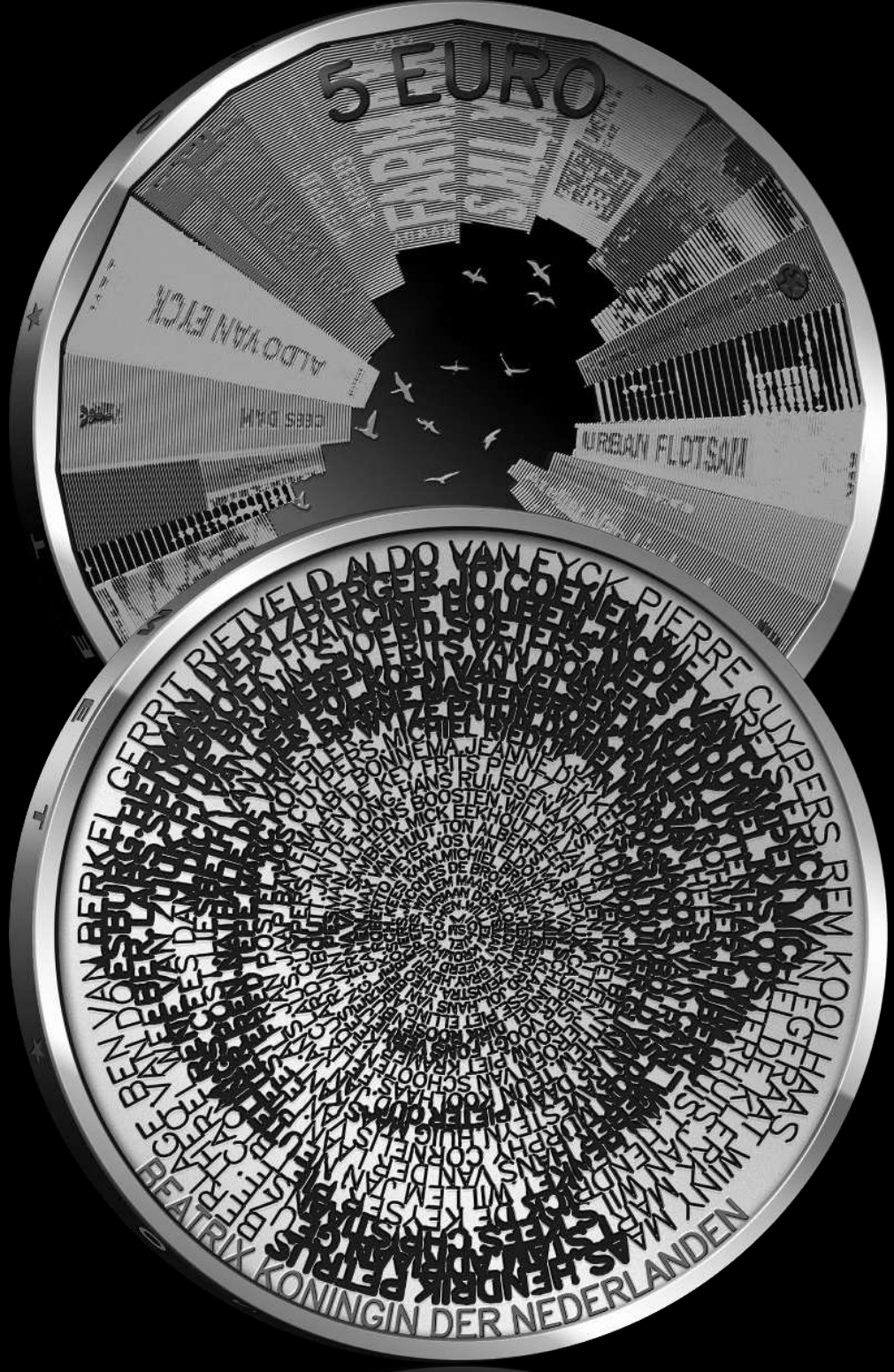
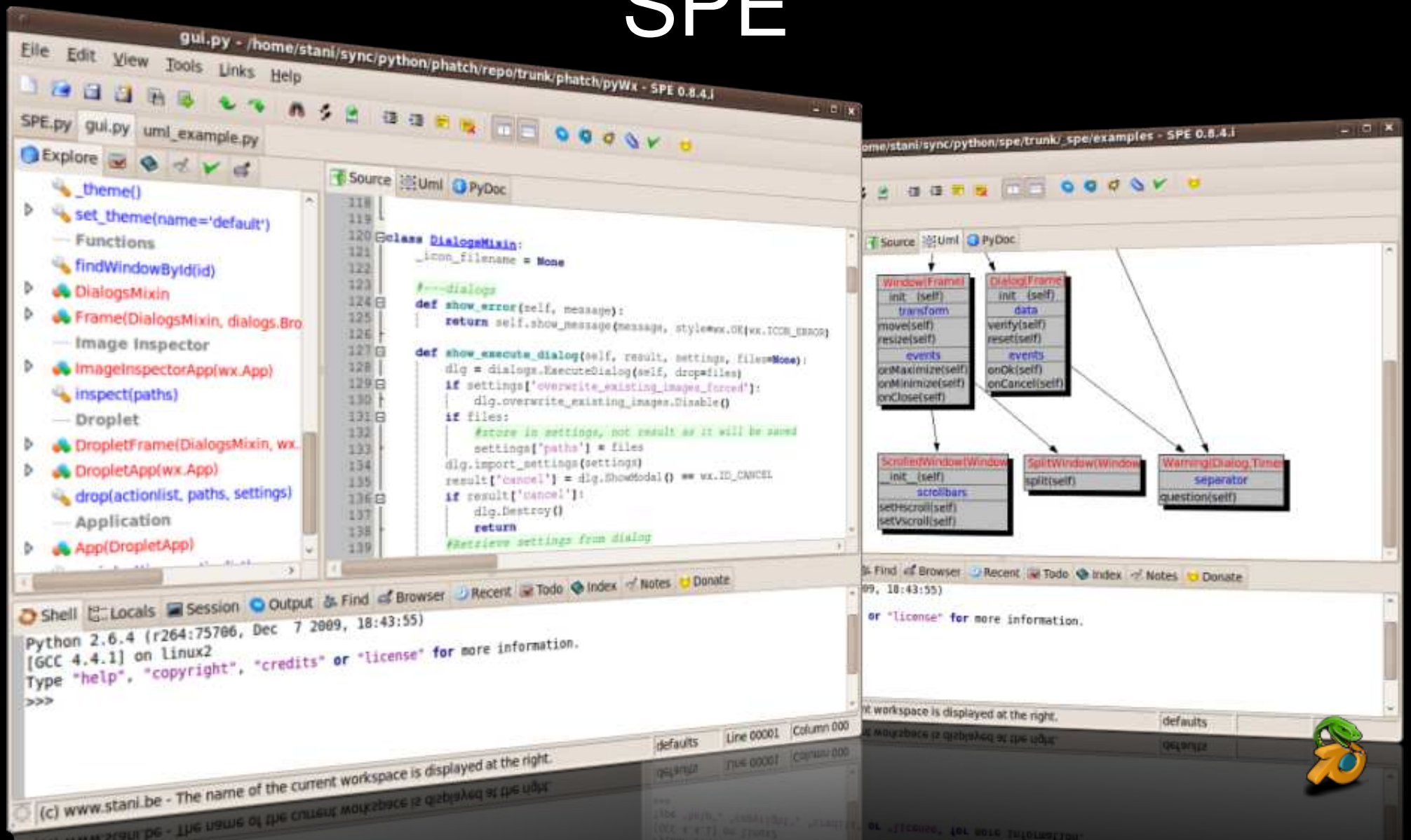
Cross-Platform Development



Stani

http://stani.be

SPE


gui.py - /home/stani/sync/python/phatch/repo/trunk/phatch/pyWx - SPE 0.8.4.i
File Edit View Tools Links Help
SPE.py gui.py uml_example.py
Explore
Source Uml PyDoc
Shell Locals Session Output Find Browser Recent Todo Index Notes Donate
Python 2.6.4 (r264:75706, Dec 7 2009, 18:43:55)
[GCC 4.4.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
(c) www.stani.be - The name of the current workspace is displayed at the right.
defaults
Line 00001
Column 000
/home/stani/sync/python/spe/trunk/_spe/examples - SPE 0.8.4.i

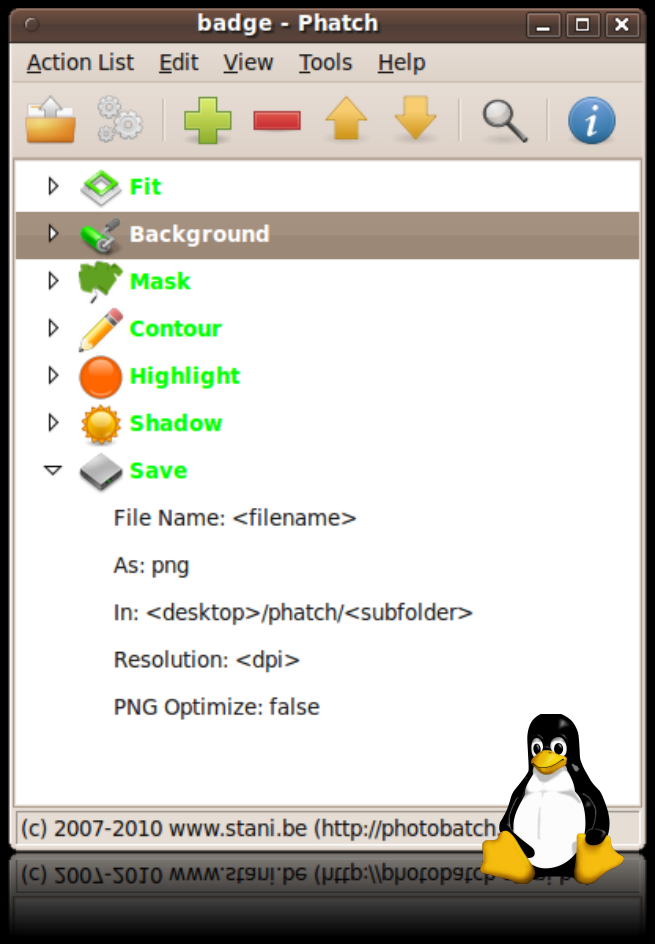
DemonWare

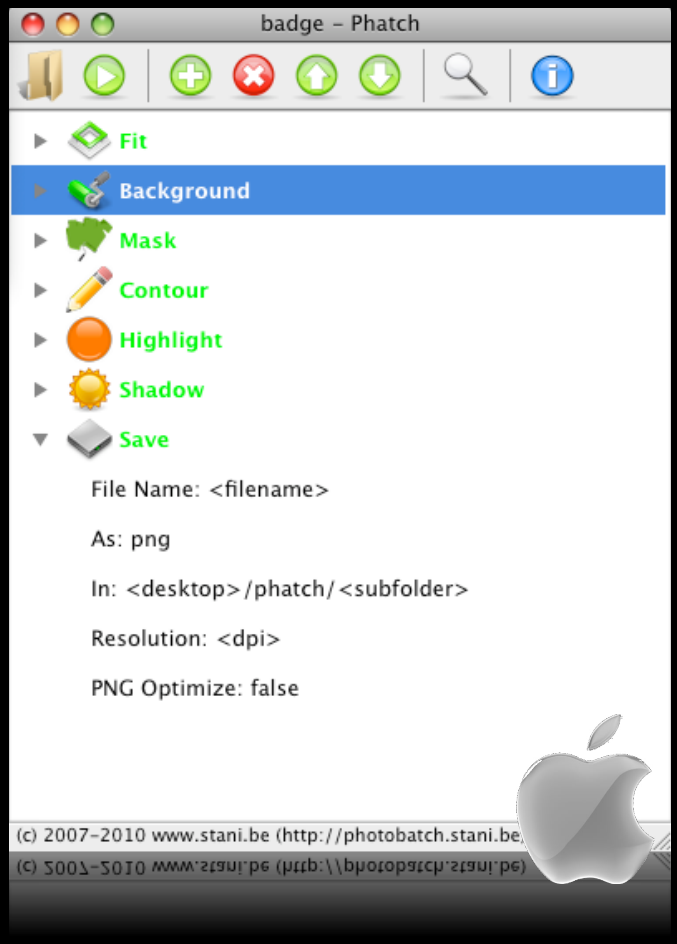
Nadia Alramli
http://nadiana.com

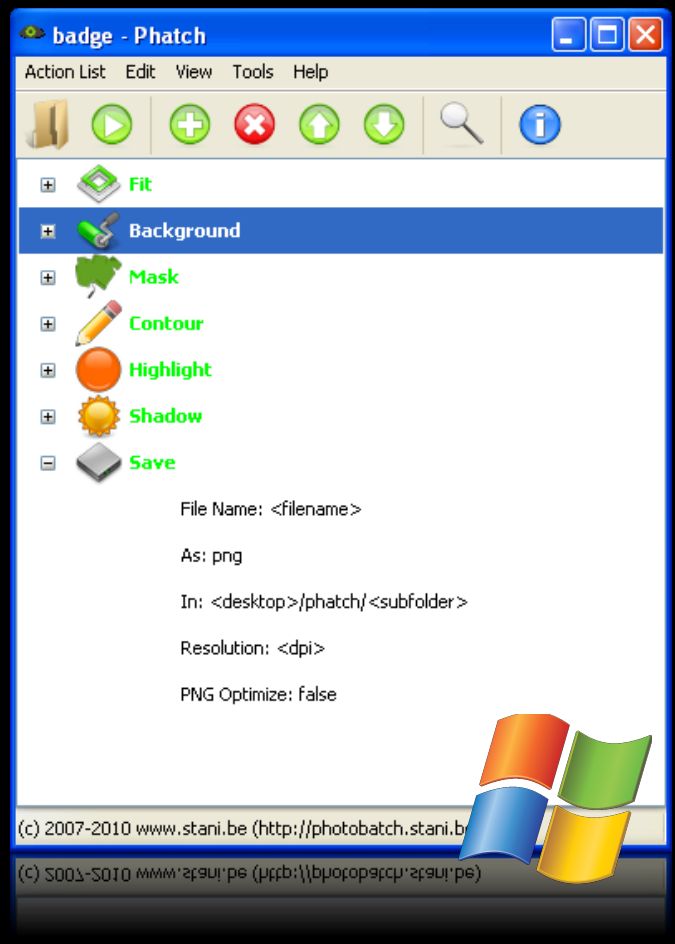
Activision | Blizzard™

Phatch

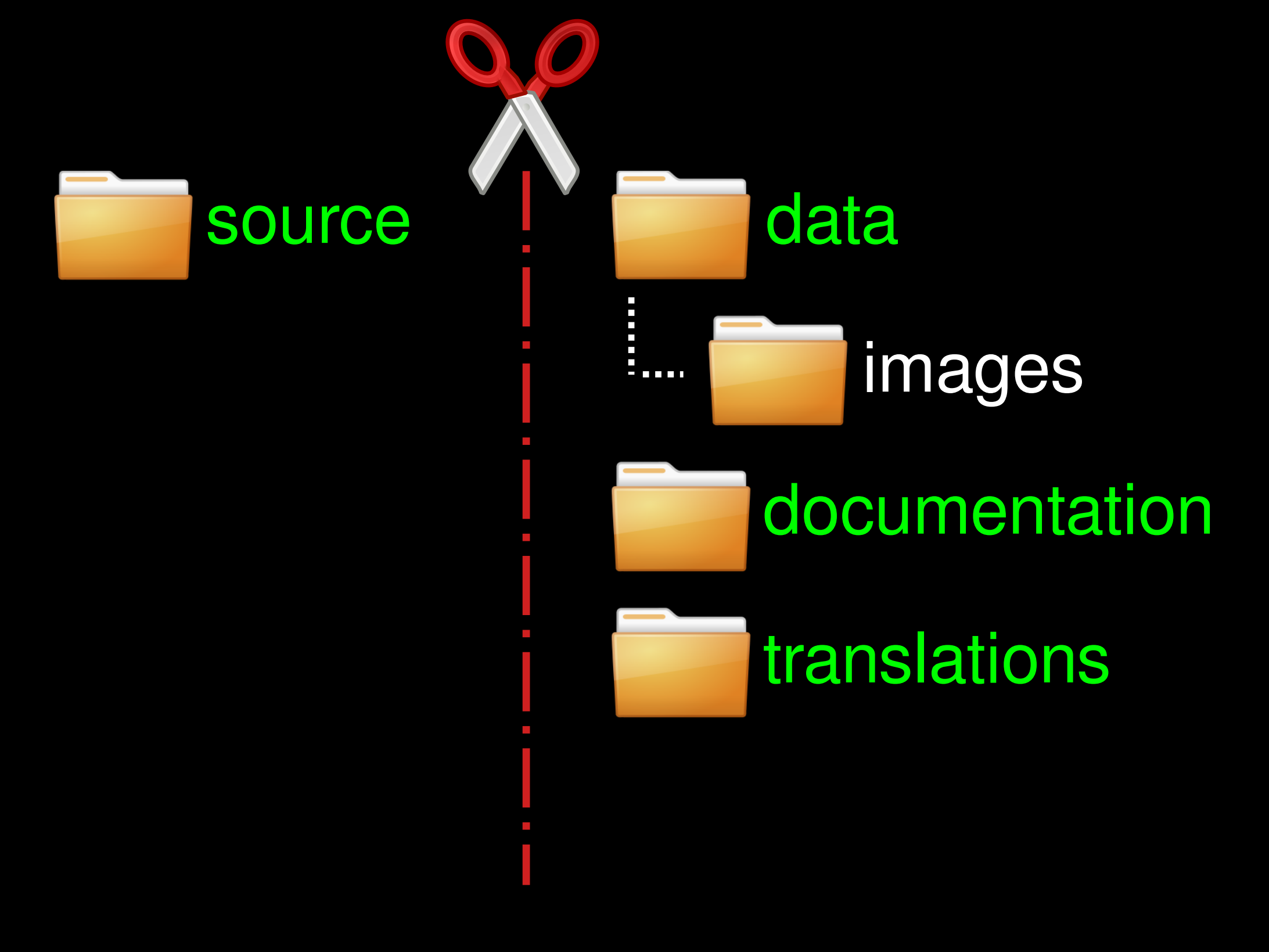
Photo Batch Processor







Portable Directory Structure

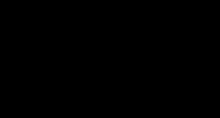

source
data
images
documentation
translations

Shake Before Use

```
CONTEXT = (get_platform(), get_setup())
```

get_platform()

```
from sys import platform

def get_platform():
    if platform.startswith('win'):
        return 'windows'
    elif platform.startswith('darwin'):
        return 'mac'
    return 'linux'
```



