This self-deprecating tutorial will hopefully help you code through a working wxPython GUI, including everything the Spring 2013 DVS team learned in one day.

Note: This assumes you have a basic understanding of object-oriented programming and Python syntax, neither of which I actually possess.

To start with, watch the tutorials by **thenewboston**:

Tutorial 1 – Building a Frame: <http://www.youtube.com/watch?v=RHvhfjVpSdE>.

2 – Creating Buttons: <http://www.youtube.com/watch?v=cp1ZeMisTNo>

7 – Static Text: <http://www.youtube.com/watch?v=qYVf09MVY14>

11 – Spinners: <http://www.youtube.com/watch?v=I6j-KiQeIYY>

These tutorials cover a few of the relevant basics that should help you get started on the DVS GUI. If you didn’t have time to watch the tutorials, then you can read my copy pasta of the important parts I picked up.

1. Setting up the file
   1. Import wx
   2. Create a class that takes in wx.Frame as an argument
      1. Define a method \_\_init\_\_, that takes 3 args: self, parent, id
   3. Create a main method that calls app and frame:

app=wx.PySimpleApp()

frame=test(parent=None, id= -1)

frame.Show()

app.MainLoop()

1. Creating a Frame
   1. Wx.Frame.\_\_init\_\_(self, parent, id, ‘Frame Title’, size=( xx, yy))

panel=ex.Panel(self)

1. Creating Buttons!
   1. Create an instance of wxButton
   2. Bind the event EVT\_BUTTON to the button, and declare and define the method you want to call on this event.

NOTE: It is important to close the frame after you have performed an action, otherwise the frames will build up and clutter the screen.

1. /\* Under Construction \*/

For more help, look over the Python documentation for [wxPython](http://wxpython.org/docs/api/). Fun fact: wxPython is actually a wrapper over the C++ GUI API called wxWidgets! If you know how to use that, then you should know more about wxPython than I do.

<http://web.mikitebeka.com/static/quick.pdf> //possible help for streamlining?

--Written by Andrew Fong on May 11, 2013