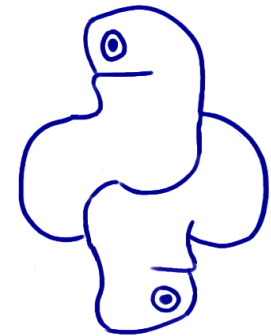


Extending GIMP with Python

Tomaž Šolc
tomaz.solc@tablix.org



Ljubljana Python Meetup
20 September 2018



About me

- In my day job I'm an electrical engineer
 - on occasion I also write software, blog posts and academic papers.
- Using Python professionally since 2007.
- Some evenings I'm learning how to draw
 - As an engineer I'm used to making my own tools so I started hacking GIMP.
 - I thought it would be interesting to present a Python topic here that's not cryptocurrencies and/or web.

Motivation: a quick demo

- 1 minute crash course in GIMP
 - GNU Image Manipulation Program
 - A drawing application (like Photoshop, but FLOSS)
 - *Actively developed since 1995! (original “G” in GTK)*
- Demonstrate some Python plug-ins
 - Import a sketch from “Animation Creator” app,
 - step through frames using onion layers plug-in,
 - export frames into individual PNG files.

Dos and Don'ts

- GIMP plug-ins are good for
 - automating repetitive tasks in interactive sessions,
 - extending GIMP functionality:
new file formats, adding image filters, etc.
- don't use them for
 - batch processing, in web backends, etc. (use PIL)
 - it's not possible to significantly alter the basic GIMP user interface (although tricks are possible and you *can* have your own dialogs, etc.)

Where to start?

- A minimal plug-in is a .py file with two parts:
 - 1) a function that implements plug-in's functionality
 - 2) registration of your function
- To install
 - chmod 755
 - copy to \$HOME/.gimp-2.8/plug-ins
 - restart GIMP
- Extend to taste
 - Multiple functions, .py files, GUI with settings, etc.

A "Hello World" plug-in

```
#!/usr/bin/env python
from gimpfu import *

def greeting(img, layer):
    pdb.gimp_text_fontname(img, None,
                           10, 10, "Hello, World!",
                           -1, False, 50.0, 0, "Sans")

register(
    "python_fu_greeting",
    "Greet the world",
    "Write \"Hello, World!\" on the current layer",
    "Tomaz Solc",
    "Open source (BSD 3-clause license)",
    "2018",
    "<Image>/Filters/Demo/Greeting",
    "*",
    [],
    [],
    greeting)

main()
```

CURRENT IMAGE & LAYER

KEYBOARD SHORTCUT NAME

TOOL TIP

EXTRA FUNC. PARAMETERS

MENU ENTRY

①

②

(GIMP WILL MAKE A NICE GUI FOR YOU!)

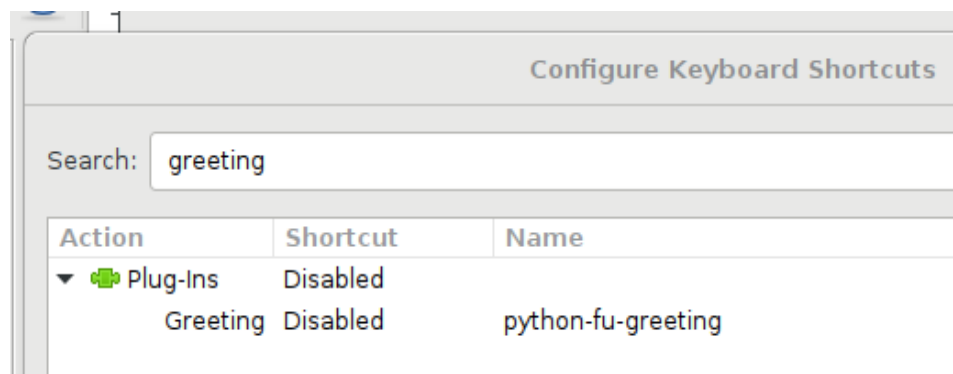
How to invoke your plug-in

(UNDER "FILTERS")

- Menu
 - top menu bar
 - right click menu



- Keyboard shortcuts
 - can also be used with Wacom buttons (but it's not simple)