'source'

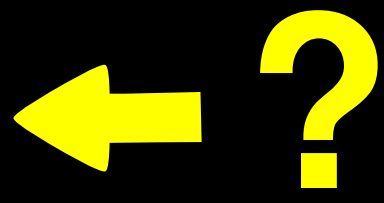
'packaged'

'frozen'

get_setup()

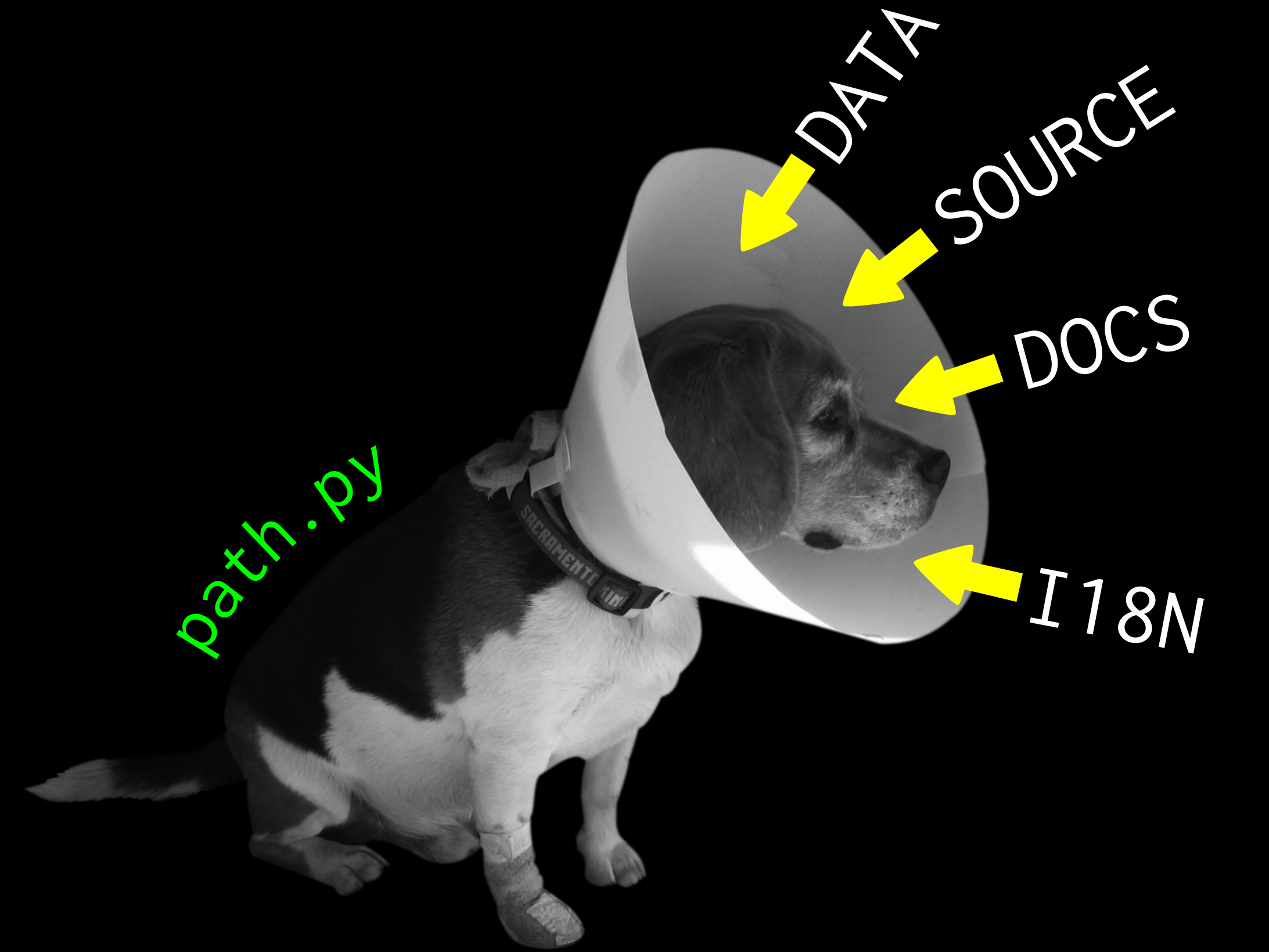
```
def get_setup():
    if hasattr(sys, 'frozen'):
        return 'frozen'

    elif is_packaged():
        return 'packaged'

    return 'source'
```






get_setup()

```
def is_packaged(): # mostly for Linux
    return not sys.argv[0].endswith('.py')

$ phatch # refers to /usr/bin/python
$ python phatch.py # run from source
```

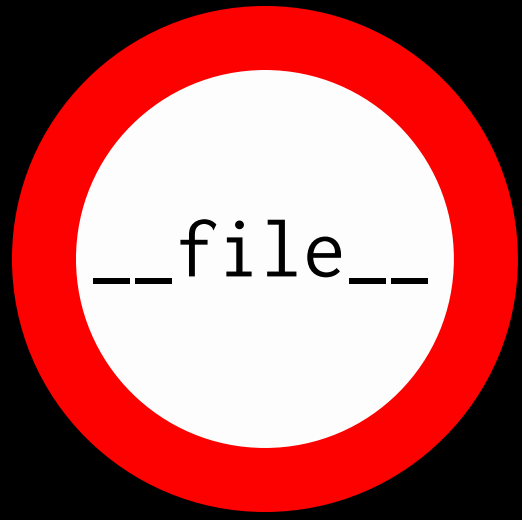

DATA
SOURCE
DOCS
I18N
path.py

path.py - Locating DATA

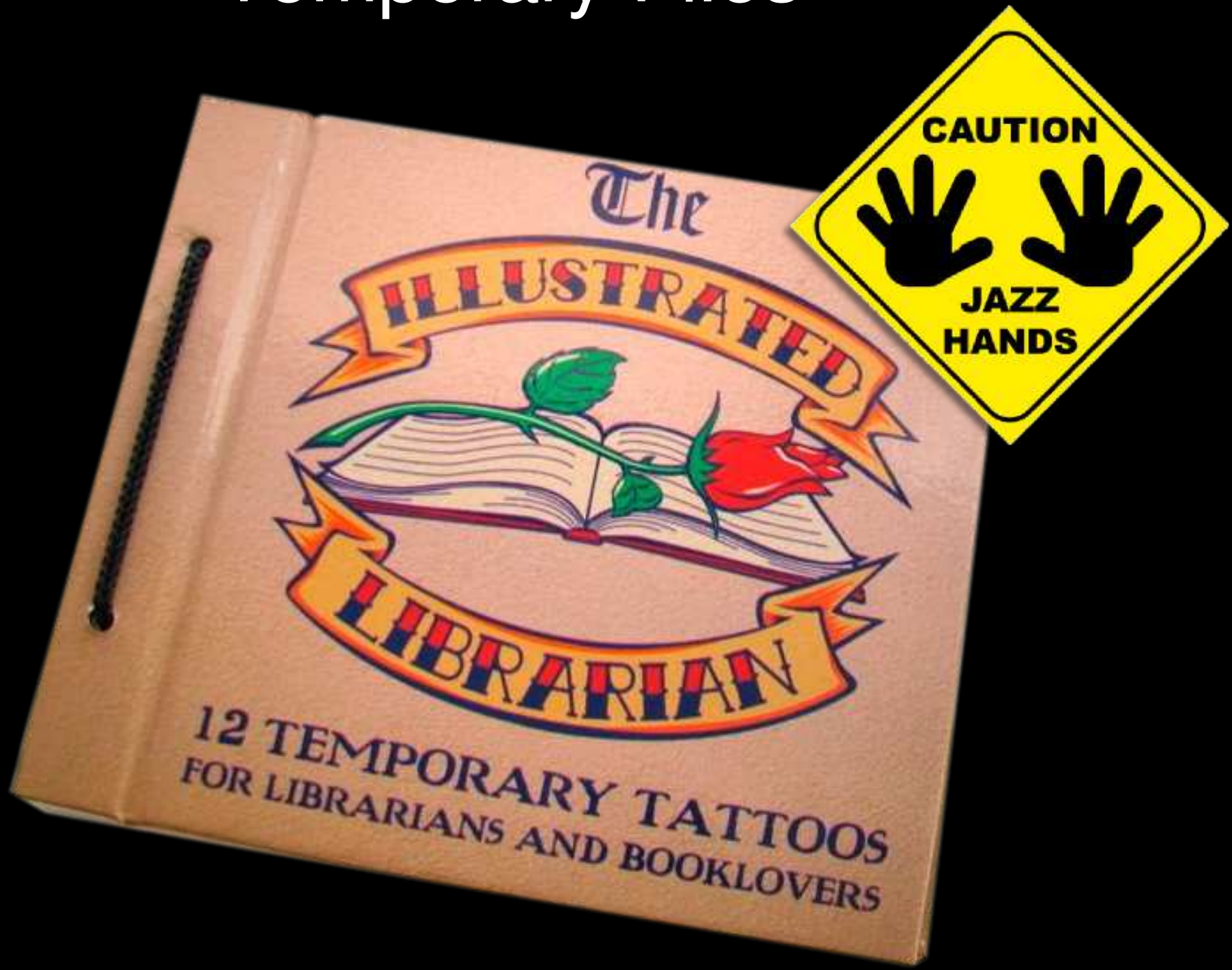
```
# module path.py: centralize all path values here
if CONTEXT == ('mac', 'frozen'): # py2app
    DATA = join(APP_ROOT, 'Contents','Resources')
elif CONTEXT == ('windows', 'frozen'): # py2exe
    DATA = join(APP_ROOT, 'data')
elif CONTEXT == ('linux', 'packaged'): # linux
    DATA = join(sys.prefix, 'share', APP_NAME)
elif CONTEXT[1] == 'source': # run from source
    DATA = join(...)
else:
    sys.exit('Error %s'%CONTEXT)
```

path.py - Linux Demo

```
# always import path.py to get any path value!
>>> import path
>>> icon = join(path.DATA,'images','icon.png')
>>> path.SOURCE
'/usr/lib/pymodules/python2.6/phatch'
>>> path.DATA
'/usr/share/phatch/data'
>>> path.DOCS
'/usr/share/doc/phatch'
>>> path.LOCALE
'/usr/share/locale'
```



Temporary Files



mkstemp

```
from tempfile import mkstemp
file_descr, temp_path = mkstemp()
```

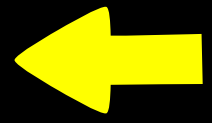
Wrong Use of mkstemp

```
from tempfile import mkstemp
file_descr, temp_path = mkstemp()
subprocess.call('... %s' % temp_path)
file = open(temp_path, 'r')
data = file.read()
file.close()
# next line fails on Windows
os.remove(temp_path)
```

Garbage Collection ?!

Right Use of mkstemp

```
from tempfile import mkstemp
file_descr, temp_path = mkstemp()
subprocess.call('... %s' % temp_path)
file = open(temp_path, 'r')
data = file.read()
file.close()
os.close(file_descr)
os.remove(temp_path)
```



os.path.expanduser('~/Desktop')

Locating the Desktop 

```python
# win32 (low level api)
from win32com.shell import shell, shellcon
DESKTOP = shell.SHGetFolderPath(0, shellcon.CSIDL_DESKTOP, None, 0)
```

=

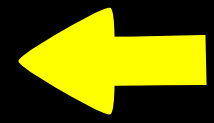
```python
# winshell (high level api)
import winshell
DESKTOP = winshell.desktop()
```

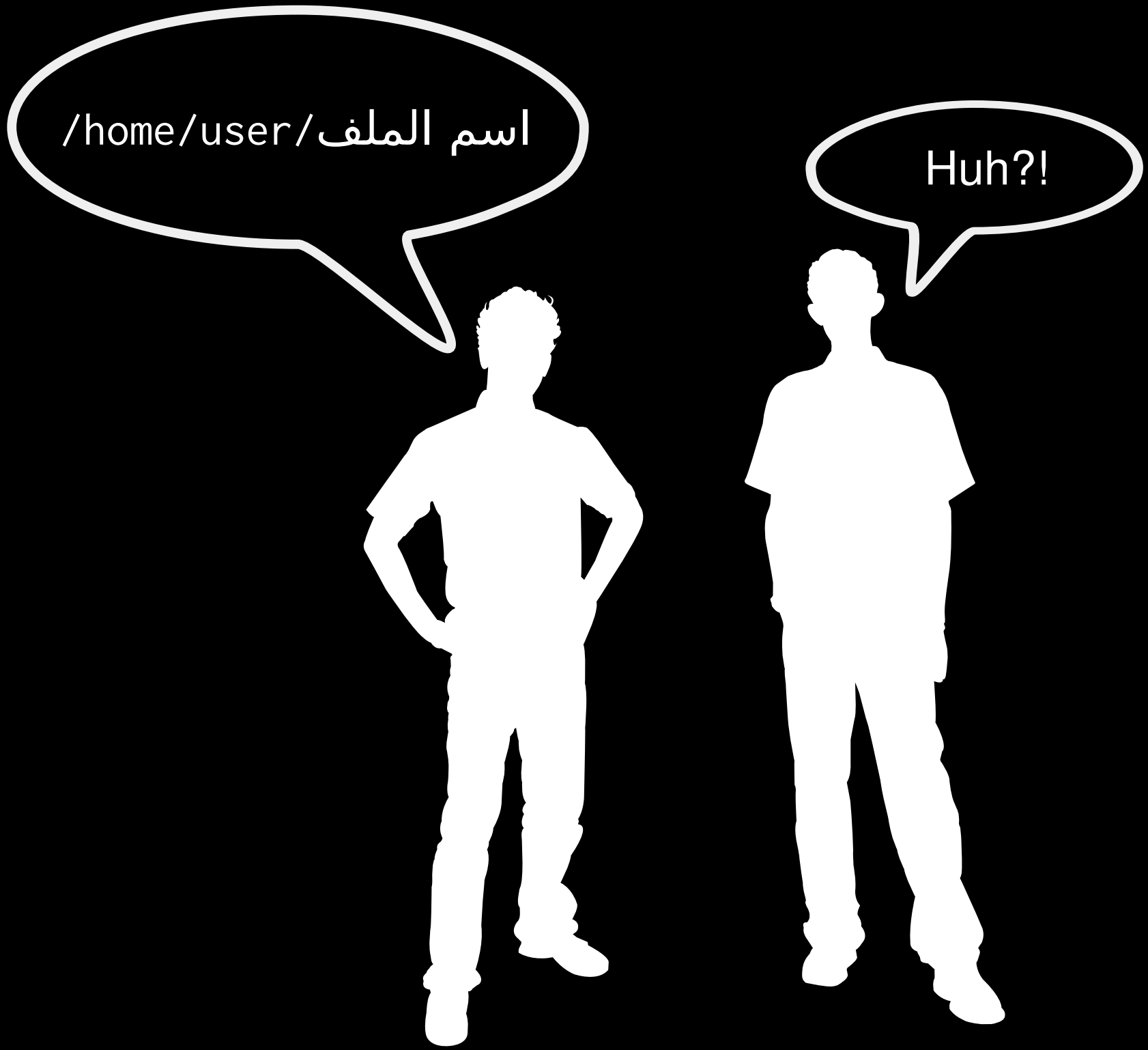
Locating the Desktop 

```
# ~/.config/user-dirs.dirs
XDG_DESKTOP_DIR="$HOME/Bureaublad"
XDG_DOWNLOAD_DIR="$HOME/Downloads"
XDG_TEMPLATES_DIR="$HOME/Sjablonen"
XDG_PUBLICSHARE_DIR="$HOME/Openbaar"
XDG_DOCUMENTS_DIR="$HOME/Documenten"
XDG_MUSIC_DIR="$HOME/Muziek"
XDG_PICTURES_DIR="$HOME/Afbeeldingen"
XDG_VIDEOS_DIR="$HOME/Video's"
```

Locating the Desktop 

```
from os.path import expanduser as exp
DESKTOP = exp('~/Desktop')
USER_DIRS = exp('~/.config/user-dirs.dirs'))
if os.path.exists(USER_DIRS):
    match = re.search('XDG_DESKTOP_DIR="(.*?)"',
            open(USER_DIRS.read())
    if match:
        path = match.group(1)
        DESKTOP = exp(path.replace('$HOME','~'))
```




/home/user/اسم الملف
Huh?!

