

Introducción a Python

Parte 2

Mariano Reingart



Arte gráfico: Diana Batista

Indice

- Introducción a la librería estandar
- Persistencia y Bases de Datos
- Desarrollo de Aplicaciones Gráficas (GUI) simples
- Desarrollo de Aplicaciones Web simples
- Consejos para usuarios de VB / PHP
- Resumen características avanzadas

Fechas y horas

```
>>> import datetime
>>> hoy = datetime.date.today()
>>> print hoy
2009-02-26
>>> ahora = datetime.datetime.now()
>>> print ahora
2009-02-26 14:15:16.025000
>>> hoy + datetime.timedelta(days=4)
datetime.date(2009, 3, 2)
>>> cumple = datetime.date(2009, 07, 23)
>>> cumple - hoy
datetime.timedelta(147)
>>> cumple.strftime("%d de %B de %Y")
'23 de Julio de 2009'
```

Números decimales “punto-fijo”

```
>>> f = 1.1    # float o número de punto flotante binario
>>> f
1.100000000000000001
>>> f * 2
2.200000000000000002
>>> import decimal
>>> d = decimal.Decimal("1.1")
>>> d * 2
Decimal("2.2")
>>> decimal.getcontext().prec = 6      # ajusto la precisión
>>> decimal.Decimal(1) / decimal.Decimal(7)
Decimal("0.142857")
```

Sistema operativo: archivos y dirs.

```
>>> import os
>>> os.listdir("c:/")
['Archivos de programa', 'AUTOEXEC.BAT' ...]
>>> os.path.isdir("C:/AUTOEXEC.BAT")
False
>>> os.mkdir("C:/prueba")
>>> os.path.join("C:\\prueba", "mi-archivo.txt")
'C:\\prueba\\mi-archivo.txt'
>>> os.path.exists('C:\\prueba\\mi-archivo.txt')
False
>>> os.path.splitext('mi-archivo.txt')
('mi-archivo', '.txt')
```

Sistema operativo: procesos

```
>>> import os
>>> os.execv("C:\\PROGRAMA.EXE", ["A","B","C"])
>>> os.system("notepad.exe")
>>> pid = os.spawnv(os.P_NOWAIT,"c:\\windows\\notepad.exe",[])
>>> os.kill(pid, 9)    # enviar señal, no funciona en windows
>>> os.waitpid(pid, 0) # esperar al proceso
(304, 0)
>>> from subprocess import Popen, PIPE
>>> p1 = Popen(["ipconfig"], stdout=PIPE)    # crear una "pipa"
>>> output = p1.communicate(input=None)
>>> output
('Configuración IP ... Adaptador Ethernet coLinux: ... Dirección IP:
192.168.0.2 ...', None)
```

Archivos

```
>>> archivo = open("C:/config.sys","r")
>>> for linea in archivo:
...     print linea
DEVICE=C:\WINDOWS\setver.exe
FILES = 255
BUFFERS = 8
>>> archivo.close()

>>> archivo = open("prueba.bin","wb")
>>> archivo.write("hola")
>>> archivo.flush() # opcional, envia los datos al disco ya
>>> archivo.close()
```

Planillas delimitadas por coma (CSV)

```
import csv
```

```
# abrir el archivo y procesar el archivo csv
```

```
csv_reader = csv.reader(open("archivo.csv"))
```

```
# recorrer las filas de la planilla
```

```
for fila in csv_reader:
```

```
    # leer las columnas y asignarlas a las variables
```

```
    celda1, celda2, celda3 ... = fila
```

```
    # autorizar la factura leída
```

```
    print celda1, celda2
```


Correo Electrónico

```
import smtplib
```

```
from email.mime.text import MIMEText
```

```
# creo el mail
```

```
msg = MIMEText("Envio factura ... CAE ...")
```

```
msg['Subject'] = 'Factura Electronica'
```

```
msg['From'] = "python@pyafipws.com.ar"
```

```
msg['To'] = "prueba@pyafipws.com.ar"
```

```
# envio el mail
```

```
smtp = smtplib.SMTP("smtp.nsis.com.ar")
```

```
smtp.sendmail(msg['From'], msg['To'], msg.as_string())
```

