Writing Linux Desktop Apps

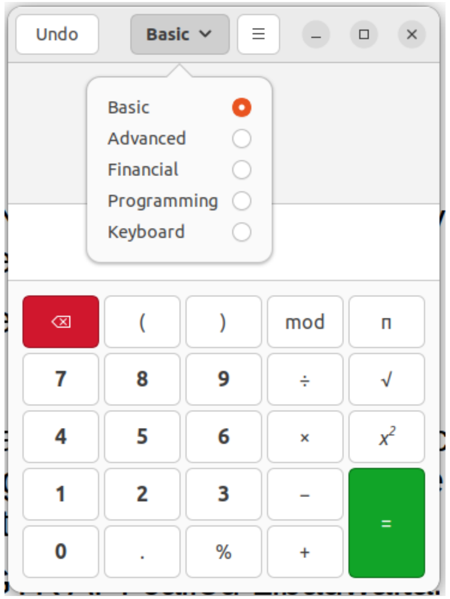
Presenter: Scott Davies, April 2023

~~Qt~~
~~WxWidgets~~
~~Tkinter~~
~~Remi~~

GNOME	desktop
GTK	graphics toolkit
PyGObject	wrapper library

GNOME

- **GNOME** stands for GNU Network Object Model Environment, is a free and open-source desktop environment.
- is available for Ubuntu, Debian, Arch, etc.
- Ubuntu 22.04 comes with 'GNOME 42'.
- The GLib data structures and utilities library, GObject object and type system and GTK widget toolkit comprise the central part of GNOME development platform.
- allows you to use a new GTK API called Libadwaita.



- **Adwaita** - is the design language of the GNOME desktop environment.
- the default theme and icon set of GNOME applications.
- painful if you're trying to get it working outside a GNOME environment.

Installation

PyGObject is a language-binding to the GTK+ widget toolkit.

It allows you to create modern, adaptive user interfaces that conform to GNOME's Human Interface Guidelines

If you're in Ubuntu 22.04 (with default GNOME desktop), you **don't have to install anything**!

Otherwise:

Installing the system provided PyGObject:

```
sudo apt install python3-gi python3-gi-cairo gir1.2-gtk-4.0
```

Installing from PyPI with pip:

```
sudo apt install libgirepository1.0-dev gcc libcairo2-dev pkg-
config python3-dev gir1.2-gtk-4.0;
```

- In a virtualenv:

```
pip3 install pycairo;
pip3 install PyGObject;
```

Some hints (to get going)

Install PyCharm as an IDE (to get autocomplete when typing).

```
sudo snap install pycharm-community –classic
```

Follow the tutorial by ‘Taiko’.

https://github.com/Taiko2k/GTK4PythonTutorial

Read the GTK4 project docs, especially about widgets.

https://docs.gtk.org/gtk4/class.Widget.html

Note about old GTK3

It still just works!

Show old Regex tester demo app.

```
cd /home/scott/ws/py/pygo/pygo_test1;
python3 py_regex_tester/regex_tester1.py
```

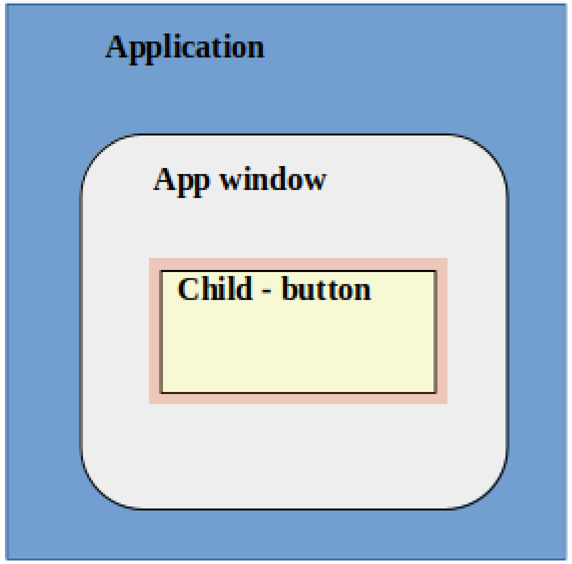
Hello world app (a)

Procedural style.

Create an Application.

- Has an application window.
- Set a child button in the app window.
- Present the app window.

