

```
{:title "Python - GTK 3.0" :layout :post :date "2017-1-08" :tags ["Python" "Code"
"Guide"]}
```

If you're looking to make GUI's using python. You should take a look at GTK 3.0 which is a nice module for building GUIs. [GTK 3.0](#) After installing it they have a couple of example code to get started with. Put the following in a file called gktut.py and go ahead and run the code below.

```
import gi
gi.require_version('Gtk', '3.0')
from gi.repository import Gtk

class MyWindow(Gtk.Window):

    def __init__(self):
        Gtk.Window.__init__(self, title="Hello World")

        self.button = Gtk.Button(label="Click Here")
        self.button.connect("clicked", self.on_button_clicked)
        self.add(self.button)

    def on_button_clicked(self, widget):
        print("Hello World")

win = MyWindow()
win.connect("delete-event", Gtk.main_quit)
win.show_all()
Gtk.main()
```

I'll go through some of the code above. So the first line is the basic import you have probably seen it just imports gi which allows you to import GTK from the gi.repository specifically you have to require Gtk 3.0.

```
Gtk.Window.__init__(self, title="Hello World")
```

This is the class constructor to set the window title.

```
self.button = Gtk.Button(label="Click Here")
```

So in this code you're just inheriting from the Gtk.Window class and initializing some basic things on the 8th line. Also then setting up a button which you do by calling Gtk.Button().

```
self.button.connect("clicked", self.on_button_clicked)
self.add(self.button)
```

Then you are connecting the button to a function called `on_button_clicked` which you define within the class.

```
win = MyWindow()
win.connect("delete-event", Gtk.main_quit)
win.show_all()
Gtk.main()
```

The first line is setting `win` as an object of the class `MyWindow()`. The second line allows you to close the window and without it you won't be able to close it. The second line shows the window and finally the `Gtk.main()` creates the main loop. So we can easily make a menu launcher just by knowing how to make buttons, a grid, and running commands in python using `call`. So let's start with the example code that the documentation gives us to start with and make our own menu launcher. First we'll start off by importing `call` from `subprocess`.

```
from subprocess import call
```

Now using `call` you can execute commands in python. So if we wanted to open firefox we could do something like the following and change `on_button_clicked`.

```
def on_button_clicked(self, widget):
    call(["firefox"])
```

Now you can launch firefox by pressing the button and you change the label of the button to 'Firefox'.

```
self.button = Gtk.Button(label="Firefox")
```

Now the button will be labeled as `firefox` and launches `firefox`. You could essentially make more of these buttons and make a launcher if you wanted to but I'll leave that up to you. So your final code for `gkttut.py` should be something like this...

```
import gi
gi.require_version('Gtk', '3.0')
from gi.repository import Gtk
from subprocess import call

class MyWindow(Gtk.Window):

    def __init__(self):
        Gtk.Window.__init__(self, title="Hello World")
```

```
        self.button = Gtk.Button(label="Firefox")
        self.button.connect("clicked", self.on_button_clicked)
        self.add(self.button)

    def on_button_clicked(self, widget):
        call(["firefox"])

win = MyWindow()
win.connect("delete-event", Gtk.main_quit)
win.show_all()
Gtk.main()
```

I just overviewed a small amount of what Gtk can do but you could look into comboboxs, entries, progressbars, and lists. Perhaps in a later blog post I'll go over a more complex project using GTK.