Generación de PDF

```
from pyfpdf import FPDF
```

```
pdf = FPDF()
```

```
pdf.AddPage()
```

```
pdf.SetFont(family='Arial',style='B',size=16)
```

```
pdf.Image('logo.png', x=20.0, y=17.0, w=58.0, h=13.0)
```

```
pdf.Interleaved2of5("1234", x=20.0, y=243.5, w=0.75)
```

```
pdf.Text(x=125.0, y=251.0, txt='Total:')
```

```
pdf.SetXY(x=150.0, y=246.0)
```

```
pdf.Cell(h=5.0, align='R', w=30.0, txt='$100.00')
```

```
pdf.Rect(x=15.0, y=15.0, w=170.0, h=245.0)
```

```
pdf.Output("factura.pdf","F")
```

Bases de datos: DbAPI

Python tiene una interfase unificada llamada DbApi (PEP249) y SQLite viene incluido desde la versión 2.5

```
>>> import sqlite3
>>> conn = sqlite3.connect('c:/ejemplo.db') # conectarse a la base
>>> cur = conn.cursor() # creo un cursor para ejecutar consultas
>>> cur.execute("CREATE TABLE prueba (id INTEGER PRIMARY KEY AUTOINCREMENT, nombre TEXT, edad INT)")
>>> cur.execute("INSERT INTO prueba (nombre, edad) VALUES (?,?)", ["Mariano", 30]) # agregar registros (parámetros separados)
>>> cur.execute("SELECT * FROM prueba WHERE edad>=?", [30])
>>> for fila in cur:
    print fila
(1, u'Mariano', 30)
```

Interfases Web: WSGI

WSGI es una interface simple y universal entre los servidores web y las aplicaciones web o frameworks (PEP 333), similar a la especificación Java Servlet o ASP/ASP.NET.

```
from wsgiref.simple_server import make_server
```

```
def hello(environ, start_response):
```

```
    start_response('200 OK', [('Content-type', 'text/plain')])
```

```
    return ['Hola PyAr!']
```

```
httpd = make_server("", 8000, hello).serve_forever()
```

Aplicaciones Web: Spyce (PSP)