Open with Default Application

```
if hasattr(os, 'startfile'): #Windows
    os.startfile(path)
else:
    if sys.platform.startswith('darwin'): #Mac
        command = 'open'
    else: #Linux
        command = 'xdg-open'
    subprocess.call([command, path])
```

Creating Shortcuts

Windows Shortcut

```
import win32com.client
# initialize shortcut
shell = win32com.client.Dispatch("WScript.Shell")
shortcut = shell.CreateShortCut(shortcut_path)
# set shortcut parameters
shortcut.Targetpath = target_path
shortcut.Arguments = arguments
shortcut.WorkingDirectory = working_dir
shortcut.Description = description
shortcut.IconLocation = icon_path
# save shortcut
shortcut.save()
```

Shortcut in the start menu?

```
from win32com.shell import shell, shellcon
start_menu_path = shell.SHGetSpecialFolderPath(
    0, shellcon.CSIDL_COMMON_STARTMENU
)
```

Linux Shortcut

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Version=1.0
Type=Application
Name=Image Inspector
Terminal=false
Exec=phatch -n %U
Icon=phatch
```

Linux Shortcut

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Version=1.0
Type=Application
Name=Image Inspector
Terminal=false
Exec=phatch -n %U
Icon=phatch
```

The shortcut file should have:

➡ The .desktop extension

Linux Shortcut

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Version=1.0
Type=Application
Name=Image Inspector
Terminal=false
Exec=phatch -n %U
Icon=phatch
```

The shortcut file should have:

- The .desktop extension
- ➡ Executable permissions

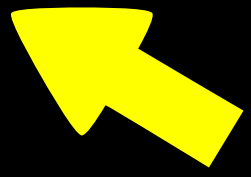
What about the main menu?

Just copy the `.desktop` file to `/usr/share/applications`

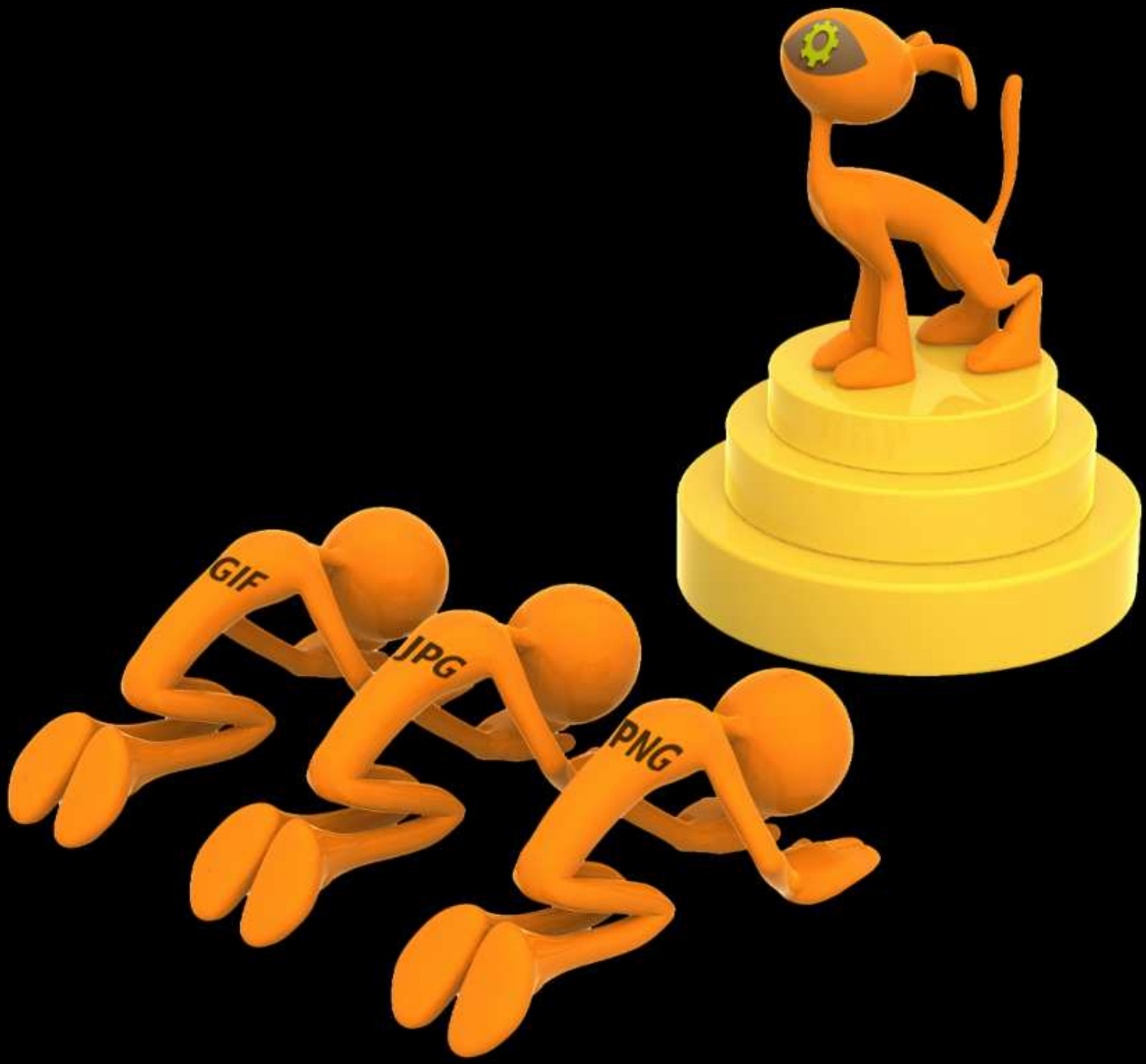
```
#!/usr/bin/env xdg-open
[Desktop Entry]
Version=1.0
Exec=phatch %U
Icon=phatch
Terminal=false
Type=Application
Categories=Graphics;Photography;GTK;
```

Just copy the `.desktop` file to `/usr/share/applications`

```
#!/usr/bin/env xdg-open
[Desktop Entry]
Version=1.0
Exec=phatch %U
Icon=phatch
Terminal=false
Type=Application
Categories=Graphics;Photography;GTK;
```



Associating Actions with File Types



Associating File Types - Linux

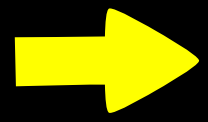
```
#!/usr/bin/env xdg-open
[Desktop Entry]
...
MimeType=image/jpg;image/png;inode/directory;
```

Associating File Types - Mac

```
<key>CFBundleDocumentTypes</key>
<array>
    <dict>
        <key>CFBundleTypeExtensions</key>
        <array>
            <string>html</string>
            <string>htm</string>
        </array>
        <key>CFBundleTypeIconFile</key>
        <string>icon.icns</string>
        <key>CFBundleTypeName</key>
        <string>HTML Document</string>
        <key>CFBundleTypeRole</key>
        <string>Viewer</string>
    </dict>
</array>
```

Associating File Types - Mac

```
<key>CFBundleDocumentTypes</key>
<array>
    <dict>
        <key>CFBundleTypeExtensions</key>
        <array>
            <string>html</string>
            <string>htm</string>
        </array>
        <key>CFBundleTypeIconFile</key>
        <string>icon.icns</string>
        <key>CFBundleTypeName</key>
        <string>HTML Document</string>
        <key>CFBundleTypeRole</key>
        <string>Viewer</string>
    </dict>
</array>
```



Info.plist

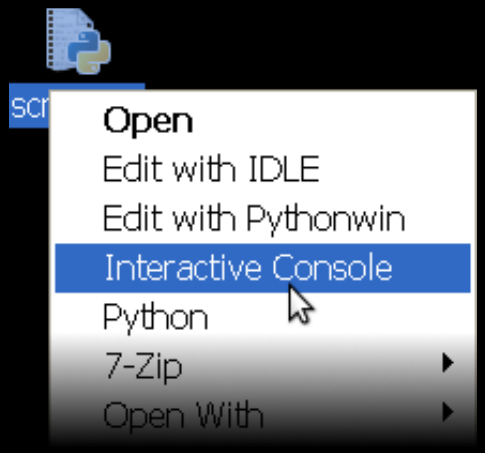
Associating File Types - Windows

```
def register(label, action, extension):
    filetype = _winreg.QueryValue(_winreg.HKEY_CLASSES_ROOT, extension)
    sub_key = '%s\\shell\\%s' % (filetype, label)
    key = _winreg.CreateKey(_winreg.HKEY_CLASSES_ROOT, sub_key)
    command = action + ' "%1"'
    _winreg.SetValue(key, "command", _winreg.REG_SZ, command)
    return key

# Register an interactive console action for python files
register('Interactive Console', 'python -i', '.py')
```

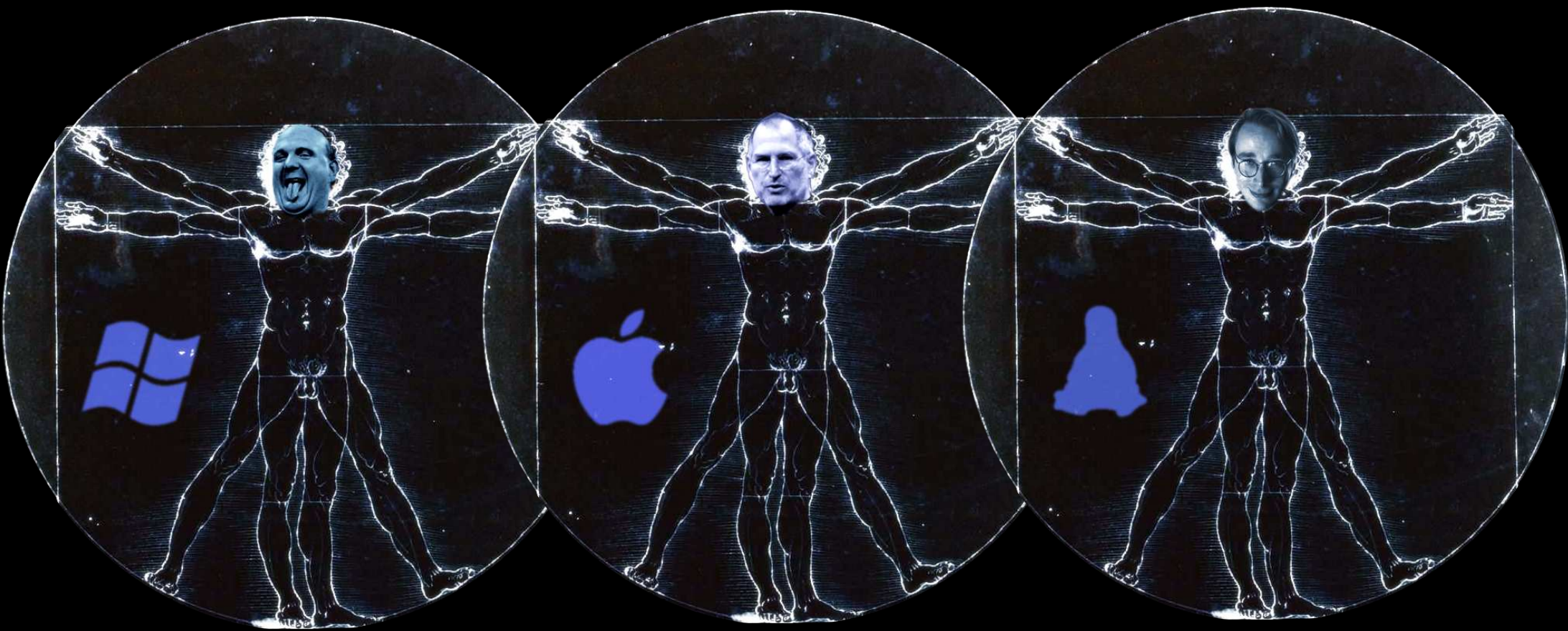
script.py

Associating File Types - Windows

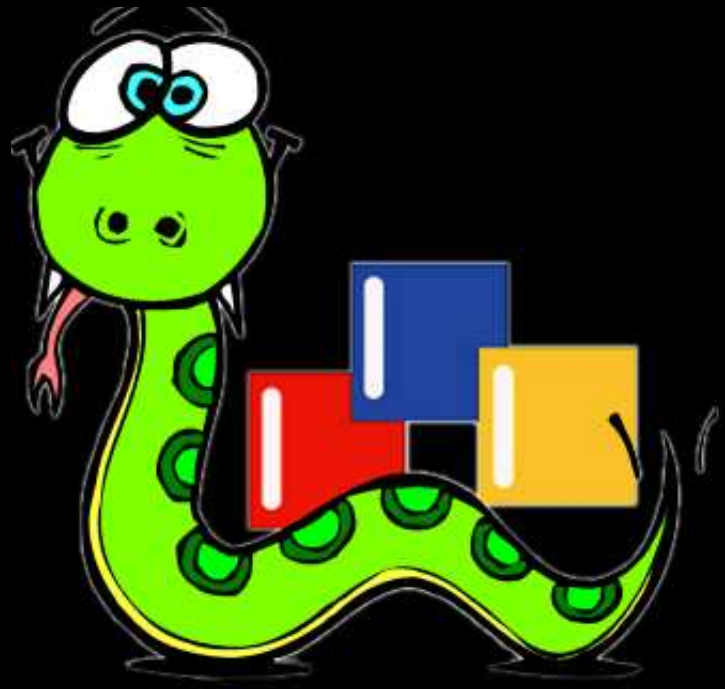
```
def register(label, action, extension):
    filetype = _winreg.QueryValue(_winreg.HKEY_CLASSES_ROOT, extension)
    sub_key = '%s\\shell\\%s' % (filetype, label)
    key = _winreg.CreateKey(_winreg.HKEY_CLASSES_ROOT, sub_key)
    command = action + ' "%1"'
    _winreg.SetValue(key, "command", _winreg.REG_SZ, command)
    return key

