



# Starting with GTK

Julita Inca







# What is GTK?

GTK+, or the GIMP Toolkit, is a multi-platform toolkit for creating graphical user interfaces.

Source: <https://www.gtk.org>



GNU/Linux and Unix



Windows



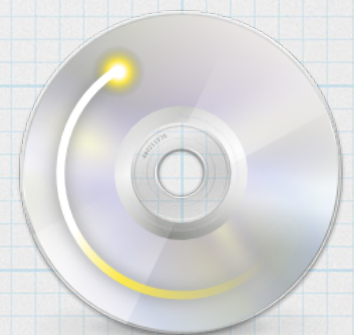
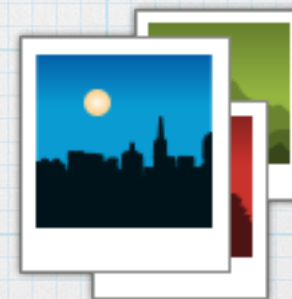
Mac OS X



Offering a complete set of widgets, GTK+ is suitable for projects ranging from small one-off tools to complete application suites.



INKSCAPE





# Stability

GTK+ is supported by a large community of developers and has core maintainers from companies such as











Source: <https://www.gtk.org/features.php>



# Language Bindings

Language Bindings (or 'wrappers') allow GTK+ to be used from other programming languages, in the style of those languages.

Language		GTK+ 2	GTK+ 3
C++ 		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Vala 		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Python 		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Javascript 		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Source: <https://www.gtk.org/language-bindings.php>