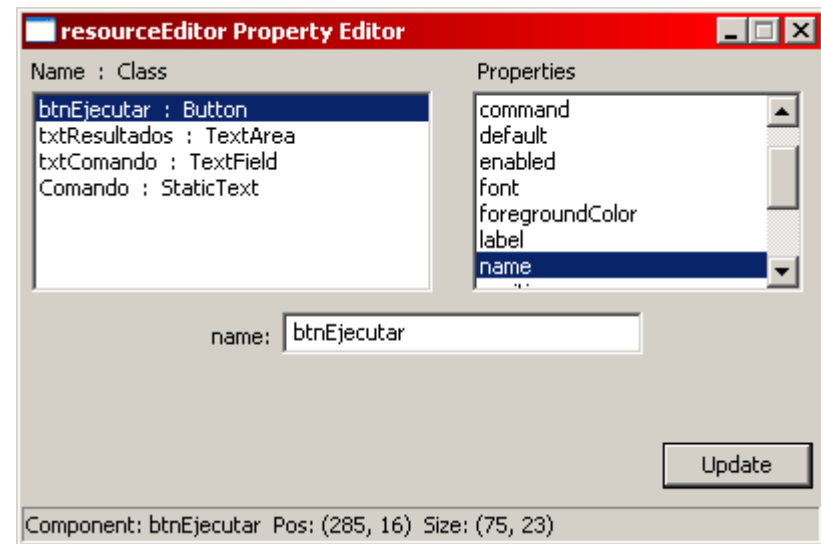
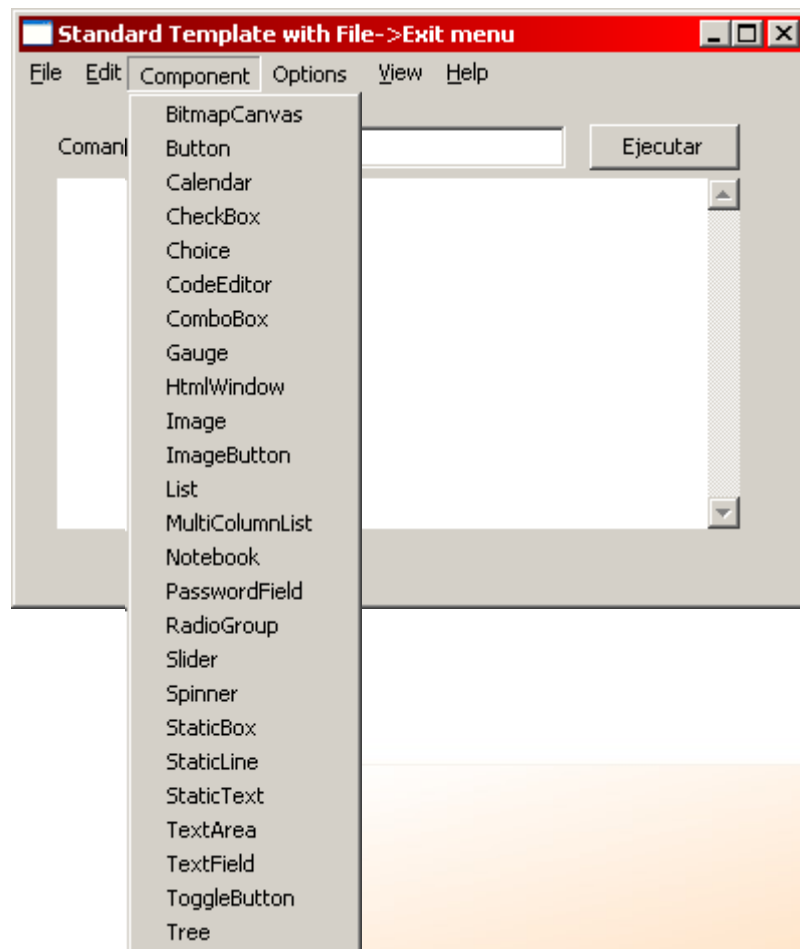
Spyce es un framework web simple que permite hacer páginas dinámicas de forma similar a PHP o Java Server Pages (por eso el nombre de Python Server Pages)

```
<html>
[[ import sys ]]
<head>
<title>Python [[ '='.join([str(v) for v in sys.version_info[0:3]]) ]]
</title></head>
<body><p>Bienvenido a Python:
[[ sys.version ]]>
</p></body></html>
```