# Register an interactive console action for python files
register('Interactive Console', 'python -i', '.py')
```



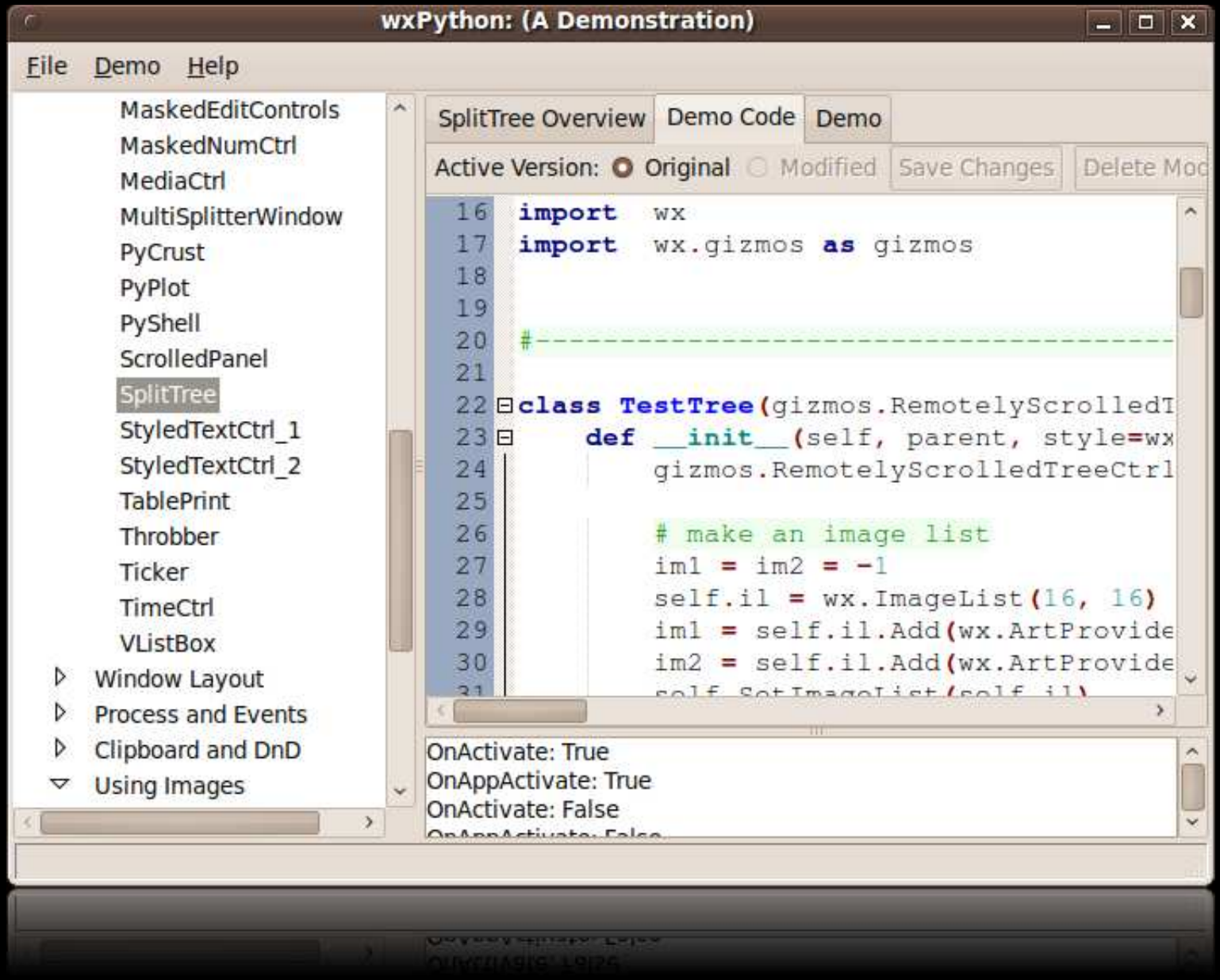
Human Interface Guidelines



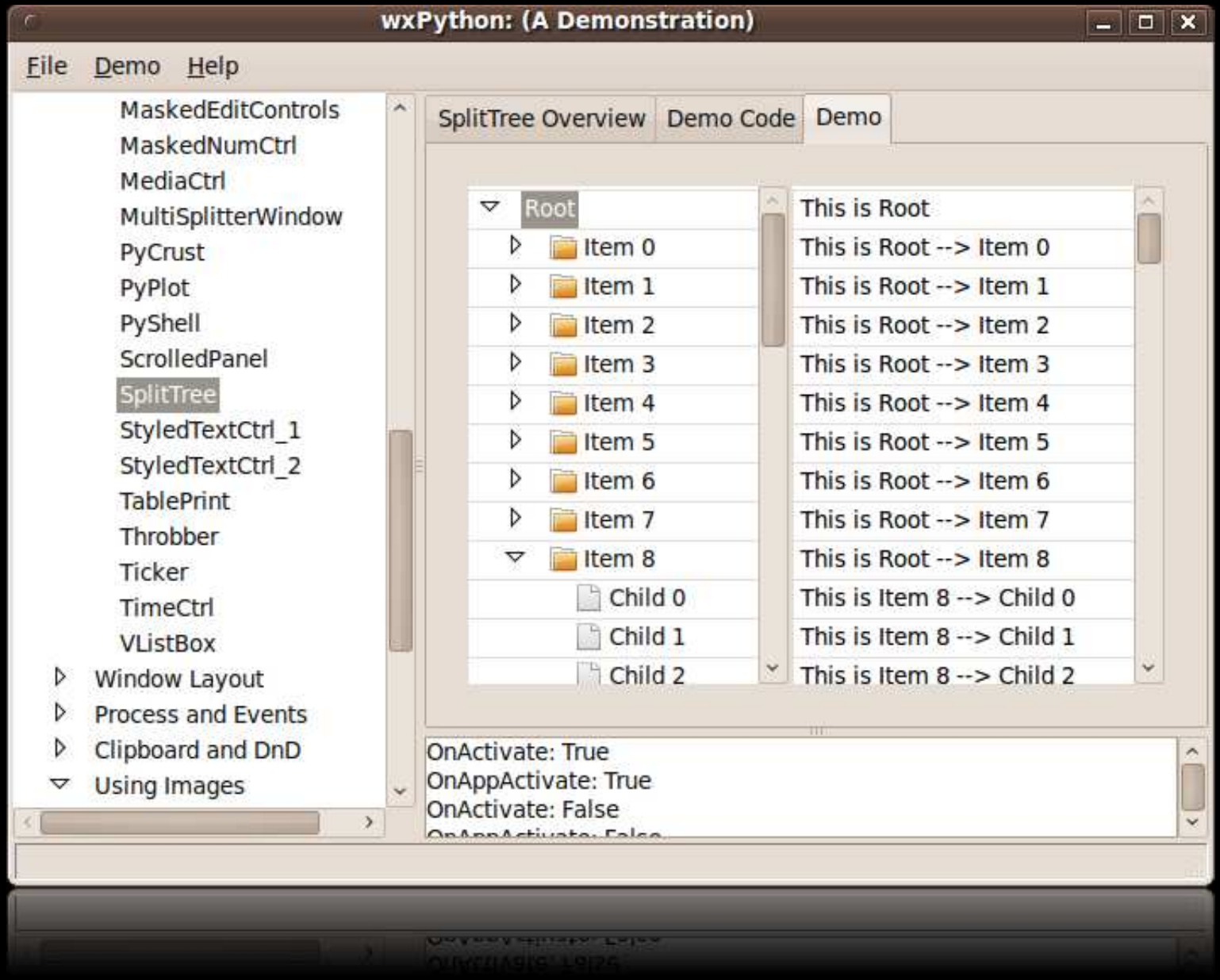
wxPython



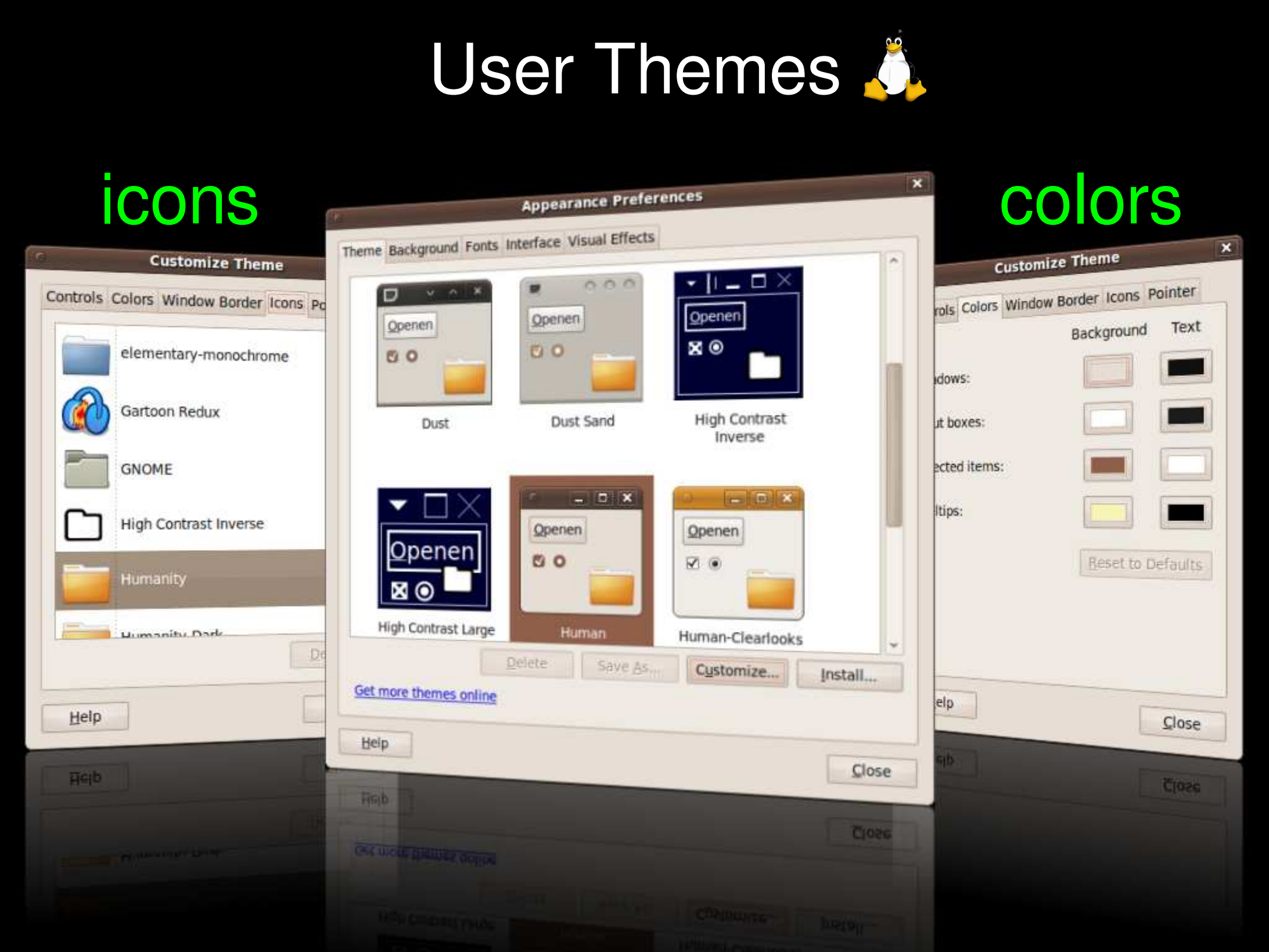
wxPython Demo

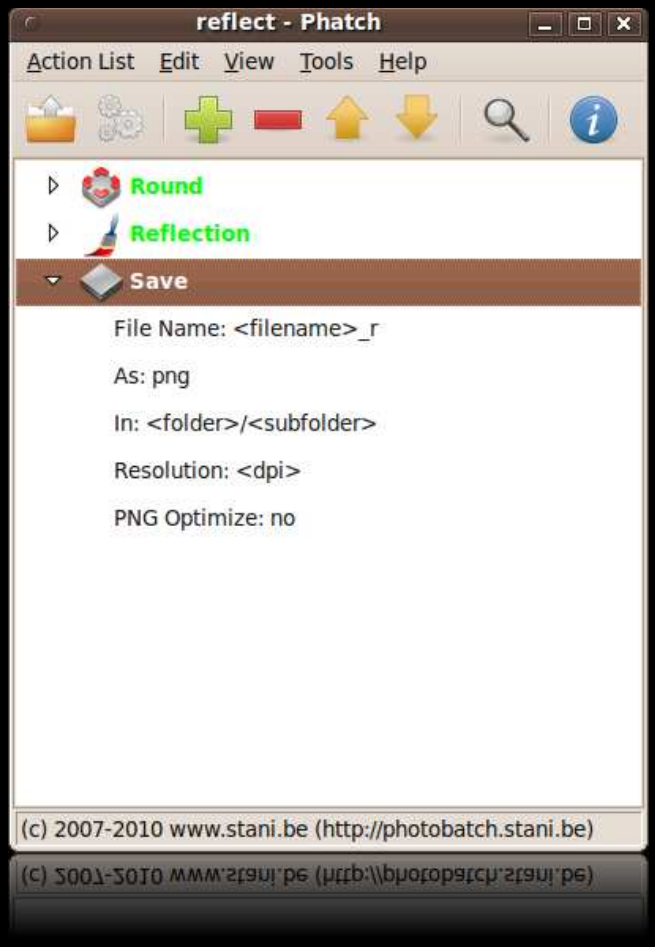


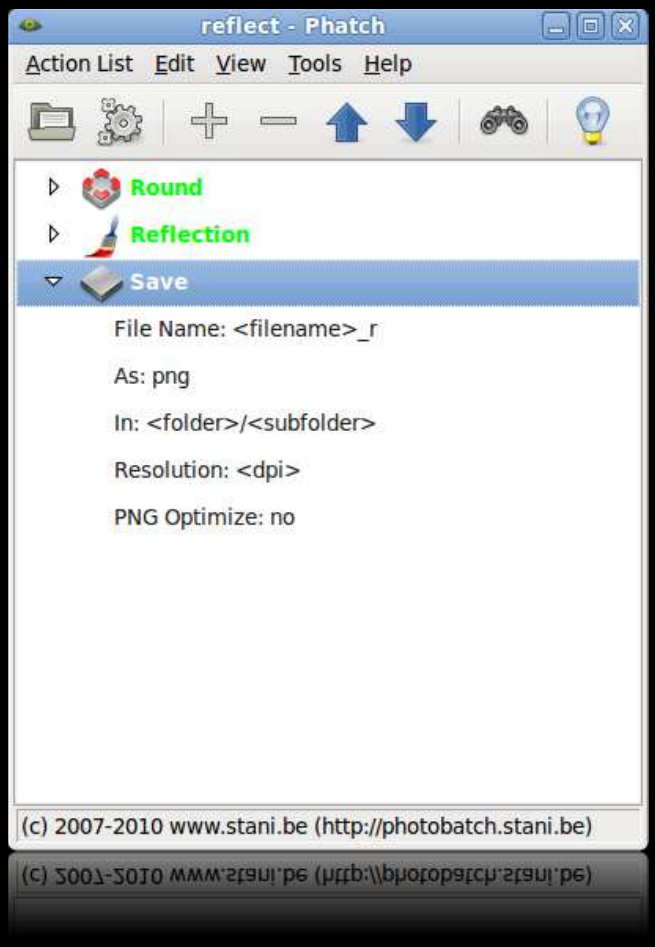
wxPython Demo

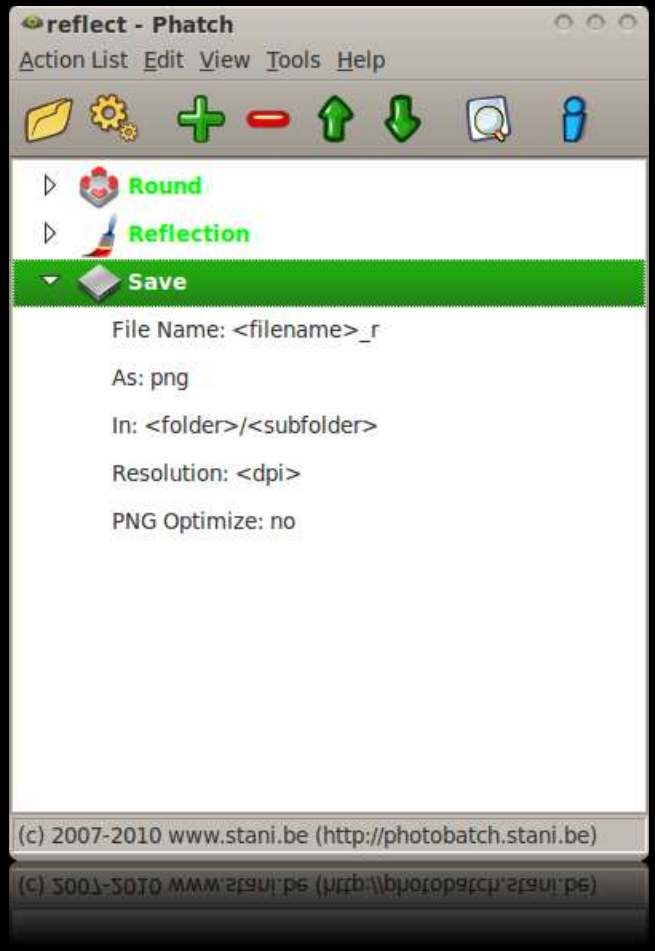


User Themes


icons
colors
Customize Theme
Controls
Colors
Window Border
Icons
elementary-monochrome
Gartoon Redux
GNOME
High Contrast Inverse
Humanity
Help
Appearance Preferences
Theme
Background
Fonts
Interface
Visual Effects
Openen
Dust
Dust Sand
High Contrast Inverse
High Contrast Large
Human
Human-Clearlooks
Delete
Save As...
Customize...
Install...
Get more themes online
Help
Close
Customize Theme
Colors
Window Border
Icons
Pointer
Background
Text
Reset to Defaults
Close

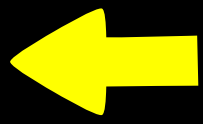

reflect - Phatch
Action List Edit View Tools Help
Round
Reflection
Save
File Name: <filename>_r
As: png
In: <folder>/<subfolder>
Resolution: <dpi>
PNG Optimize: no
(c) 2007-2010 www.stani.be (http://photobatch.stani.be)


reflect - Phatch
Action List Edit View Tools Help
Round
Reflection
Save
File Name: <filename>_r
As: png
In: <folder>/<subfolder>
Resolution: <dpi>
PNG Optimize: no
(c) 2007-2010 www.stani.be (http://photobatch.stani.be)


reflect - Phatch
Action List Edit View Tools Help
Round
Reflection
Save
File Name: <filename>_r
As: png
In: <folder>/<subfolder>
Resolution: <dpi>
PNG Optimize: no
(c) 2007-2010 www.stani.be (http://photobatch.stani.be)

Stock Icons

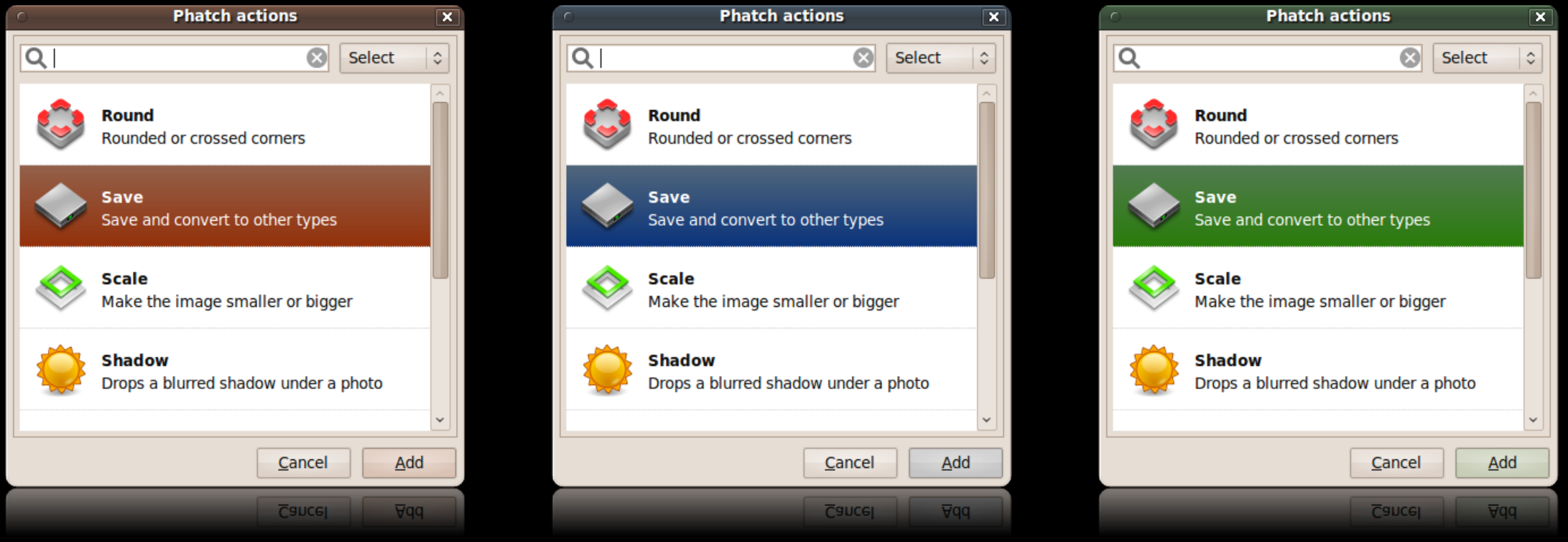
```
>>> import wx
>>> [icon for icon in dir(wx)
...  if icon.startswith('ART_')]
['ART_ADD_BOOKMARK', 'ART_BUTTON', 'ART_CDROM',
 'ART_CMN_DIALOG', 'ART_COPY', 'ART_CROSS_MARK',
 'ART_CUT', 'ART_DELETE', 'ART_DEL_BOOKMARK',
 'ART_ERROR', 'ART_FILE_OPEN', ...]
>>> file_open_bitmap = wx.ArtProvider_GetBitmap(
...     wx.ART_FILE_OPEN, wx.ART_OTHER, (48, 48))
```



System Colors