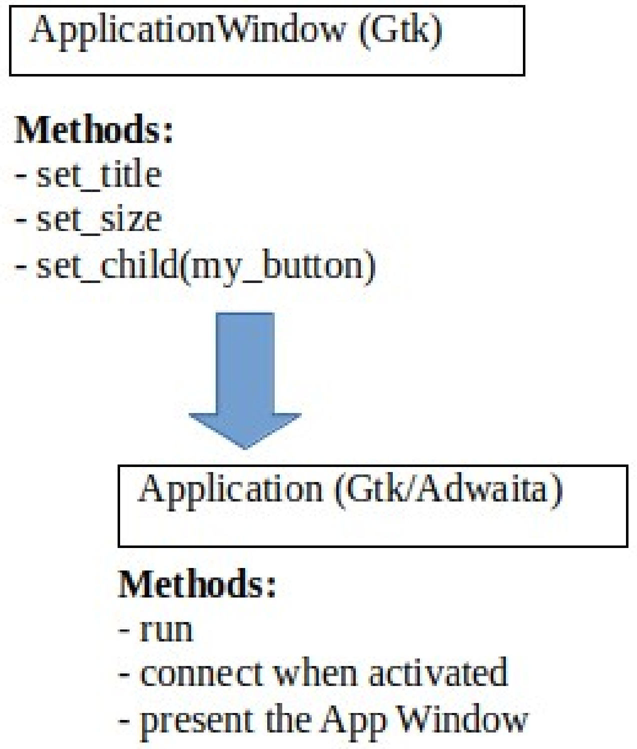
Show demo:
‘a_hello_world_v1_procedural.py



Hello world app (b)

Object oriented style.

- Use **classes** for ApplicationWindow and Application.
- The ApplicationWindow is a property on the Application.
- Call the Application run method,
- Which calls the ApplicationWindow present method.

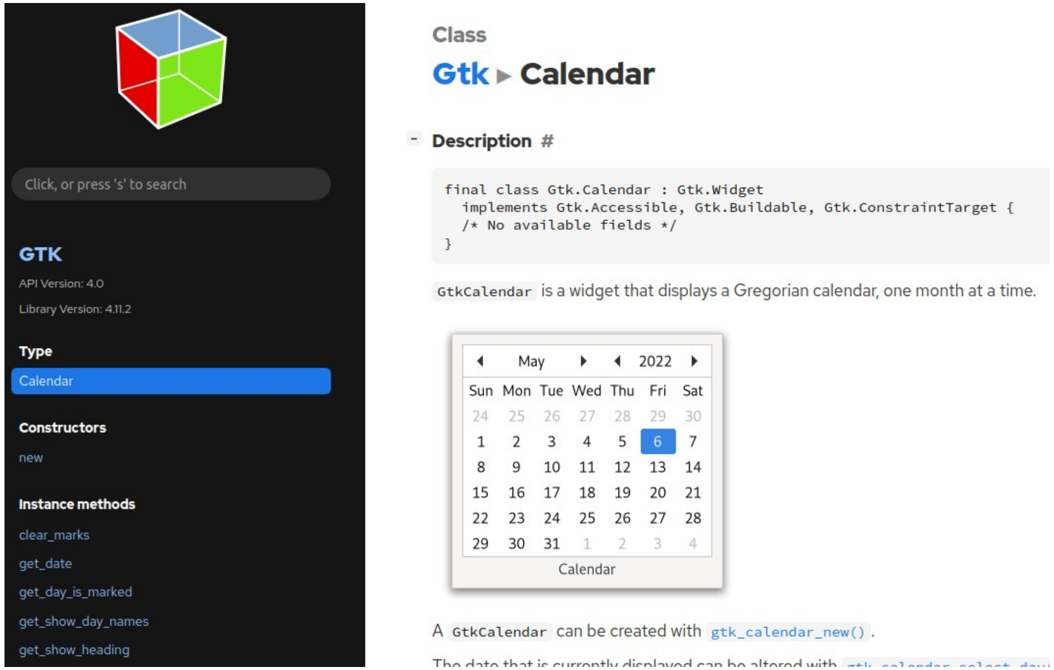


Boxes for layout & a theme (c)

- A GTK **Box widget** is a container allowing items to be placed within in, e.g. for layout.
- Can act as a row or a column.
- Using Application ‘set_color_scheme’ method with a predefined **theme**: ColorScheme.FORCE_DARK.