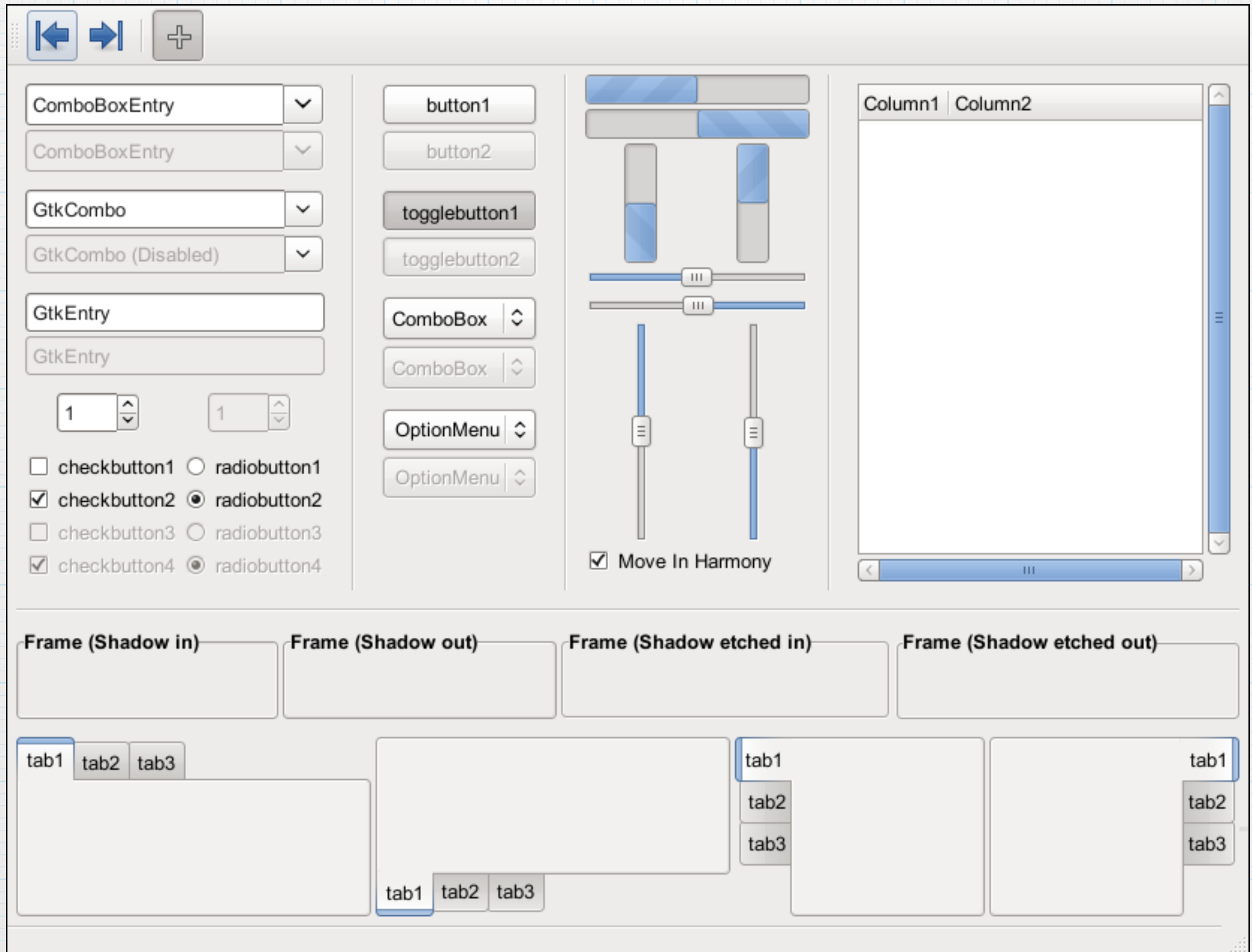


# Interfaces

GTK+ has a comprehensive collection of core widgets and interfaces for use in your application. GTK+ has features pertaining to mobile and embedded platform requirements:

Nokia 770 / N800 / N810 / N900, OLPC.







# Let's Build our firsts widgets!!!



Your GTK+ Application

GTK+

Pango

GDK

ATK

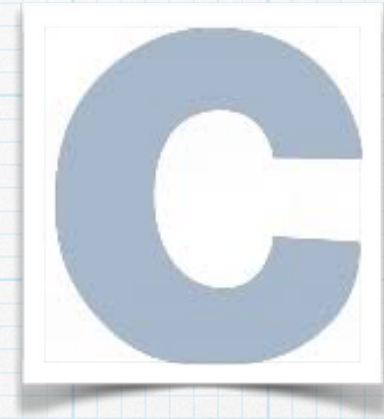
GIO

Cairo

GLib



# Working with Window



```
#include<gtk/gtk.h>
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    gtk_init(&argc,&argv);
```

```
    GtkWidget *window;
```

```
    window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
```

```
    g_signal_connect(window,"delete_event",G_CALLBACK(gtk_main_quit), NULL);
```

```
    gtk_widget_show(window);
```

```
    gtk_main();
```

```
    return 0;
```

```
}
```

```
sudo dnf install libgtk-3-dev
```

```
gcc `pkg-config --cflags gtk+-3.0` -o window01 window01.c `pkg-config --libs gtk+-3.0`
```



# Adding features to a Window

```
#include<gtk/gtk.h>
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    . . .
```

```
    gtk_window_set_default_size(window,300,300);
```

```
    gtk_window_set_title(GTK_WINDOW(window),"Linux Foundation");
```

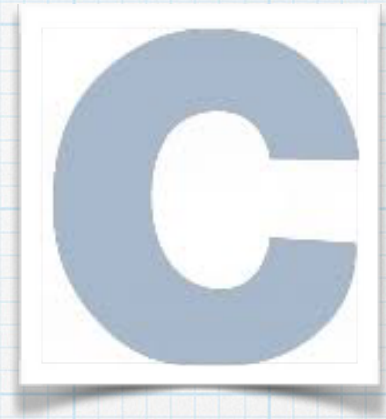
```
    gtk_widget_show(window);
```

```
    gtk_main();
```

```
    return 0;
```

```
}
```

```
gcc `pkg-config --cflags gtk+-3.0` -o window02 window02.c `pkg-config --libs gtk+-3.0`
```





# Adding a Label into the Window

```
#include<gtk/gtk.h>
```

```
int main(int argc, char* argv[])
```

```
{    . . .
```

```
    GtkWidget *window, *label;
```

```
    label = gtk_label_new("Welcome to APISTRAT");
```

```
    g_signal_connect(window,"delete_event",G_CALLBACK(gtk_main_quit), NULL);
```

```
    gtk_container_add(GTK_CONTAINER(window),label);
```

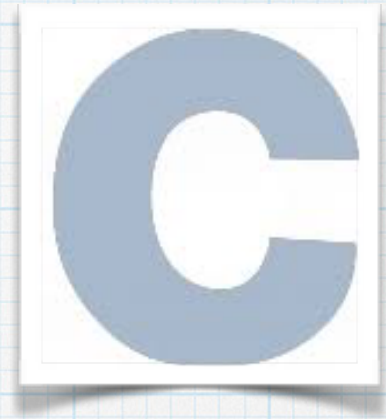
```
    gtk_widget_show_all(window);
```

```
    gtk_main();
```

```
    return 0;
```

```
}
```

```
gcc `pkg-config --cflags gtk+-3.0` -o window03 window03.c `pkg-config --libs gtk+-3.0`
```





