e).
math.log10(x)     # Return the base-10 logarithm of x. This is usually more accurate than log(x, 10).
math.pow(x, y)    # Return x raised to power y.
math.sqrt(x)      # Return the square root of x.
math.acos(x)      # Return the arc cosine of x, in radians.
math.asin(x)      # Return the arc sine of x, in radians.
math.atan(x)      # Return the arc tangent of x, in radians.
math.atan2(y, x)  # Return atan(y / x), in radians.
math.cos(x)       # Return the cosine of x radians.
math.hypot(x, y)  # Return Euclidean norm, sqrt(x*x + y*y). length of vector from origin to point (x, y).
math.sin(x)       # Return the sine of x radians.
math.tan(x)       # Return the tangent of x radians.
math.degrees(x)   # Converts angle x from radians to degrees.
math.radians(x)   # Converts angle x from degrees to radians.
math.acosh(x)     # Return the inverse hyperbolic cosine of x.
math.asinh(x)     # Return the inverse hyperbolic sine of x.
math.atanh(x)     # Return the inverse hyperbolic tangent of x.
math.cosh(x)      # Return the hyperbolic cosine of x.
math.sinh(x)      # Return the hyperbolic sine of x.
math.tanh(x)      # Return the hyperbolic tangent of x.
math.erf(x)       # Return the error function at x.
math.erfc(x)      # Return the complementary error function at x.
math.gamma(x)     # Return the Gamma function at x.
math.lgamma(x)    # Return the natural logarithm of the absolute value of the Gamma function at x.

math.pi          # 3.141592..., to available precision.
math.e            # 2.718281..., to available precision.

=====GRADIENTS=====
pdb.gimp_gradients_set_gradient("Full saturation spectrum CCW")
name = pdb.gimp_gradients_get_gradient()
pdb.plugin_gradmap(newImage, layer1)

thermalgrad = pdb.gimp_gradient_new("thermalgrad") # 'thermalgrad'
pdb.gimp_gradient_segment_range_split_uniform(thermalgrad, 0, 1, 2) # start , end, numb parts
pdb.gimp_gradient_segment_set_left_color(thermalgrad, 0, (0,0,255), 100)
pdb.gimp_gradient_segment_set_right_color(thermalgrad, 0, (255,0,0), 100)
pdb.gimp_gradient_segment_set_left_color(thermalgrad, 1, (255,0,0), 100)
pdb.gimp_gradient_segment_set_right_color(thermalgrad, 1, (255,255,0), 100)

pdb.gimp_context_set_gradient(thermalgrad)

grad2 = pdb.gimp_gradient_new("grad2") # 'grad2'
pdb.gimp_gradient_segment_range_split_uniform(grad2, 0, 3, 4)
pdb.gimp_gradient_segment_set_left_color(grad2, 0, (255,0,0), 100)
pdb.gimp_gradient_segment_set_right_color(grad2, 0, (255,255,0), 100)
pdb.gimp_gradient_segment_set_left_color(grad2, 1, (255,255,0), 100)
pdb.gimp_gradient_segment_set_right_color(grad2, 1, (0,255,0), 100)
pdb.gimp_gradient_segment_set_left_color(grad2, 2, (0,255,0), 100)
pdb.gimp_gradient_segment_set_right_color(grad2, 2, (0,255,255), 100)
pdb.gimp_gradient_segment_set_left_color(grad2, 3, (0,255,255), 100)
pdb.gimp_gradient_segment_set_right_color(grad2, 3, (0,0,255), 100)

```



```

pdb.gimp_gradient_delete(grad3)

=====GIF_MOVIE=====
str1 = 'now'
str2 = 'then'

for i in range(36):
    str1 = str(i).zfill(2)
    str2 = "lay"+str1
    newlayer = pdb.gimp_layer_copy(layer1, 1)
    pdb.gimp_layer_set_name(newlayer, str2)
    pdb.gimp_image_add_layer(newImage, newlayer, 0)
    pdb.gimp_hue_saturation(layer1, 0, 10, 0, 0)
    # 01, 02, ....
    # lay01, lay02....
    # newlayer copy of layer1 + alpha
    # add layer on top

#Now play as a movie
pdb.plug_in_animationplay(newImage, layer1)
#stop movie to continue

#save as gif movie
filename2 = "/Users/don_sauer/Downloads/Bob_Dylan.gif"
pdb.gimp_convert_indexed(newImage, 0, 0, 255, 0, 0, "palette")