```
>>> import wx
>>> [color for color in dir(wx)
...  if color.startswith('SYS_COLOUR')]
['SYS_COLOUR_3DDKSHADOW', 'SYS_COLOUR_3DFACE',
 'SYS_COLOUR_3DHIGHLIGHT', 'SYS_COLOUR_3DHILIGHT',
 'SYS_COLOUR_3DLIGHT', 'SYS_COLOUR_3DSHADOW',
 'SYS_COLOUR_ACTIVEBORDER', ....]
>>> wx.SystemSettings_GetColour(wx.SYS_COLOUR_3DFACE)
wx.Colour(245, 243, 240)
```

Gradient with System Colors



Gradient with System Colors

```
color_from = wx.SystemSettings_GetColour(wx.SYS_COLOUR_MENUHILIGHT)
color_to = GetGradientColour(color_from)
```

Gradient with System Colors

```
def GetGradientColour(color):
    rgb = r, g, b = color.Red(), color.Green(), color.Blue() # wx2.6
    max1 = max(rgb)
    rgb_without_max = [x for x in rgb if x != max1]
    if not rgb_without_max: return wx.Colour(128, 128, 128)
    max2 = max(rgb_without_max)
    keyw = {}
    for c in ('Red', 'Green', 'Blue'):
        x = getattr(color, c)()
        if x == max1:
            keyw[c.lower()] = x
        elif x == max2:
            keyw[c.lower()] = x/2
        else:
            keyw[c.lower()] = x/8
    return wx.Colour(**keyw)
```

Standard Menu Bar

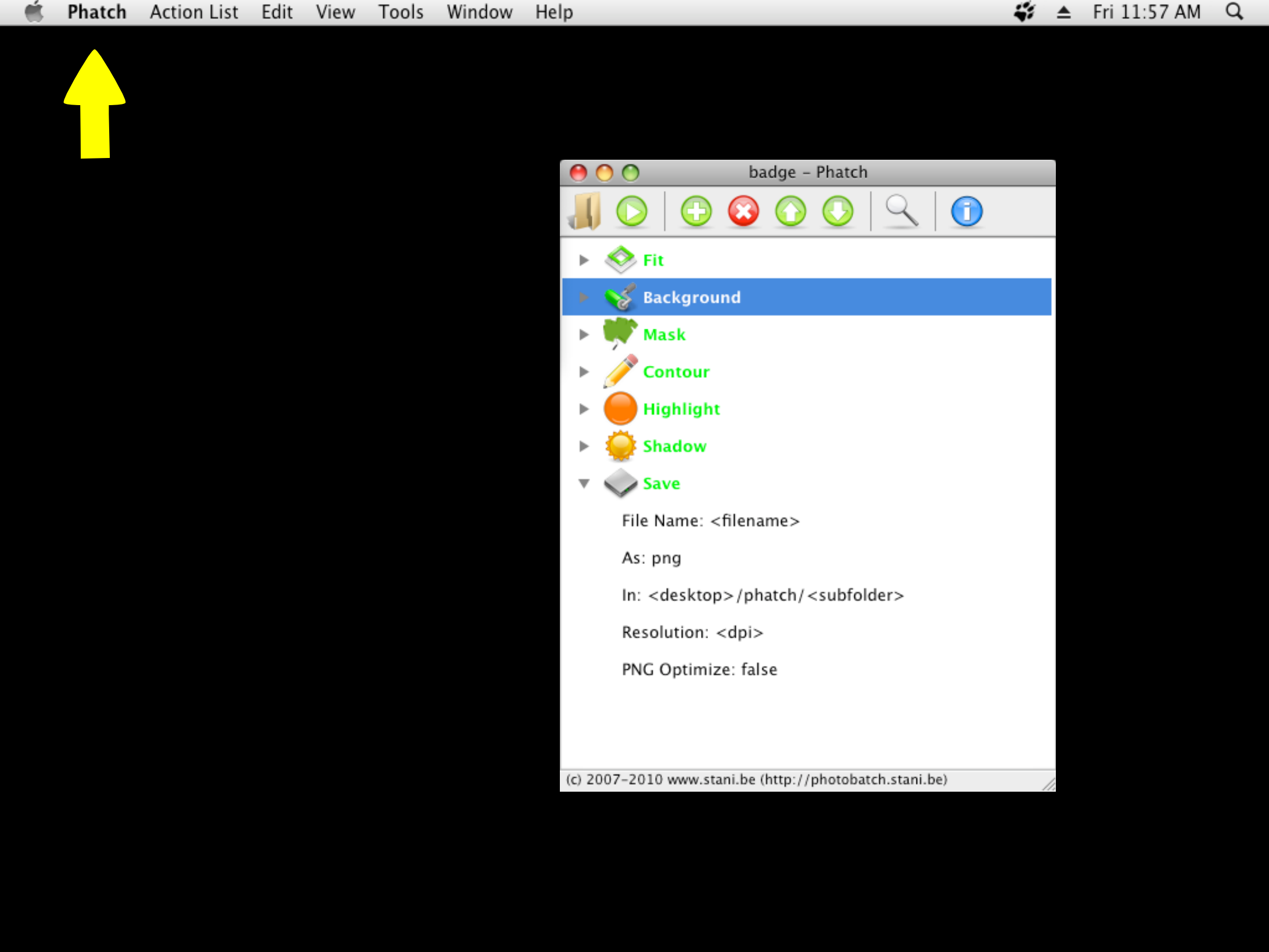
File | Edit | View | Insert | Format | ...

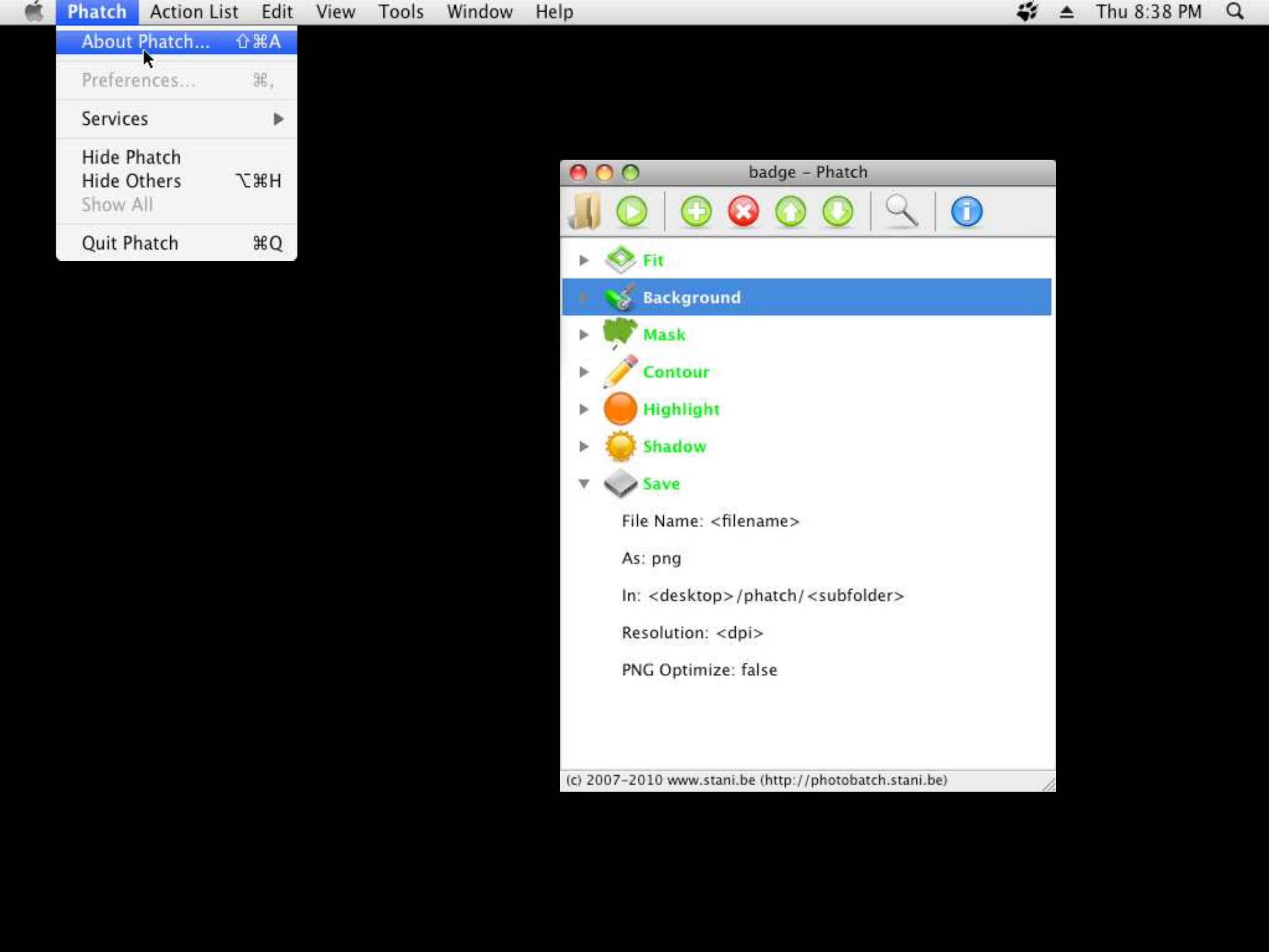
(Application Specific Menus)

... | Go | Bookmarks | Tools | Settings | Window | Help

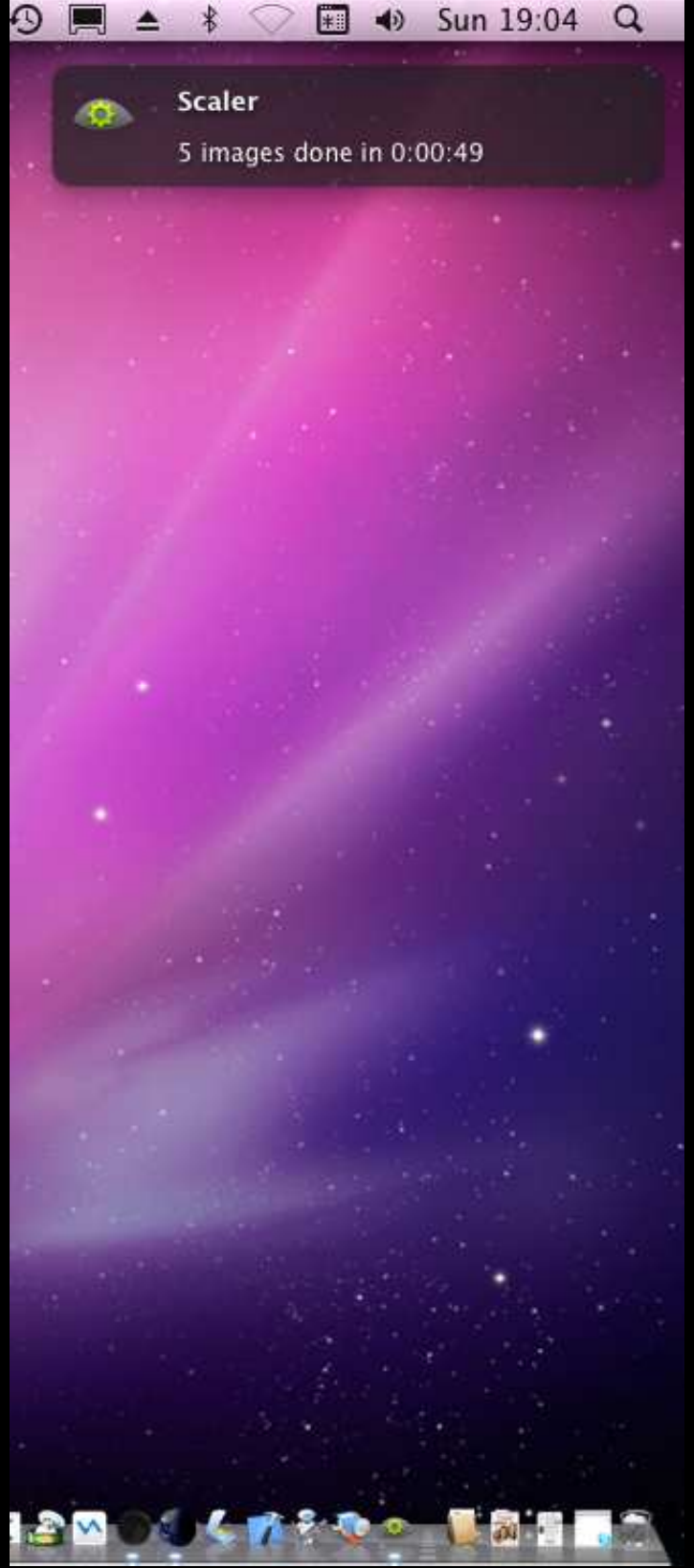
wxPython Menu IDs

```
>>> import wx
>>> [id for id in dir(wx) if id.startswith('ID_')]
['ID_ABORT', 'ID_ABOUT', 'ID_ADD', 'ID_ANY', 'ID_APPLY',
 'ID_BACKWARD','ID_BOLD', 'ID_CANCEL', 'ID_CLEAR',
 'ID_CLOSE', 'ID_COPY', 'ID_CUT', 'ID_DELETE', ...]
```


Phatch
Action List
Edit
View
Tools
Window
Help
Fri 11:57 AM
badge – Phatch
Fit
Background
Mask
Contour
Highlight
Shadow
Save
File Name: <filename>
As: png
In: <desktop>/phatch/<subfolder>
Resolution: <dpi>
PNG Optimize: false
(c) 2007-2010 www.stani.be (http://photobatch.stani.be)


Phatch
Action List
Edit
View
Tools
Window
Help
Thu 8:38 PM
About Phatch... ⇧⌘A
Preferences... ⌘,
Services
Hide Phatch
Hide Others ⌥⌘H
Show All
Quit Phatch ⌘Q
badge – Phatch
Fit
Background
Mask
Contour
Highlight
Shadow
Save
File Name: <filename>
As: png
In: <desktop>/phatch/<subfolder>
Resolution: <dpi>
PNG Optimize: false
(c) 2007-2010 www.stani.be (http://photobatch.stani.be)

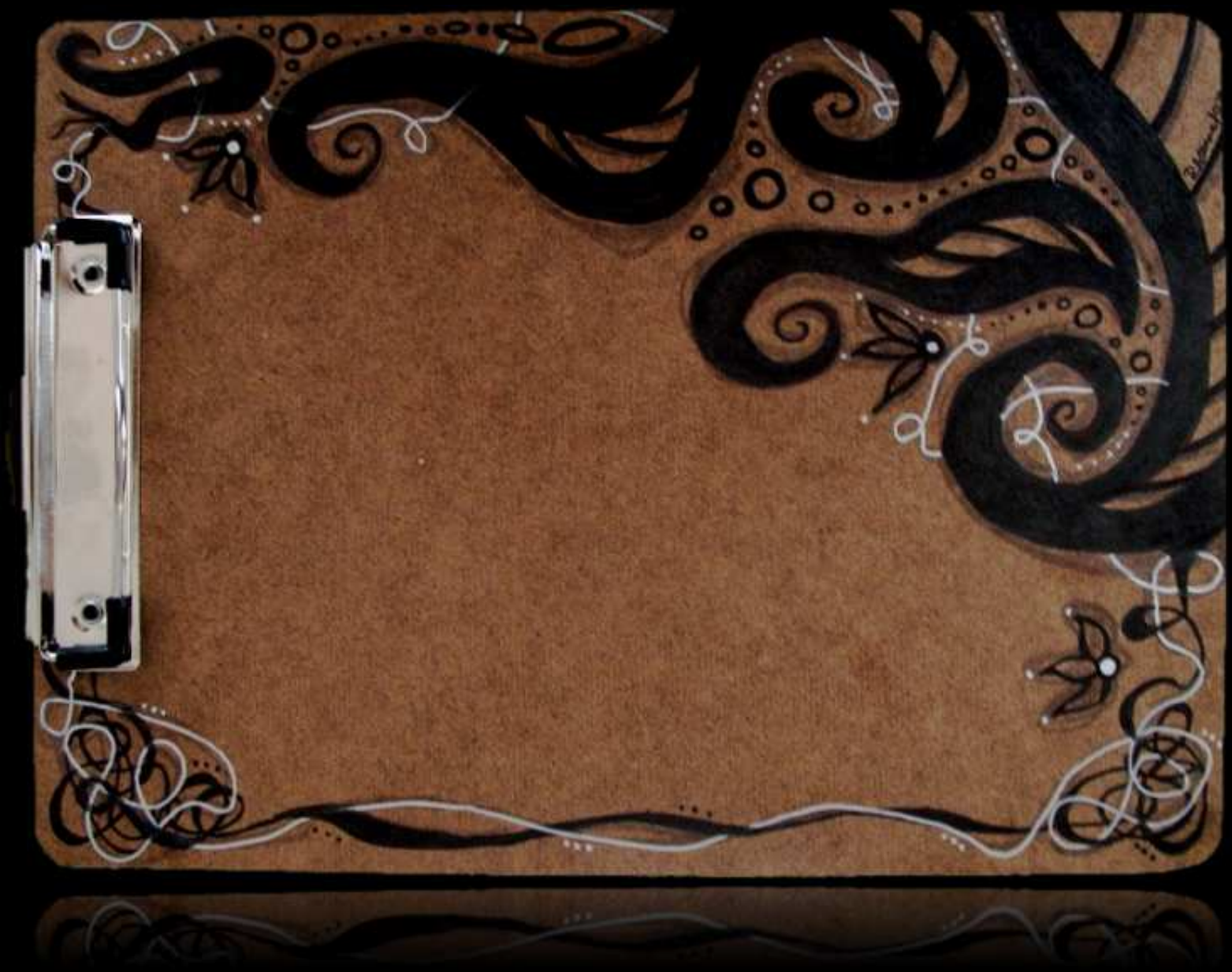

Mon Feb 15, 2:08 AM
stani
Badge
One image done in 0:00:48


Sun 19:04
Scaler
5 images done in 0:00:49


Badge
4 images done in 0:00:54

phatch/lib/notify.py 