# Adding a button to display a message

```
static void button_clicked(GtkWidget* widget, gpointer data)
{
    g_print("Welcome to APISTRAT");
}

. . .

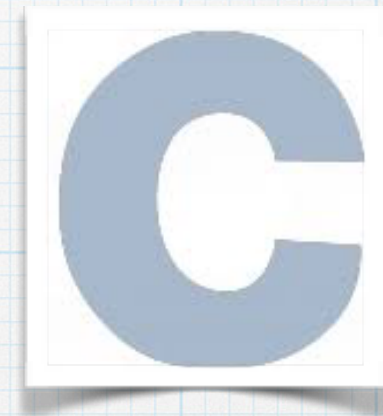
GtkWidget *window, *button;

button = gtk_button_new_with_label("Click me");

gtk_container_add(GTK_CONTAINER(window), button);

. . .

}
```

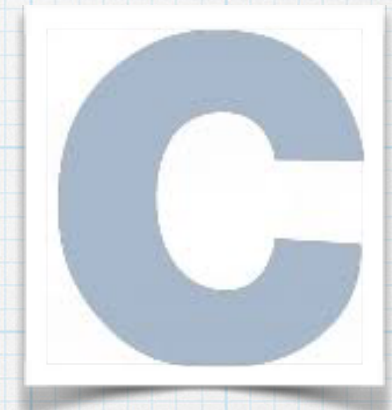


```
gcc `pkg-config --cflags gtk+-3.0` -o window04 window04.c `pkg-config --libs gtk+-3.0`
```



# Using a button to display a label (Box)

```
static void button_clicked(GtkWidget* widget, gpointer data){  
    gtk_label_set_text(GTK_LABEL(data), "Welcome to APISTRAT");  
}  
.  
.  
.  
  
GtkWidget *window, *button, *label, *box;  
  
label = gtk_label_new("                ");  
  
box = gtk_box_new(0, 0);  
  
gtk_box_pack_start(GTK_BOX(box), label, 0, 0, 0);  
  
gtk_box_pack_start(GTK_BOX(box), button, 1, 1, 0);  
  
gtk_container_add(GTK_CONTAINER(window), box);  
  
g_signal_connect(button, "clicked", G_CALLBACK(button_clicked), (gpointer)label);  
  
.  
.  
.  
}
```

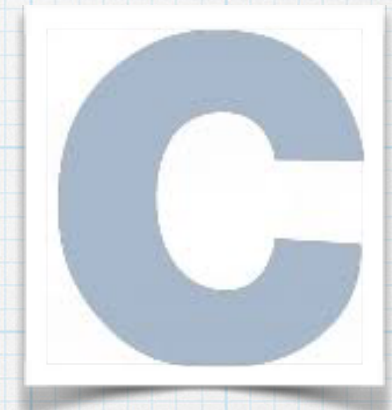


```
gcc `pkg-config --cflags gtk+-3.0` -o window05 window05.c `pkg-config --libs gtk+-3.0`
```



# Using a button to display a label (Grid)

```
static void button_clicked(GtkWidget* widget, gpointer data){  
    gtk_label_set_text(GTK_LABEL(data), "Welcome to APISTRAT");  
}  
.  
.  
.  
  
GtkWidget *window, *button, *label, *grid;  
  
grid = gtk_grid_new();  
  
label = gtk_label_new("                ");  
  
gtk_grid_attach(GTK_GRID(grid), label, 0, 0, 1, 1);  
  
gtk_grid_attach(GTK_GRID(grid), button, 3, 0, 1, 1);  
  
gtk_container_add(GTK_CONTAINER(window), grid);  
  
g_signal_connect(button, "clicked", G_CALLBACK(button_clicked), (gpointer)label);  
.  
.  
.  
}
```

