



# DESARROLLANDO APLICACIONES CON PYGTK

**Por: Fabio Esteban Durán Verdugo.**  
**Estudiante Ingeniería en Informática**  
**Universidad Autónoma de Chile – Sede Talca**

[fabio@gnome.cl](mailto:fabio@gnome.cl)  
<http://www.gnome.cl>





*¿Que Veremos?... ojalá sea bueno...*

Introducción.

Reseñas.

Datos.

Algunos Ejemplos Prácticos.





## *INTRODUCCION...*

Según yo...

Observo que...



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





*¿Que es PYGTK+?*

**PYGTK = Python + GTK**



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





*Python...*

Lenguaje Interpretado de Alto Nivel  
Es Orientado a Objetos.  
Creado por Guido Van Rossum en 1990  
Multi-plataforma

Estructura del Código:

Fin de línea marca fin de sentencia  
La Sangría expresa la estructura de un bloque  
Es sensible a la indentación.





*Python...*

```
palabra = "Hola Mundo!!!"
```

```
if palabra == "Hola Mundo!!!":
```

```
    print 'estas en lo correcto'
```

```
else:
```

```
    print 'que bonito pero estas mal'
```

```
>> estas en lo correcto
```



---

fabio@gnome.cl





GNOME™

*Python... ¿Como se hace?*

# SENTENCIAS DE FLUJO DE CONTROL



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

*Python...*

**if <CONDICIÓN>:**

**(4 espacios) <SENTENCIA>**

**elif <CONDICIÓN>:**

**(4 espacios) <SENTENCIA>**

**else: <CONDICIÓN>:**

**(4 espacios) <SENTENCIA>**



[fabio@gnome.cl](mailto:fabio@gnome.cl)







GNOME™

Python...

EJEMPLO 1

```
lista = [1,2,3]
if lista == [1, 2, 3]:
    (4 espacios) print 'si'
>> si
```

EJEMPLO 2

```
lista1 = [1,2,3]
lista2 = [3,4,5]

if lista1 > lista2:
    (4 espacios) print 'Mayor'
elif lista1 < lista2:
    (4 espacios) print 'Menor'
else:
    (4 espacios) print 'Igual'
>> Mayor
```



python  
powered

[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

*Python...*

**for <VARIABLE> in <SECUENCIA>:  
(4 espacios) <SENTENCIA>**



python  
powered

---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

Python...

EJEMPLO 1

```
for i in range (1, 3):
```

```
    print i
```

```
>> 1
```

```
>> 2
```

```
>> 3
```

EJEMPLO 2

```
coordenada = [(1, 1), (1, 2), (1, 3)]
```

```
for x,y in coordenada:
```

```
    print x,y
```

```
>> 1 1
```

```
>> 1 2
```

```
>> 1 3
```



python  
powered

[fabio@gnome.cl](mailto:fabio@gnome.cl)





*Python...*

ejemplo 3

```
diccionario = {1:'a', 3:'c',2:'b' }
```

```
letras = diccionario.keys()
```

```
letras.sort()
```

```
for i in letras:
```

```
    print diccionario[i]
```

```
>> a
```

```
>> b
```

```
>> c
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)





*Python...*

**while <CONDICIÓN>:  
(4 espacios) <SENTENCIA>**



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

Python...

ejemplo 1

```
i = 0
```

```
while i != 3:
```

```
    (4 espacios) print i
```

```
        i = i+1
```

```
>> 0
```

```
>> 1
```

```
>> 2
```

ejemplo 2

```
x = 3
```

```
while True:
```

```
    (4 espacios) x = x-1
```

```
    if x == 3:
```

```
        print x
```

```
    if x == 0
```

```
        break
```

```
>> 3
```



python  
powered

[fabio@gnome.cl](mailto:fabio@gnome.cl)





*Python...*

# DEFINICIÓN DE FUNCIONES



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

*Python...*

**Se realiza con la palabra “ *def* ”**

**No se distinguen entre procedimientos y funciones**

**Todos los valores se traspasan por referencia**

**Un función puede devolver cualquier dato o estructura de datos.**

ejemplo 1

```
def holamundo():
```

```
    (4 espacios) print 'Hola Mundo'
```

```
holamundo():
```

```
>> Hola Mundo
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)







GNOME™

*Python...*

ejemplo 2

```
def suma(i):  
    return i + 2  
print suma (6)  
>> 8
```

ejemplo 4

```
def valores(x=0, y=0):  
    print x ' , ' y  
valores()  
>> 0 , 0  
valores (x= 1, y=2)  
>> 1 , 2
```

ejemplo 3

```
i = 100  
def veamos(i):  
    print i  
veamos(5)  
>> 5
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

*Python...*

# DEFINICIÓN DE CLASES



python  
powered

---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

*Python...*

**Una colección de métodos.**

**Cada método contiene como primer parámetro la palabra “*self*” que hace referencia a un objeto.**

ejemplo 1