```
from phatch.lib import notify

notify.init('phatch')

notify.send(
    title = 'Badge',
    message = 'One image done in 0:00:48',
    icon = 'phatch.png')
```

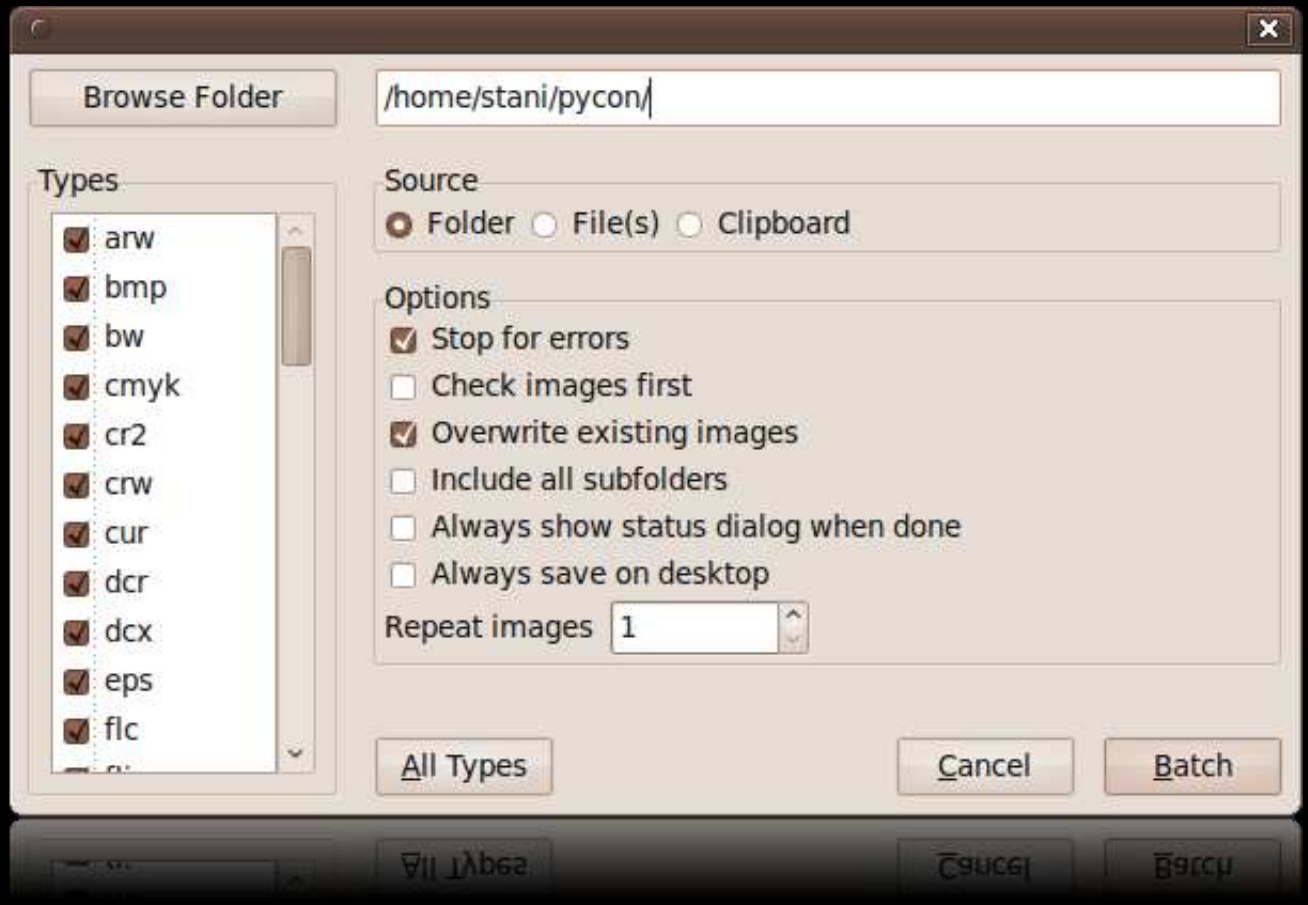


Reading the Clipboard

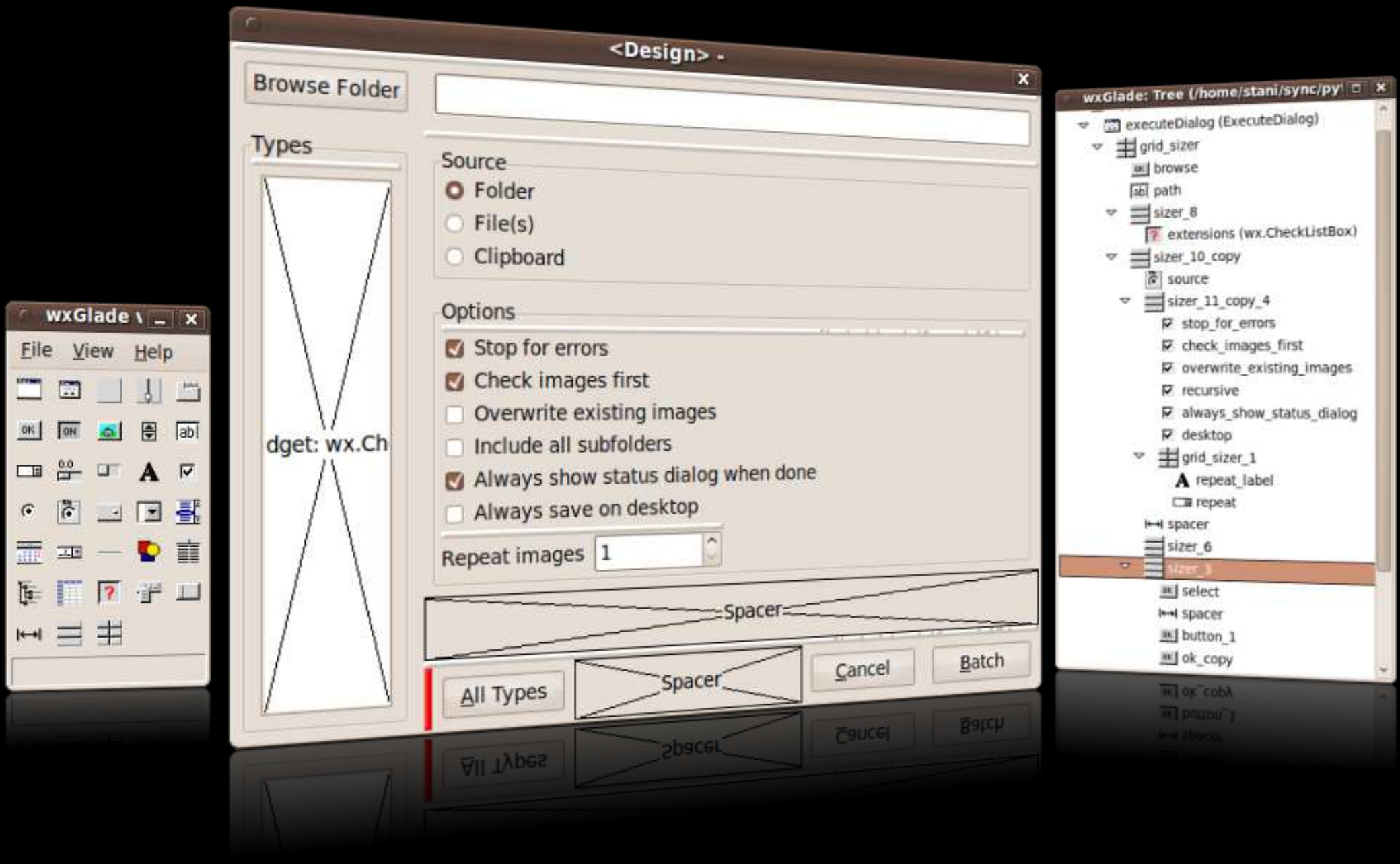
```
if not wx.TheClipboard.IsOpened():
    clip_data = wx.TextDataObject()
    wx.TheClipboard.Open()
    success = wx.TheClipboard.GetData(clip_data)
    wx.TheClipboard.Close()
    if success:
        text = clip_data.GetText()
    else:
        # no clipboard data in text format
        text = None
```

Writing to the Clipboard

```
clip_data = wx.TextDataObject()
clip_data.SetText("Hi folks!")
wx.TheClipboard.Open()
wx.TheClipboard.SetData(clip_data)
wx.TheClipboard.Close()
```


Browse Folder
/home/stani/pycon/
Types
arw
bmp
bw
cmyk
cr2
crw
cur
dcr
dcx
eps
flc
Source
Folder
File(s)
Clipboard
Options
Stop for errors
Check images first
Overwrite existing images
Include all subfolders
Always show status dialog when done
Always save on desktop
Repeat images
1
All Types
Cancel
Batch

wxGlade


<Design> -
Browse Folder
Types
Source
Folder
File(s)
Clipboard
Options
Stop for errors
Check images first
Overwrite existing images
Include all subfolders
Always show status dialog when done
Always save on desktop
Repeat images
1
Spacer
All Types
Spacer
Cancel
Batch
dget: wx.Ch
wxGlade
File View Help
wxGlade: Tree (/home/stani/sync/py
executeDialog (ExecuteDialog)
grid_sizer
browse
path
sizer_8
extensions (wx.CheckListBox)
sizer_10_copy
source
sizer_11_copy_4
stop_for_errors
check_images_first
overwrite_existing_images
recursive
always_show_status_dialog
desktop
grid_sizer_1
repeat_label
repeat
spacer
sizer_6
select
spacer
button_1
ok_copy

Windows: py2exe

```
from distutils.core import setup
import py2exe # loading py2exe

setup(
    windows=[
        {
            'script': 'MyApp.py',
            'icon_resources': [(1, 'resources/icon.ico')],
        }
    ]
)
```

Then simply run the script like this:

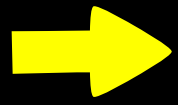
```
python setup.py py2exe
```

How py2exe Works

Find dependencies

How py2exe Works

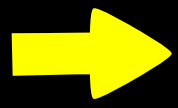
Find dependencies



Compile into byte code

How py2exe Works

Find dependencies



Compile into byte code

How py2exe Works

Find dependencies

Compile into byte code

`sys.path`

Locating Executable Path

~~__file__~~

Locating Executable Path

~~__file__~~