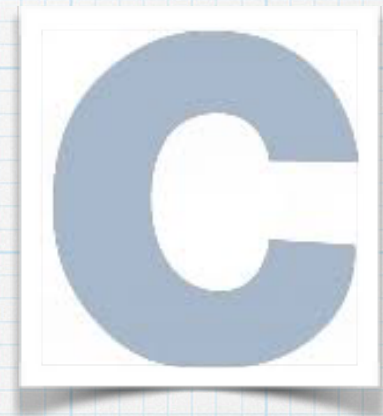


```
gcc `pkg-config --cflags gtk+-3.0` -o window06 window06.c `pkg-config --libs gtk+-3.0`
```



# Using an entry and a button into a Box

```
static void button_clicked(GtkWidget* widget, gpointer data){  
    g_print("%s\n", gtk_entry_get_text(GTK_ENTRY(data)));  
}  
.  
.  
.  
  
GtkWidget *window,*button,*label,*box;  
  
entry = gtk_entry_new();  
  
button = gtk_button_new_with_label("Click me");  
  
g_signal_connect(button,"clicked",G_CALLBACK(button_clicked), entry);  
  
box = gtk_box_new(0,0);  
  
gtk_box_pack_start(GTK_BOX(box),entry,0,0,0);  
  
gtk_box_pack_start(GTK_BOX(box),button,1,1,0);  
  
gtk_container_add(GTK_CONTAINER(window),box);  
  
.  
.  
.  
}
```

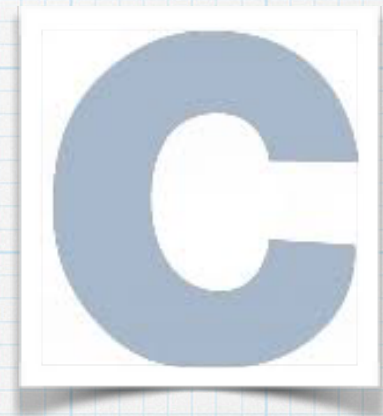


```
gcc `pkg-config --cflags gtk+-3.0` -o window07 window07.c `pkg-config --libs gtk+-3.0`
```



# Using an entry and a button into a Grid

```
static void button_clicked(GtkWidget* widget, gpointer data){  
    g_print("%s\n", gtk_entry_get_text(GTK_ENTRY(data)));  
}  
.  
.  
.  
  
GtkWidget *window,*button,*label,*box;  
  
entry = gtk_entry_new();  
  
button = gtk_button_new_with_label("Click me");  
  
g_signal_connect(button,"clicked",G_CALLBACK(button_clicked), entry);  
  
box = gtk_box_new(0,0);  
  
gtk_box_pack_start(GTK_BOX(box),entry,0,0,0);  
  
gtk_box_pack_start(GTK_BOX(box),button,1,1,0);  
  
gtk_container_add(GTK_CONTAINER(window),box);  
  
.  
.  
.  
}
```

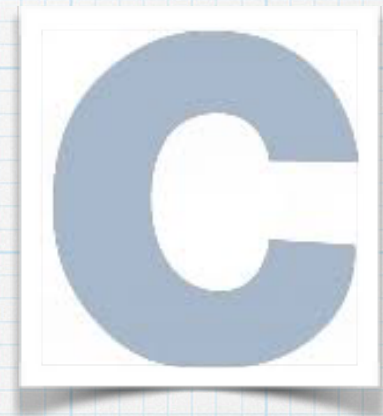


```
gcc `pkg-config --cflags gtk+-3.0` -o window08 window08.c `pkg-config --libs gtk+-3.0`
```



# Using an entry, a button and a label (Grid)

```
static void button_clicked(GtkWidget* widget, gpointer data){  
    g_print("%s\n", gtk_entry_get_text(GTK_ENTRY(data)));  
}  
.  
.  
.  
  
GtkWidget *window,*button,*label,*box;  
  
entry = gtk_entry_new();  
  
button = gtk_button_new_with_label("Click me");  
  
g_signal_connect(button,"clicked",G_CALLBACK(button_clicked), entry);  
  
box = gtk_box_new(0,0);  
  
gtk_box_pack_start(GTK_BOX(box),entry,0,0,0);  
  
gtk_box_pack_start(GTK_BOX(box),button,1,1,0);  
  
gtk_container_add(GTK_CONTAINER(window),box);  
  
.  
.  
.  
}
```



```
gcc `pkg-config --cflags gtk+-3.0` -o window09 window09.c `pkg-config --libs gtk+-3.0`
```