=====PATTERN=====

name = pdb.gimp_context_get_pattern()
pdb.gimp_context_set_pattern(name)
width, height, bpp, num_color_bytes, color_bytes = pdb.gimp_pattern_get_pixels(name)
num_patterns, pattern_list = pdb.gimp_patterns_get_list(filter)
pdb.script_fu_paste_as_pattern(string, string) # Paste the clipboard contents into a new pattern
pdb.script_fu_selection_to_pattern(image, drawable, string, string)

=====PALETTES_COLORMAPS=====

name = pdb.gimp_context_get_palette()
pdb.gimp_context_set_palette(name)
color = pdb.gimp_palette_entry_get_color(name, entry_num)

=====RUN_script_fu_eval=====

num_images, image_ids = pdb.gimp_image_list()
num_layers, layer_ids = pdb.gimp_image_get_layers(newImage)
# (1, (33,))
# (2, (84, 87))

test = "(plug-in-lighting 1 "+ str(image_ids[0])+" " + str(layer_ids[0])+" " +str(layer_ids[1])+" "
test = test + " 0 1 0 0 0 '(255 255 255) -1.67 -1.5 3 -1 -1 1 .5 .5 .1 .2 3 .2 3 0 )"
test = test + "# '(plug-in-lighting 1 17 45 46 0 1 0 0 0 '(255 255 255) -1.67 -1.5 3 -1 -1 1 .5 .5 .1 .2 3 .2 3 0 )"
pdb.plug_in_script_fu_eval(test)

=====TEXT_FILES=====

filename = "/Users/don_sauer/Desktop/newfile.txt"
file = open(filename, "w")
file.write("hello world in the new file\n")
file.write("and another line\n")
file.close()

filename = "/Users/don_sauer/Desktop/newfile.txt"
f = open(filename)
lines = f.readlines()
f.close()
#['hello world in the new file\n', 'and another line\n']

lines = [line.strip() for line in open(filename)] #['hello world in the new file', 'and another line']

ins = open( "/Users/don_sauer/Desktop/newfile.txt", "r" )
array = []
for line in ins:
    array.append( line )

ins.close()
array
# ['hello world in the new file\n', 'and another line\n']
# 'and another line\n'

array = []
with open(filename, "r") as f:
    for line in f:
        array.append(line)

array
# ['hello world in the new file\n', 'and another line\n']

filename = "/Users/don_sauer/Desktop/newfile.txt"
with open(filename) as f:
    content = f.readlines()
    #If you want the \n included:

content
#['hello world in the new file\n', 'and another line\n']

with open(filename) as f:
    content = f.read().splitlines()
    #If you do not want \n included:

content
# ['hello world in the new file', 'and another line']
content[0]
# 'hello world in the new file'

file = open(filename, 'r')
print file.read()
# hello world in the new file
# and another line
# string containing all characters in file,

file = open("/Users/don_sauer/Desktop/newfile.txt", 'r')
print file.read(5)

```

```

# hello

file = open(filename, 'r')
print file.readline()
# hello world in the new file

file = open(filename, 'r')
print file.readlines()
#['hello world in the new file\n', 'and another line\n']

file = open(filename, 'r')
i = 0
for line in file:
    print "# "+str(i) + ' '+line
    i = i + 1
# 0 hello world in the new file
# 1 and another line

filename = "/Users/don_sauer/Desktop/newfile.txt"
file = open(filename, 'r')
lines_of_text = file.read()

lines_of_text #'hello world in the new file\nand another line\n'
len(lines_of_text) #45
lines_of_text[0] # h

filename = "/Users/don_sauer/Desktop/newfile.txt"
file = open(filename, 'r')
lines_of_text = file.readlines() # ['hello world in the new file\n', 'and another line\n']
len(lines_of_text) #2
lines_of_text[0] # 'hello world in the new file\n'

fh = open("Hello.txt", "a") # To append to file, use:
fh = open("hello.txt", "w") # To write to a file, use:
fh = open("hello.txt", "r") # To read a text file, use:

with open("/Users/don_sauer/Desktop/newfile.txt", 'r') as f:
    data = f.readlines()
    for line in data:
        words = line.split()
        print words
#['hello', 'world', 'in', 'the', 'new', 'file']
#['and', 'another', 'line']

data # ['hello world in the new file\n', 'and another line\n']

```

```

=====
3-14-14-14-42-13
dsauersanjose@aol.com
Don Sauer http://www.idea2ic.com/

```