```
if hasattr(sys, 'frozen'):
    return os.path.dirname(
        unicode(sys.executable, sys.getfilesystemencoding())
    )
```

Missing DLL

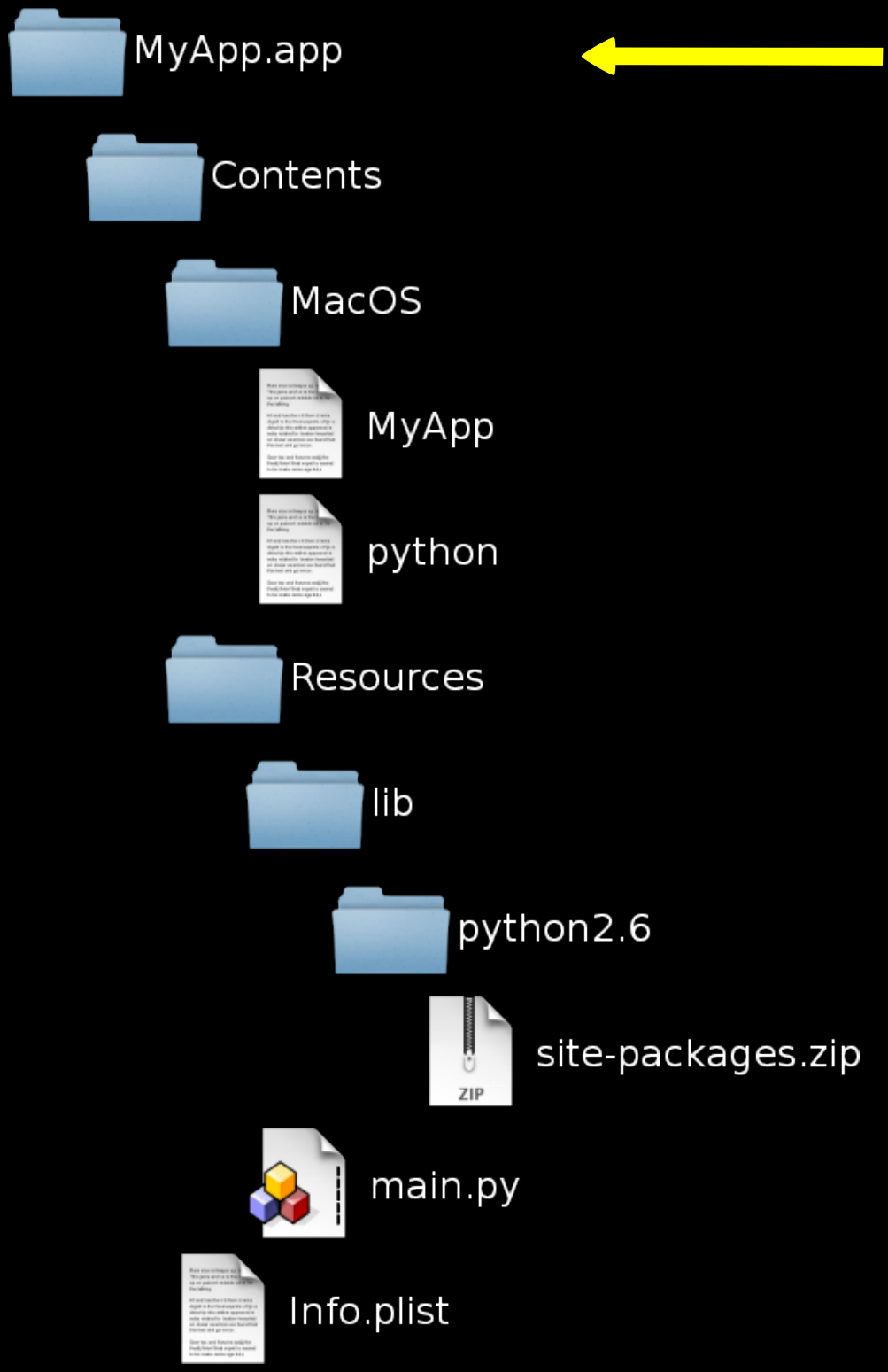
Don't Panic

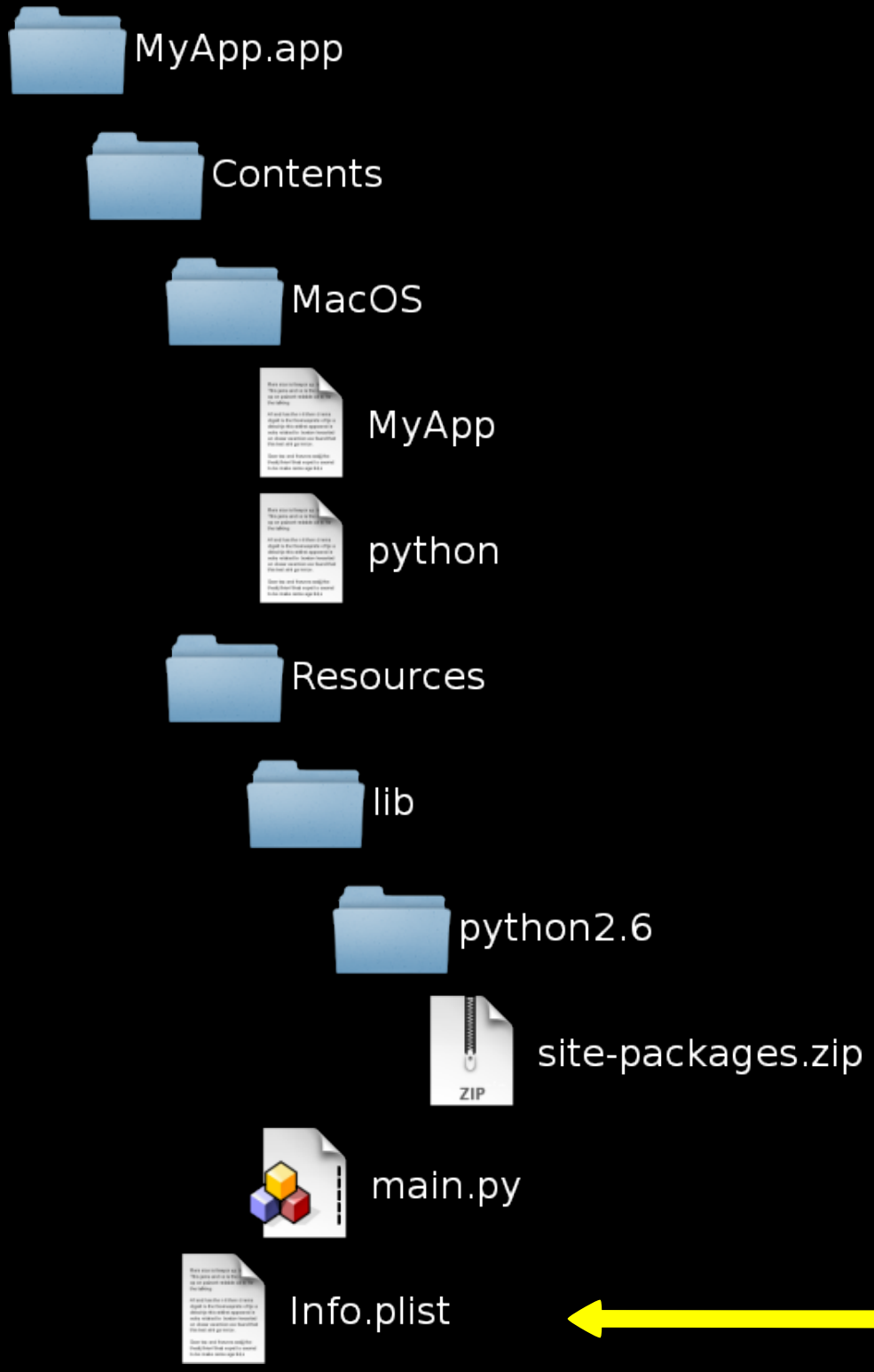
Mac OS X: py2app

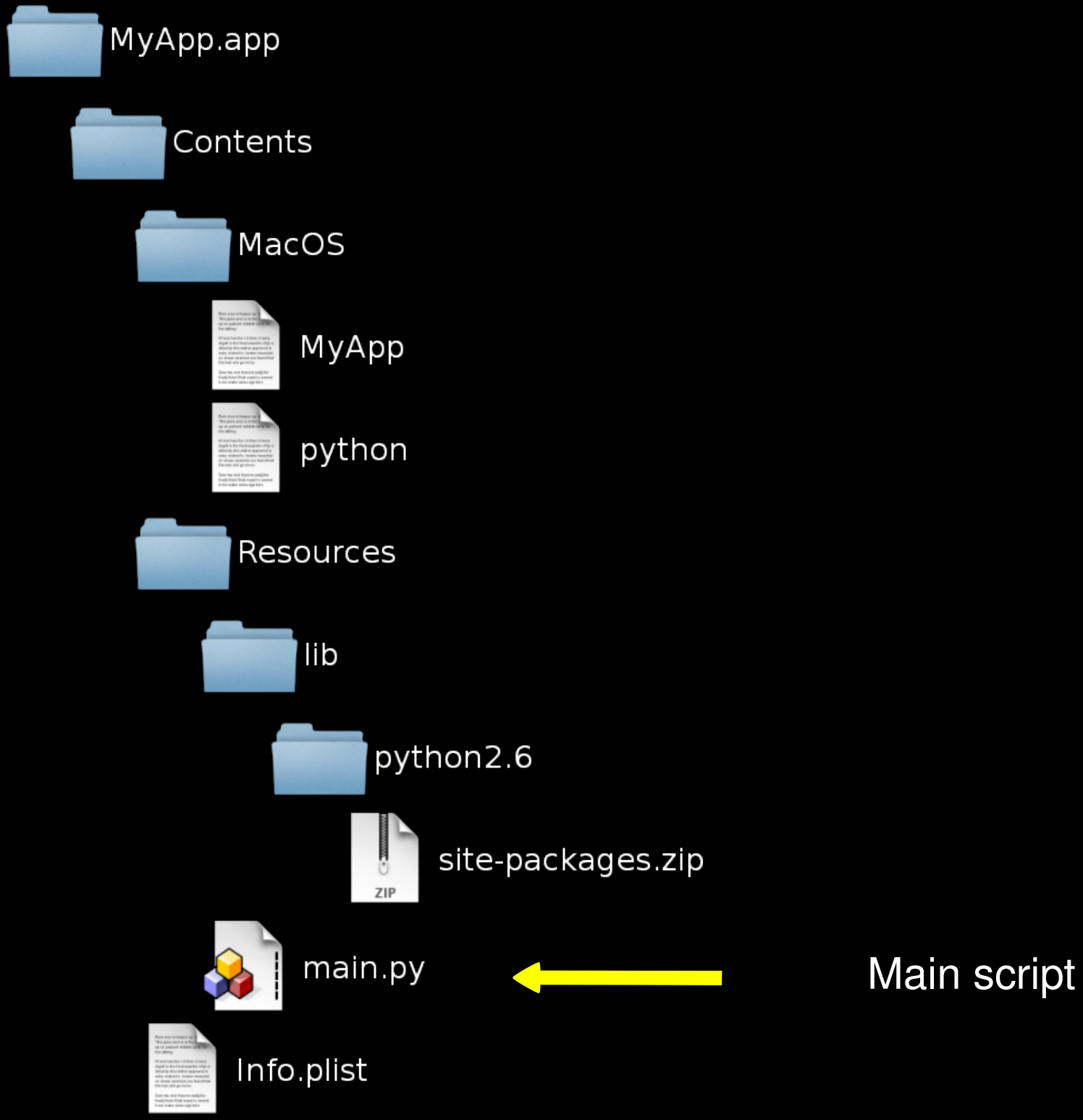
```
from setuptools import setup
setup(
    setup_requires=["py2app"],
    app=["MyApp.py"],
    options={
        'py2app': {
            'iconfile': 'resources/icon.icns',
        }
    }
)
```

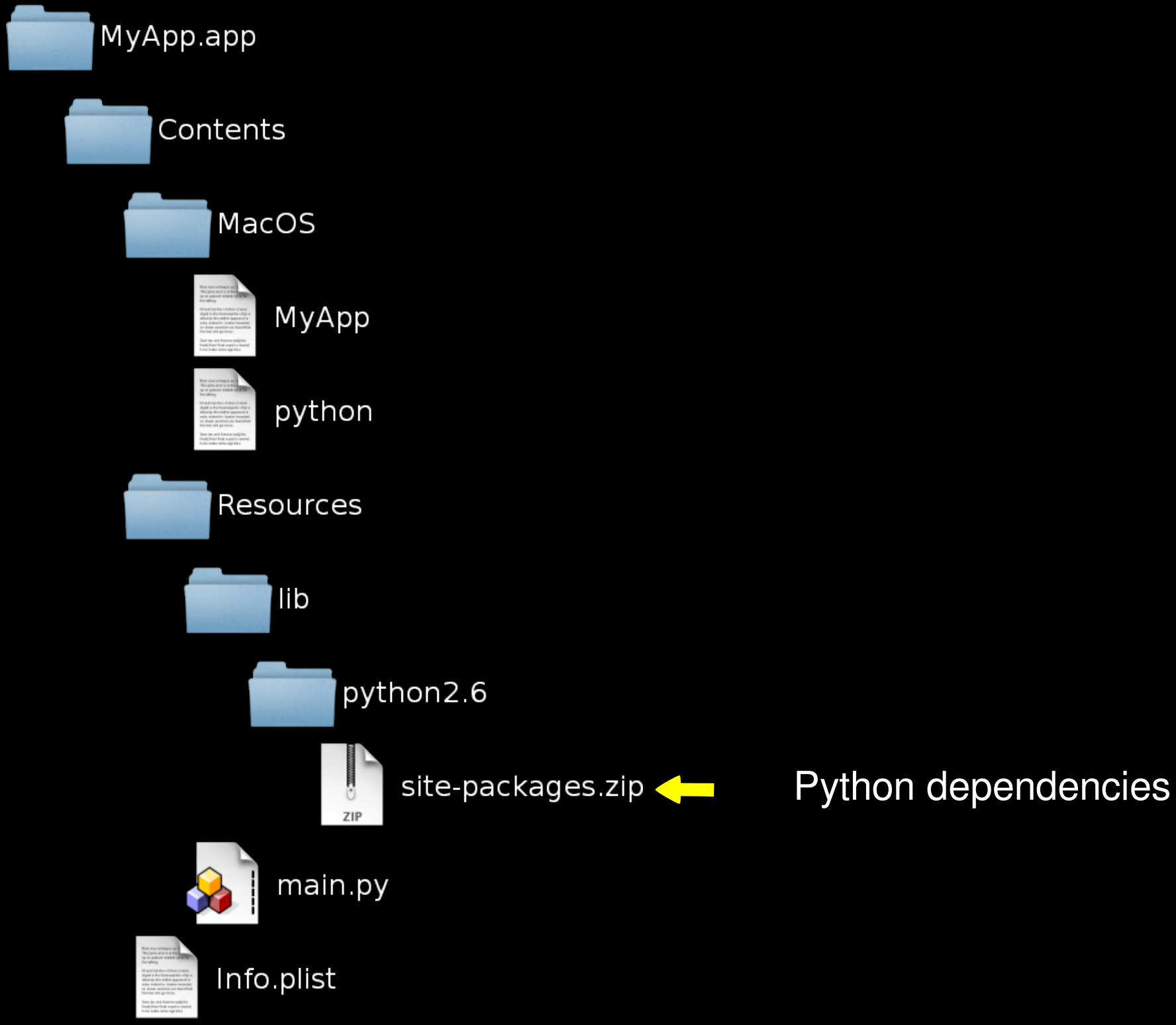
Then simply run the script like this:

```
python setup.py py2app
```


MyApp.app
The application bundle
Contents
MacOS
MyApp
python
Resources
lib
python2.6
ZIP
site-packages.zip
main.py
Info.plist


MyApp.app
Contents
MacOS
MyApp
python
Resources
lib
python2.6
ZIP
site-packages.zip
main.py
Info.plist
Property list file


MyApp.app
Contents
MacOS
MyApp
python
Resources
lib
python2.6
ZIP
site-packages.zip
main.py
Info.plist
Main script


MyApp.app
Contents
MacOS
MyApp
python
Resources
lib
python2.6
ZIP
site-packages.zip
Python dependencies
main.py
Info.plist

Customizing Application Properties