# Working with Window

```
sudo dnf install pygobject3-devel webkitgtk3-devel
```

```
from gi.repository import Gtk
```

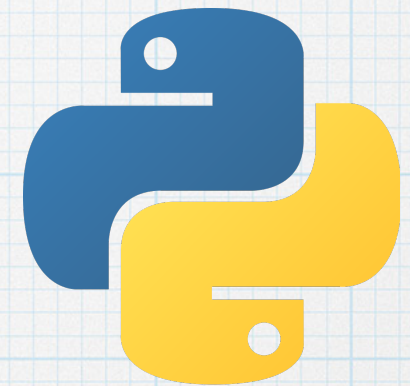
```
window = Gtk.Window()
```

```
window.connect("destroy", Gtk.main_quit())
```

```
window.show()
```

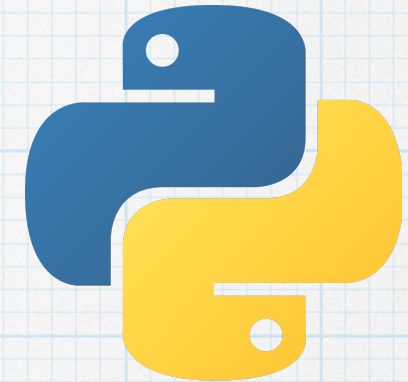
```
Gtk.main()
```

```
python window01.py
```





# Adding features to a Window



• • •

```
window.connect("destroy", Gtk.main_quit())  
window.set_default_size(400, 600)  
window.modify_bg(Gtk.StateType.NORMAL,  
Gdk.Color.from_floats(0.8,0.1,1.0))  
window.set_title("Linux Foundation")  
window_show()  
Gtk.main()
```

```
python window02.py
```



# Adding a Label into the Window

```
from gi.repository import Gtk
```

```
window = Gtk.Window()
```

```
label = Gtk.Label("Welcome to APISTRAT")
```

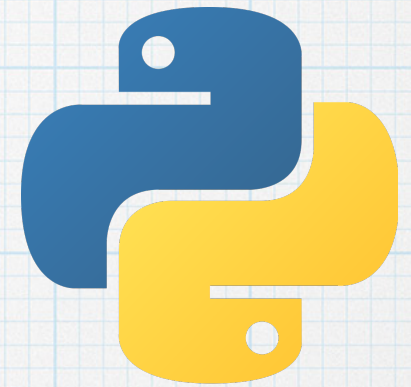
```
window.connect("destroy", Gtk.main_quit())
```

```
window.add(label)
```

```
window.show_all()
```

```
Gtk.main()
```

```
python window03.py
```





# Adding a button to display a message

```
def button_clicked(button):  
    print("Welcome to APISTRAT")
```

```
window = Gtk.Window()
```

```
button = Gtk.Button("Click me")
```

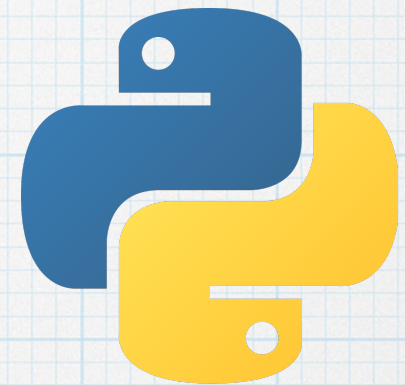
```
button.connect("clicked", button_clicked)
```

```
window.add(button)
```

```
window.show_all()
```

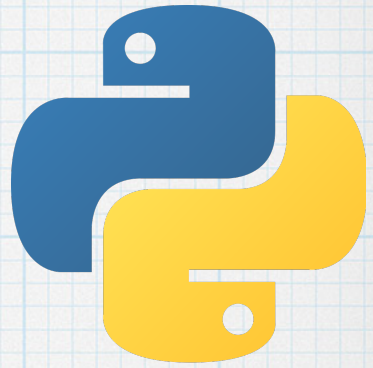
```
Gtk.main()
```

```
python window04.py
```





# Playing with a button to display a label



```
def button_clicked(button):  
    hbox.add(label)
```

```
    .  
    .  
    .  
label = Gtk.Label("Welcome to APISTRAT")
```

```
button.connect("clicked",button_clicked)
```

```
hbox = Gtk.Box(spacing=10)
```

```
hbox.pack_start(vbox_left, True, True, 0)
```

```
hbox.pack_start(vbox_right, True, True, 0)
```

```
window.add(box)
```

```
    .  
    .  
    .
```

```
python window05.py
```



jınca@gnome.org

@yulwitter