Show me the Widgets! (d)

- Box
- Label
- Entry
- CheckButton (radio buttons)
- Switch
- Scale
- FileChooserNative
- HeaderBar
- Menu (Gio)
- PopoverMenu
- AboutDialog



Class

Gtk ▸ Calendar

Description #

```
final class Gtk.Calendar : Gtk.Widget
  implements Gtk.Accessible, Gtk.Buildable, Gtk.ConstraintTarget {
  /* No available fields */
}
```

`GtkCalendar` is a widget that displays a Gregorian calendar, one month at a time.

A `GtkCalendar` can be created with `gtk_calendar_new()`.

The date that is currently displayed can be altered with

Lovely CSS (still d.)

You can use Cascading Style Sheets to style app windows and widgets.

```css
.title {
    font-size: 25px;
    font-weight: bold;
}

.bg-green {
    background-color:  #2ecc71;
    font-size: 120%;
}

.button-strong {
    background-color: #aed6f1;
}
```

Demo usage (e)

- Redis is an in-memory storage database.
- I need to (a) set some common config file values and (b) connect and get some keys from the database.
- I hate having to put customised lines like this in the config file repeatedly:

```
port 0
tls-auth-clients no
tls-cert-file /etc/redis/mydomain1.crt
tls-key-file /etc/redis/mydomain1.key
tls-ca-cert-file /etc/redis/mylocalauthority-root.pem
tls-port 6379
```

- I don't like this terrible long command line for connecting to redis-cli with TLS:

```
redis-cli -h mydomain1.org.local --tls --cert /etc/redis/mydomain1.crt --key /etc/redis/mydomain1.key --
cacert /etc/redis/mylocalauthority-root.pem
```

Skipped: RAD tools & layout

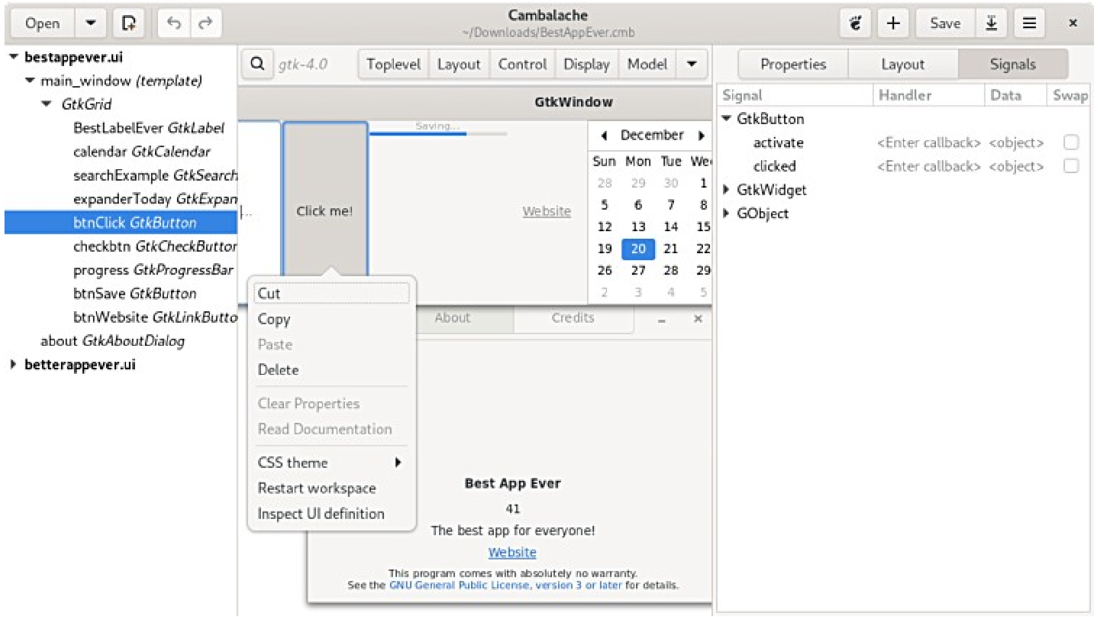
There are **Rapid Application Design tools** for GTK:

- Glade
- Cambalache

  https://gitlab.gnome.org/jpu/cambalache

There are different **layout possibilities** for PyGObject:

- Gtk.CenterBox
- Gtk.HeaderBar
- Gtk.Grid
- Gtk.ListBox
- Gtk.FlowBox
- Gtk.Stack
- Gtk.Notebook



Best App Ever

41

The best app for everyone!

Website

This program comes with absolutely no warranty.
See the GNU General Public License, version 3 or later for details.