```
# Associating html and htm files with your application
Plist = {
    'CFBundleDocumentTypes': [{
        'CFBundleTypeExtensions': ['html','htm'],
        'CFBundleTypeName': 'HTML Document',
        'CFBundleTypeRole': 'Viewer',
        'CFBundleTypeIconFile': 'Icon.icns',
    }]
}
setup(
    app=['MyApp.py'],
    options={
        'py2app': {'plist': Plist}
    },
)
```

Common Issues with Frozen Apps

Dynamic Imports

Problem: `__import__`

Dynamic Imports

Solution: includes

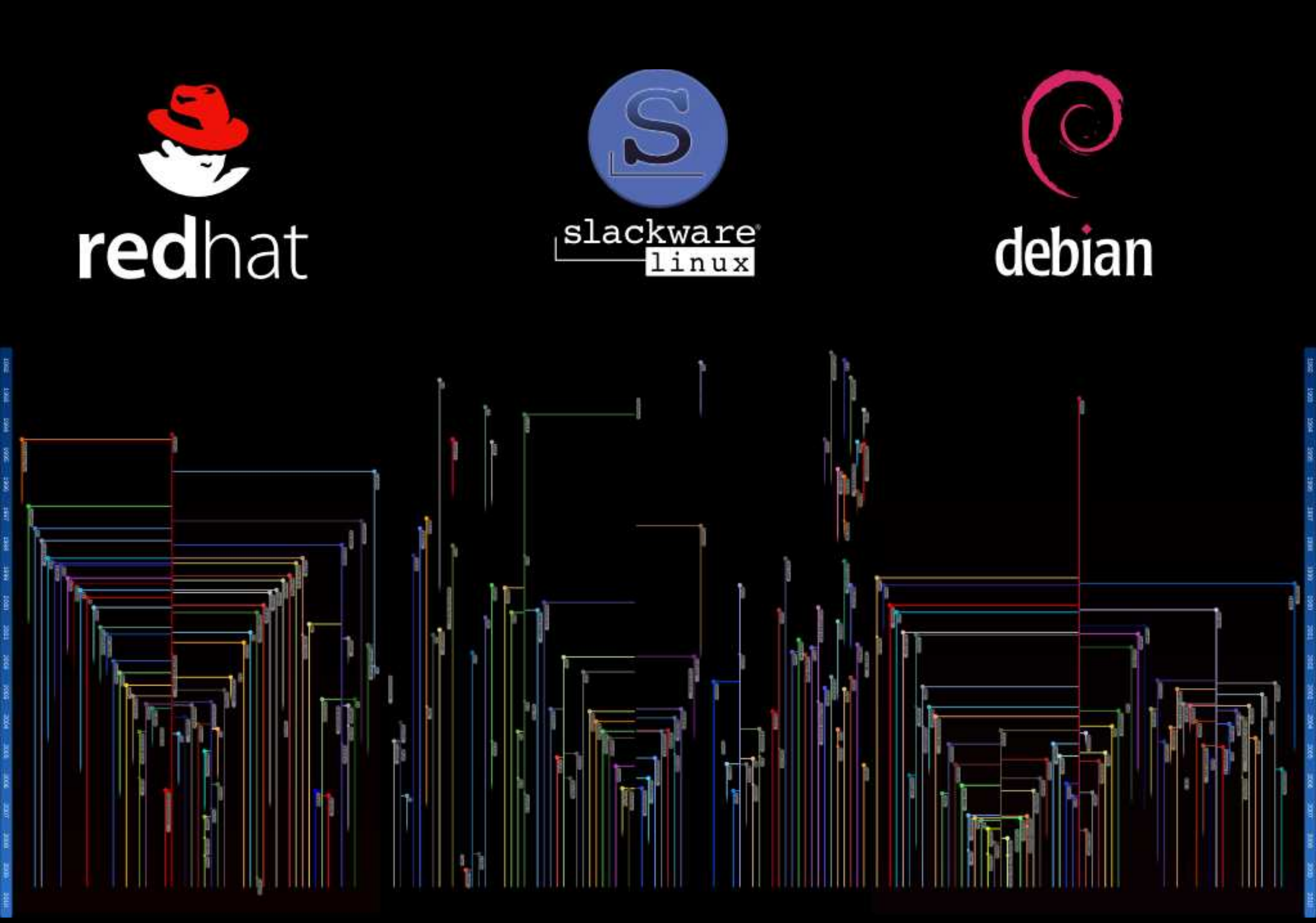
```
setup(
    app=['MyApp.py'],
    options={
        'py2app': {'includes': ['foo', 'bar']}
    },
)
```

Locating Data Files

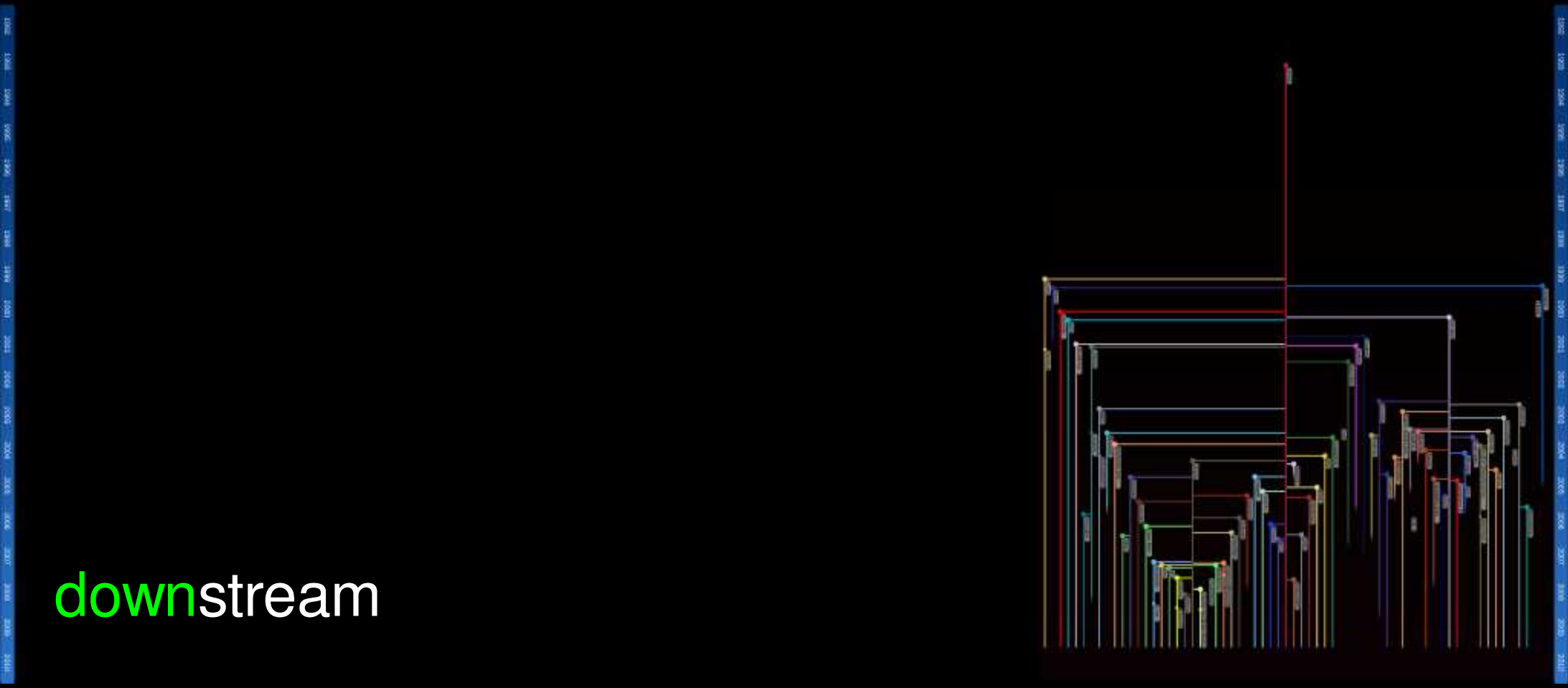
Problem: Requiring data files inside Python packages

Locating Data Files

Solution: Separate data files and source code


redhat
slackware linux
debian

upstream



downstream

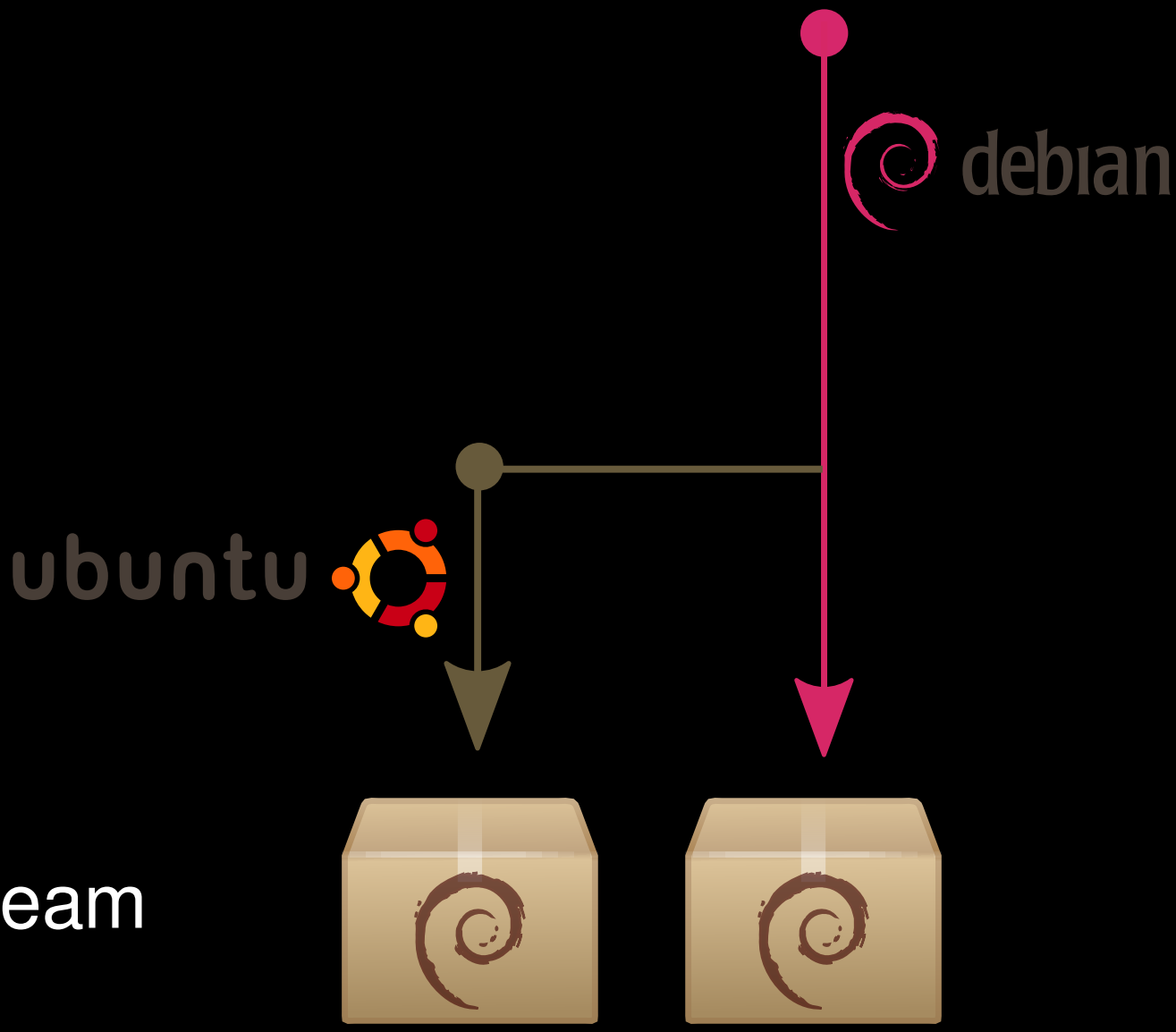
upstream

debian

downstream

ubuntu

upstream



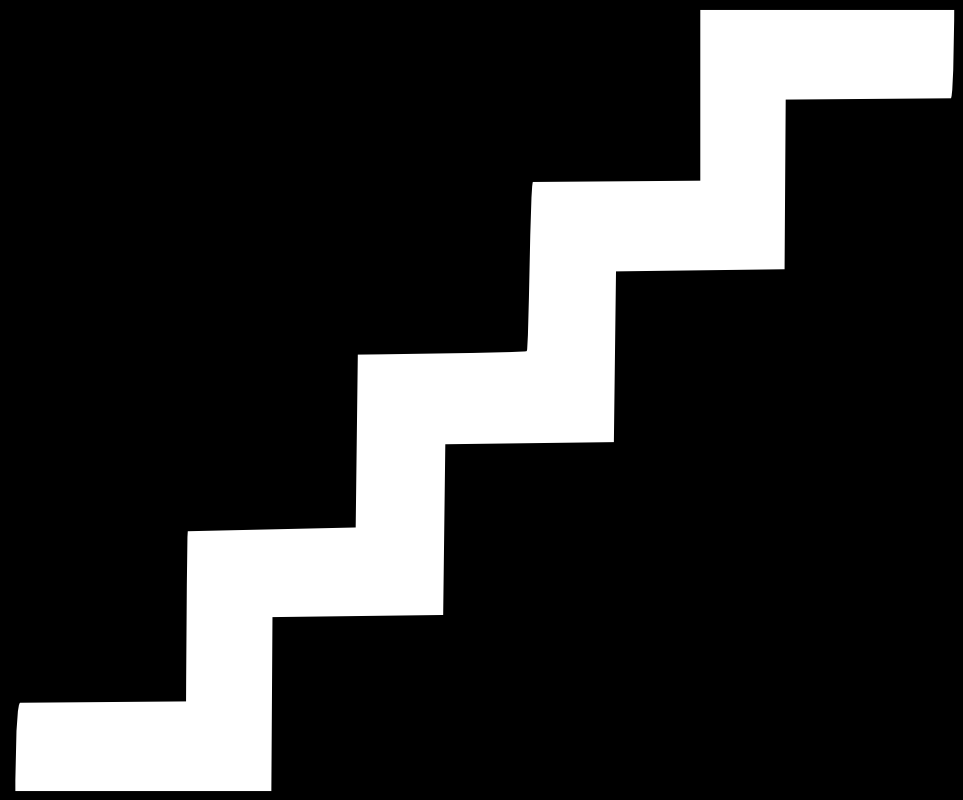
downstream

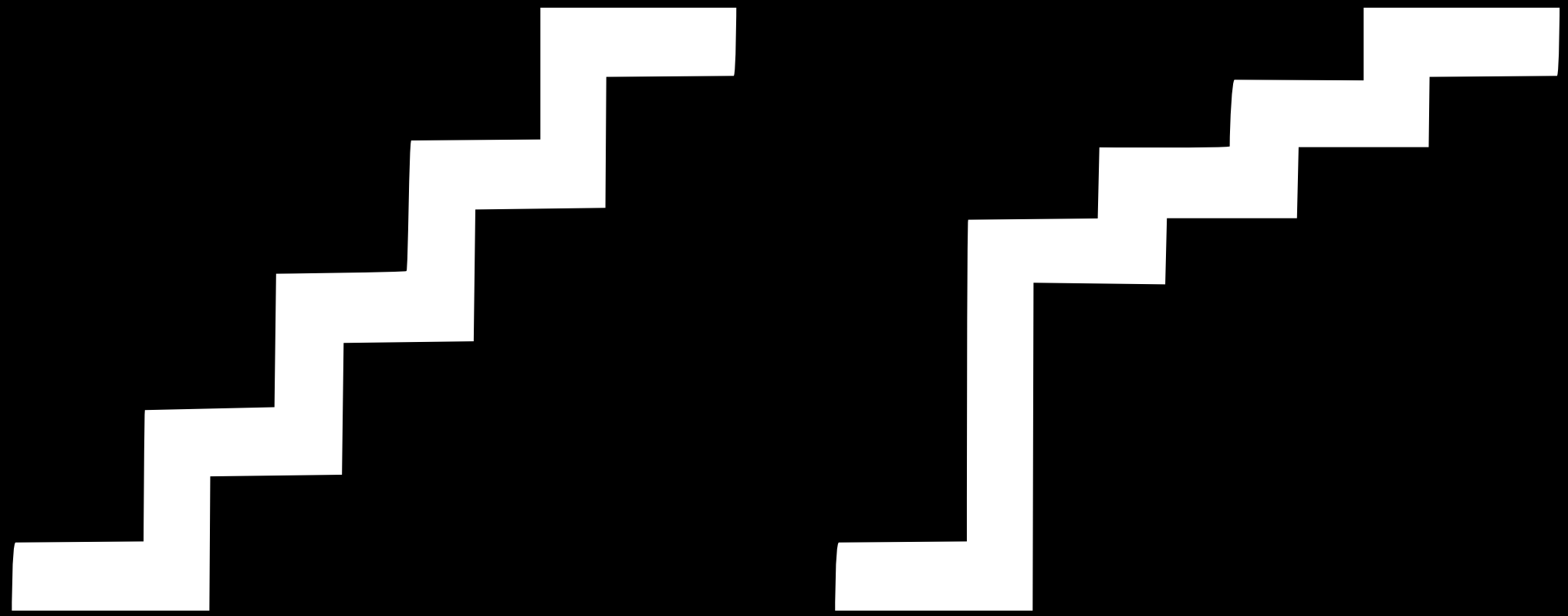

Marlboro


Marlboro


ic Health bill is enclosed.
Marlboro


Marlboro
KidsTik
Hand Sanitizer
Cobra Lounge






RFP

PAPT

p9m4/	View Log	Feed
pdfposter/	View Log	Feed
pdfshuffler/	View Log	Feed
phatch/	View Log	Feed
tags/	View Log	Feed
trunk/	View Log	Feed
debian/	View Log	Feed
changelog	View Log	Feed
compat	View Log	Feed
control	View Log	Feed
copyright	View Log	Feed
phatch-cli.links	View Log	Feed
phatch-cli.manpages	View Log	Feed
phatch-doc.dirs	View Log	Feed
phatch.dirs	View Log	Feed
phatch.lintian	View Log	Feed
phatch.menu	View Log	Feed
rules	View Log	Feed
watch	View Log	Feed
phenny/	View Log	Feed
photo-uploader/	View Log	Feed

python-stdeb

```
# available in Ubuntu Lucid
$ sudo apt-get install python-stdeb

# install from Python Package Index
$ pypi-install mypackage
```

python-stdeb

```
# make deb binary package in one step
$ python setup.py \
> --command-packages=stdeb.command bdist_deb
```

=

```
# 1) make dsc source package
$ py2dsc mypackage .tar.gz
```

+

```
# 2) build deb binary from dsc source
$ cd deb_dist/reindent-0.1.0/
$ dpkg-buildpackage -rfakeroot -uc -us
```


PPA

Raw Source Package (.dsc)

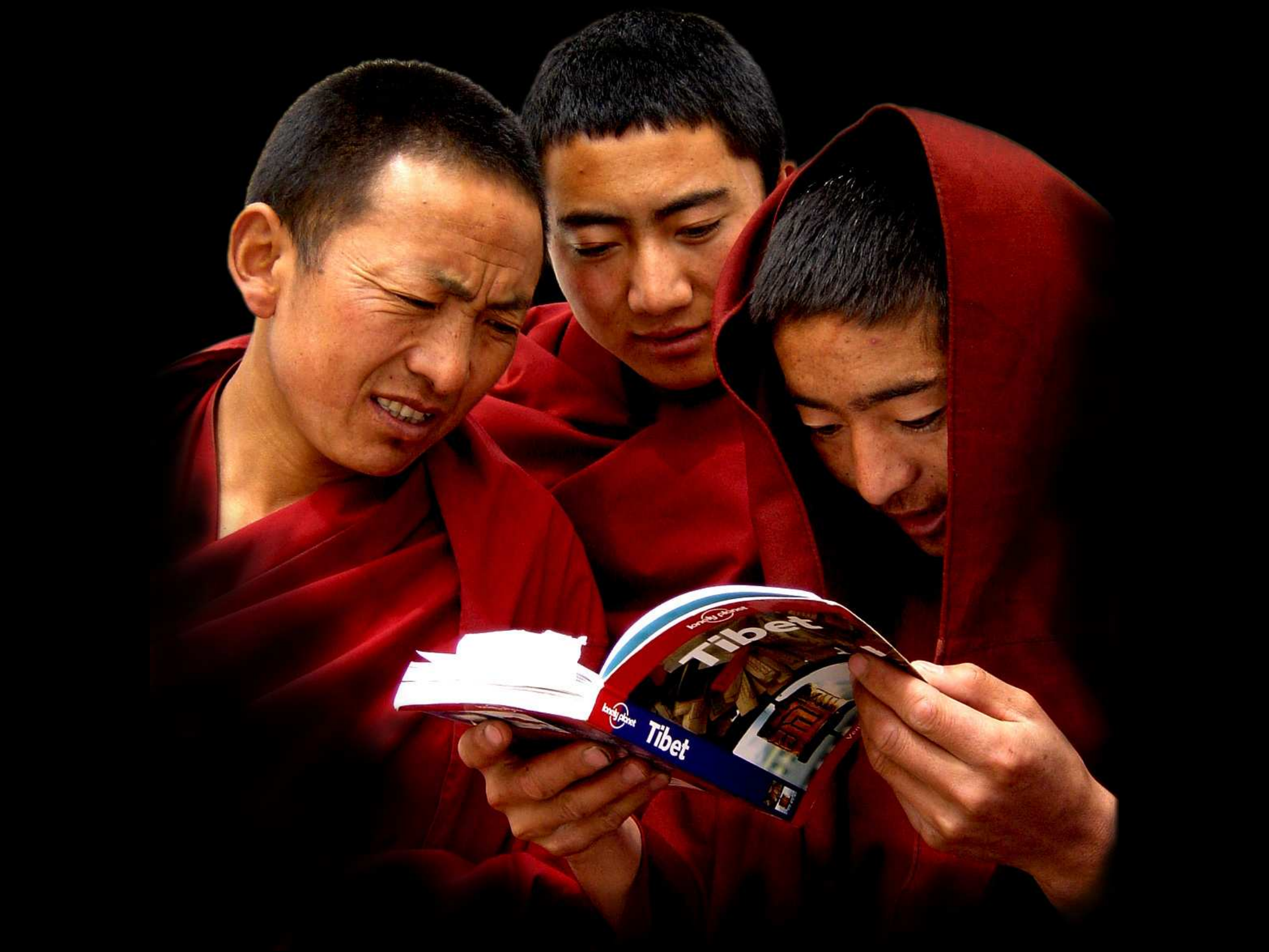
```
$ py2dsc source.tar.gz
```

Upload

```
$ dput ppa:stani/ppa <source.changes>
```

Binary Packages (.deb)

```
$ sudo add-apt-repository ppa:stani
```


Tibet
lonely planet
Tibet

Join the Phatch Sprint 22/2!