Aplicaciones GUI: PythonCard

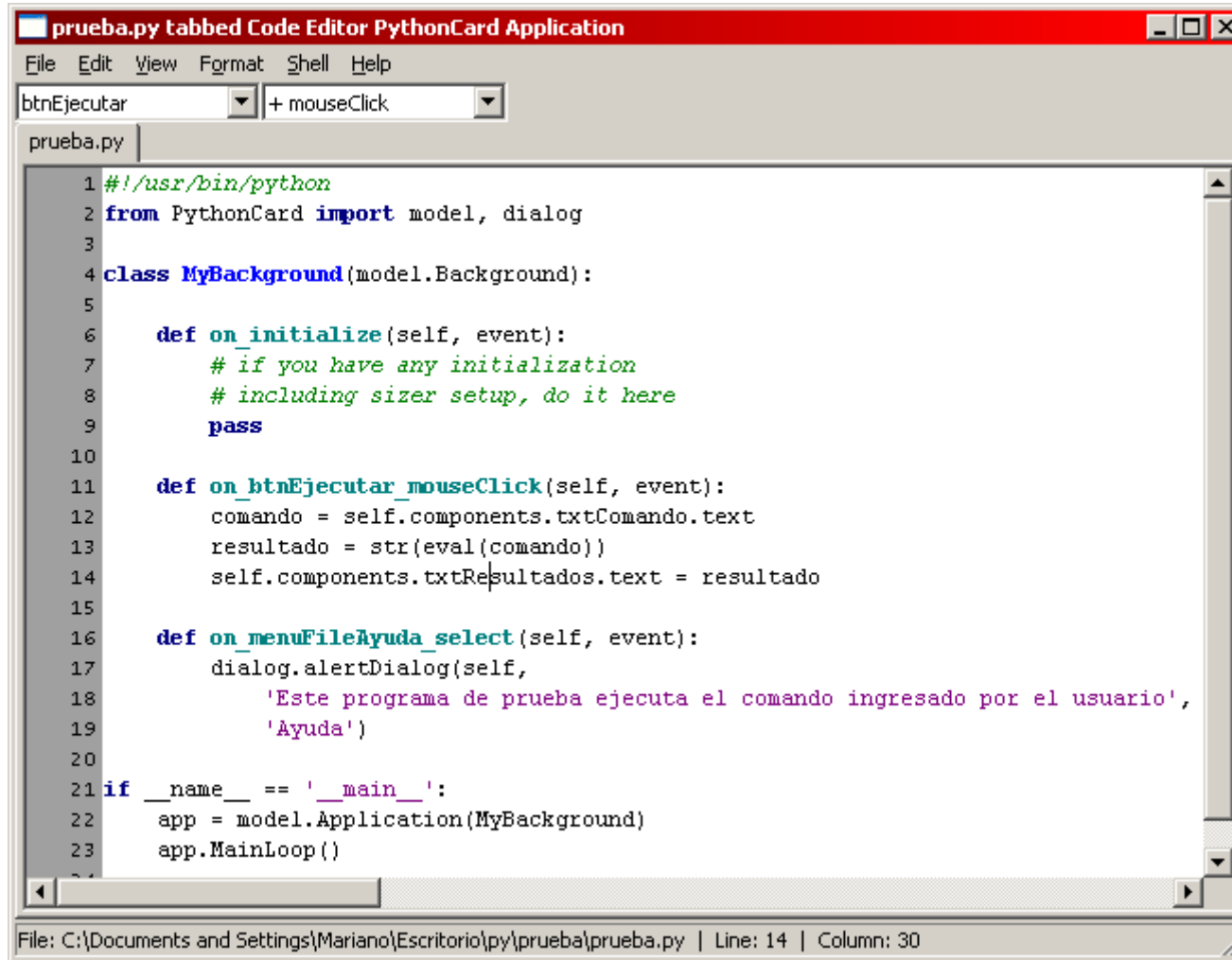
PythonCard es un conjunto de herramientas que facilitan el diseño de aplicaciones simples bajo la librería wxPython. Similar a VB clásico (diseño de pantallas, código simple, etc.)

Diseñador de Pantallas:



Aplicaciones GUI: PythonCard

Editor de Código:



The screenshot shows a window titled "prueba.py tabbed Code Editor PythonCard Application". The menu bar includes File, Edit, View, Format, Shell, and Help. Below the menu bar, there is a dropdown menu showing "btnEjecutar" and a button "+ mouseClicked". The main text area displays a Python script for a GUI application. The script defines a class "MyBackground" that inherits from "model.Background". It includes methods for initialization, handling button clicks, and handling menu selections. The script also includes a main block that creates an application and starts the main loop.

```
1#!/usr/bin/python
2from PythonCard import model, dialog
3
4class MyBackground(model.Background):
5
6    def on_initialize(self, event):
7        # if you have any initialization
8        # including sizer setup, do it here
9        pass
10
11    def on_btnEjecutar_mouseClick(self, event):
12        comando = self.components.txtComando.text
13        resultado = str(eval(comando))
14        self.components.txtResultados.text = resultado
15
16    def on_menuFileAyuda_select(self, event):
17        dialog.alertDialog(self,
18            'Este programa de prueba ejecuta el comando ingresado por el usuario',
19            'Ayuda')
20
21if __name__ == '__main__':
22    app = model.Application(MyBackground)
23    app.MainLoop()
```