```
class ejemplo():
    def __init__(self):
        print self.suma(6)
    def suma(self, i):
        return i + 2
if __name__ == '__main__':
    d = ejemplo()
>> 8
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)





*Python...*

ejemplo 2

```
class uno():
    def __init__(self):
        dos()
class dos ()
    def __init__(self):
        print self.suma(6)
    def suma(i):
        return i + 2
if __name__ == '__main__':
    x = uno()
>> 8
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)





*Python...*

# IMPORTANDO MODULOS



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





*Python...*

**from <MODULO> import <CLASE>**

**import <MODULO>**



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





*Python...*

ejemplos

```
# importando modulo de sistema  
import sys
```

```
# importando gtk+  
import gtk
```

```
# importando modulo connect desde pycopg  
from pycopg import connect
```



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





*GTK+...*

**GTK = the Gimp ToolKit**



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)







GNOME™

GTK+...

### 3 Librerías Principales:

**GLIB = Manejo de estructuras de datos para C. (Conjunto de Widgets)**

**ATK = Creada para crear interfaces para gente discapacitada o minusválidos**

**PANGO = Renderizado y diseño de texto, Internacionalización.**



[fabio@gnome.cl](mailto:fabio@gnome.cl)





*GTK+...*

## Algunos widgets:

**gtk.Window**

**gtk.Button**

**gtk.Label**

**gtk.TreeView**

**gtk.ListStore**

**etc... etc... etc...**



[fabio@gnome.cl](mailto:fabio@gnome.cl)





*GTK+...*

### Ejemplo PYGTK:

```
import gtk
class HolaMundo:
    def hello(self, widget, data=None):
        print "Hello World"
    def delete_event(self, widget, event, data=None):
        print "delete event occurred"
        return False
    def destroy(self, widget, data=None):
        print "destroy signal occurred"
        gtk.main_quit()
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)





GTK+...

```
def __init__(self):
    self.window = gtk.Window(gtk.WINDOW_TOPLEVEL)
    self.window.connect("delete_event", self.delete_event)
    self.window.connect("destroy", self.destroy)
    self.window.set_border_width(10)
    self.button = gtk.Button("Hello World")
    self.button.connect("clicked", self.hello, None)
    self.button.connect_object("clicked", gtk.Widget.destroy, self.window)
    self.window.add(self.button)
    self.button.show()
    self.window.show()

def main(self):
    gtk.main()

if __name__ == "__main__":
    hello = HolaMundo()
    hello.main()
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)



# Glade.

## Constructor de Interfaces

### Independencia entre código e interfaz





*GTK+...*

# GLADECONNECT...



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

GTK+...

**Librería o Clase Base para conectar Glade con Python.**

**Ventaja:**

**Independencia entre Código Python e Interfaz.**

**¿Como Utilizar la librería?**

```
from GladeConnect import GladeConnect
```

```
Class Ejemplo (GladeConnect)
```

```
def __init__(self):
```

```
    GladeConnect.__init__(self, "ruta_del_archivo .glade", "objeto_a_mostrar_")
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

GTK+...

## NOTA IMPORTANTE:

**“La librería GladeConnect debe estar en mi espacio de trabajo”**



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)







GNOME™

*GTK+...*

### ejemplo 1 GladeConnect

```
import sys
import gtk
from GladeConnect import GladeConnect

class Ejemplo(GladeConnect):
    def __init__(self):
        GladeConnect.__init__(self, "ejemplo.glade", "wnEjemplo")

if __name__ == '__main__':
    w = ejemplo()
    gtk.main()
```



[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

*GTK+...*

**Referencias:**

**DevHelp.**

**python:**

**aprenda a pensar como programador con Python.**

**Dive into Python.**

**pygtk:**

**PyGTK 2.0 Reference Manual**

**PyGTK 2.0 Tutorial**



python  
powered

---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





GNOME™

*GTK+...*

## Referencias:

**Guía de como hacer preguntas inteligente**

**[www.sindominio.net/ayuda/preguntas-inteligentes.html](http://www.sindominio.net/ayuda/preguntas-inteligentes.html)**

**Construcción de Interfaces.**

**HIG.**

**Guía de recomendaciones de como diseñar interfaces para GNOME.**

**<http://developer.gnome.org/projects/gup/hig/>**



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)





*GTK+...*

**¿PREGUNTAS?**

**manden sus SPAM**

**fabio@gnome.cl**



---

[fabio@gnome.cl](mailto:fabio@gnome.cl)