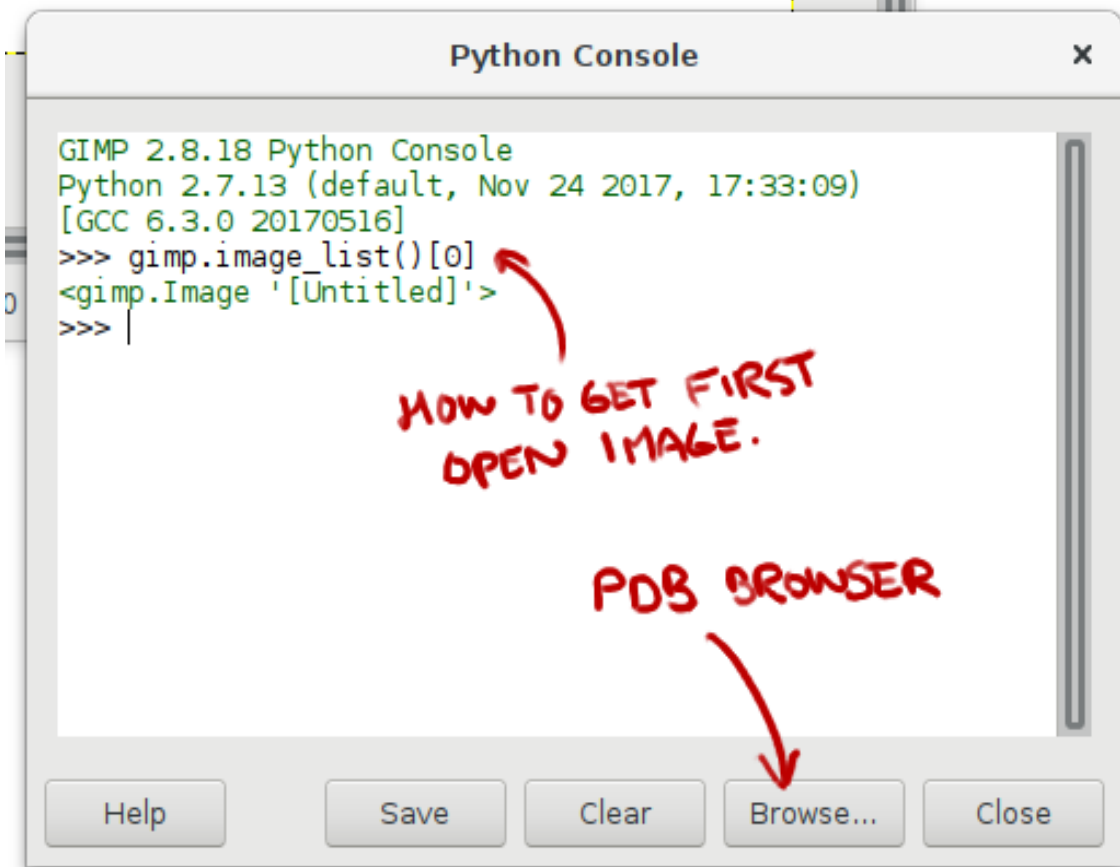
- From another plug-in via PDB

Implementing functionality

- Three ways of interacting with GIMP:
 - “native” Python bindings for GIMP objects
 - Somewhat Pythonic interface (with odd quirks – no GC)
 - Image, Channel, Layer, Parasite, etc. objects
 - Covers basic object buerocracy, but not much else
 - The procedural database (PDB)
 - Something like a RPC interface (think web APIs)
 - Offers complete access to all GIMP UI functionality, including interfacing with other plug-ins.
 - some minor support modules
 - work around some quirks to make things easier.

Figuring things out: REPL

- Menu -> Filters -> Python-fu -> Console
- Just like Python in a terminal
- <tab> for autocomplete
- dir() is your friend.



```
GIMP 2.8.18 Python Console
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516]
>>> gimp.image_list()[0]
<gimp.Image '[Untitled]'\>
>>> |
```

HOW TO GET FIRST OPEN IMAGE.

PDB BROWSER

Help Save Clear Browse... Close

GIMP PDB browser

Python Procedure Browser

Search: by name ▾

- gimp-selection-translate
- gimp-selection-value
- gimp-shear
- gimp-smudge
- gimp-smudge-default
- gimp-temp-PDB-name
- gimp-temp-name
- gimp-text
- gimp-text-fontname**
- gimp-text-get-extents
- gimp-text-get-extents-fontname
- gimp-text-layer-get-antialias
- gimp-text-layer-get-base-direction
- gimp-text-layer-get-color
- gimp-text-layer-get-font
- gimp-text-layer-get-font-size
- gimp-text-layer-get-hint-style

1204 procedures

gimp-text-fontname
Internal GIMP procedure

Add text at the specified location as a floating selection or a new layer.

Parameters

image	IMAGE	The image
drawable	DRAWABLE	The affected drawable: <u>-1</u> for a new text layer
x	FLOAT	The x coordinate for the left of the text bounding box
y	FLOAT	The y coordinate for the top of the text bounding box
text	STRING	The text to generate (in UTF-8 encoding)
border	INT32	The size of the border (border >= -1)
antialias	INT32	Antialiasing (TRUE or FALSE)
size	FLOAT	The size of text in either pixels or points (size >= 0)
size-type	INT32	The units of specified size { <u>PIXELS (0), POINTS (1)</u> }
fontname	STRING	The name of the font

Return Values

text-layer	LAYER	The new text layer or -1 if no layer was created.
------------	-------	---

Additional Information

SOMETIMES INT 105, SOMETIMES OBJECTS

MAGIC NUMBERS

COPIES AN EXAMPLE TO PROMPT

Apply Close

Odd things to watch for

- Python 2.7 :(
- Each function call is in a separate process
 - Global variables don't keep their values
 - Use “gimpshelf” or parasites if you need persistence
 - Be wary of concurrency issues if using keybd. shortcuts.
- Sometimes int can stand for other things
 - -1 vs. None, int vs. Image, int vs. bool
- Run gimp from a terminal
 - Only way to see stacktraces for unhandled exceptions and your print statements.

Getting more info

- GIMP Python documentation
<https://www.gimp.org/docs/python/index.html>
- Plug-ins to ~~steal code~~learn by example from
<https://github.com/topics/gimp-plugin>
- There used to be a “registry.gimp.org” with plug-ins, but it’s off-line and/or broken
- Useful Google keyword: “python fu”.
- GIMP mailing lists, forums, IRC
https://www.gimp.org/mail_lists.html

Plugins I've shown

- **gimp-plugin-hello**

a minimal example Python plug-in

<https://github.com/avian2/gimp-plugin-hello>

- **gimp-plugin-ac-import**

import a project from “Animation Creator HD” iOS app.

<https://github.com/avian2/gimp-plugin-ac-import>

- **gimp-plugin-onion-layers**

move to next/previous frame + some other useful animation functions.

<https://github.com/avian2/gimp-plugin-onion-layers>

- **gimp-plugin-export-layers**

export layers as separate images

<https://github.com/khalim19/gimp-plugin-export-layers>

Questions



Tomaz Šolc
tomaz.solc@tablix.org

(I've also written a few posts on GIMP plug-ins on my blog)
<https://www.tablix.org/~avian/blog/>