File: C:\Documents and Settings\Mariano\Escritorio\py\prueba\prueba.py | Line: 14 | Column: 30

Aplicaciones GUI: PythonCard

```
from PythonCard import model, dialog
```

```
class FrmPrueba(model.Background):
```

```
    def on_initialize(self, event):
```

```
        pass
```

```
    def on_btnEjecutar_mouseClick(self, event):
```

```
        comando = self.components.txtComando.text
```

```
        resultado = str(eval(comando))
```

```
        self.components.txtResultados.text = resultado
```

```
        dialog.alertDialog(self, resultado, "Resultado!")
```

```
app = model.Application(FrmPrueba)
```

```
app.MainLoop()
```


“Ejecutables” e Instaladores

Archivo con directivas de instalación *setup.py*:

```
from distutils.core import setup  
import py2exe
```

```
setup(console=['miapp.py'])
```

Comandos para generar el .EXE e instalador:

```
python setup.py py2exe
```

```
7z.exe a -sfx setup.exe dist
```

Interfases COM con otros lenguajes

```
class MiClaseCOM:
```

```
    _public_methods_ = ['MiMetodo']
```

```
    _public_attrs_ = _readonly_attrs_ = []
```

```
    _reg_progid_ = "MiClaseCOM"
```

```
    _reg_clsid_ = "{6268820C-8900-4AE9-8A2D-F0A1EBD4CAC5}"
```

```
    def MiMetodo(self):
```

```
        return "hola mundo!"
```

```
import win32com.server.register
```

```
win32com.server.register.UseCommandLine(MiClaseCOM)
```

Interfases COM con otros lenguajes

Registrar el servidor COM:

```
C:\EJEMPLO> python miclasecom.py -register
```

Ejemplo de uso interfase COM en Visual Basic 5:

```
Set MiClaseCOM = CreateObject("MiClaseCOM")  
mensaje = MiClaseCOM.MiMetodo()
```

Python para usuarios de VB y PHP

- Tipos de datos: Fuerte y Dinámico
- Variables: Inmutables: “por valor”, Mutables: “por referencia”
- Variables Definidas e Inicializadas (sino, NameError!)
- Un solo objeto nulo: None (vs Nothing, Null, Empty, Missing)
- Listas, Diccionarios y Conjuntos (vs Vectores, Collection y Array)
- Función = Subrutina (siempre devuelven algo, o None)
- Parámetros por posición y/o nombre más flexibles
- No hay variables estáticas (usar parámetros mutables = {})
- Soporta programación funcional: lambda
- Soporte completo para POO. No hay atributos privados.
- No hay atributos por defecto, todo debe ser explícito
- Identación obligatoria, distinción entre mayúsculas y minúsculas

Características Avanzadas

Frameworks WEB:

- Django
- Web2Py
- TurboGears

Aplicaciones GUI:

- WxPython (wxGlade para pantallas)
- Qt
- GTK

Librerías avanzadas: matemáticas, ciencias, protocolos, formatos, internet, etc.

¿Preguntas?

¡Muchas gracias!

¡Los esperamos el Viernes 13 en la reunión N°35!

Proximamente novedades sobre la PyCon Argentina

www.python.org.ar

Mariano Reingart



Licencia: Creative Commons

Atribución-NoComercial-CompartirDerivadasIgual 2.5 Argentina

http://creativecommons.org/licenses/by-nc-sa/2.5/deed.es_AR