

M4. Introducción Ficheros PDF en Phyton

Para crear archivos **pdf** con Python utilizaremos el conjunto de librerías **OpenSource de ReportLab**.

1. Obviamente se necesita tener **Python** instalado, y en caso de que deseemos incluir imágenes debemos de instalar la *Librería de Imágenes de Python* (Python Imaging Library).

```
# apt-get install python-imaging
```

2. Luego nos dirigimos a la web de **ReportLAB** y descargamos el archivo .tar, descomprimos e instalamos:

```
# wget http://www.reportlab.com/ftp/reportlab-daily-unix.tar.gz
```

```
# tar zxvf report-dayli-unix.tar.gz
```

```
# cd reportlab-2012xxxxxxxxxx
```

```
# python setup.py install
```

3. Una vez instalado, podemos realizar una prueba sencilla *probandopdf.py*:

```
import os
from reportlab.pdfgen import canvas

def helloWorld(c):
    c.drawString(250,420,"Ubicado en el centro?")
    c.drawString(0,0,"Hola Mundo")

c = canvas.Canvas("HolaMundo.pdf")

helloWorld(c)
c.showPage()
c.save()

os.system("/usr/bin/evince HolaMundo.pdf")
```

4. Y al ejecutar el código anterior, deberíamos de obtener un documento pdf en el directorio. La última línea permitiría además lanzar un lector para leerlo. Similar a esto:



codigo ejemplo de un informe con reportlab en python

```
#!/usr/bin/python
#-----
# Fichero: informe.py
# Objetivo: Hacer un informe con los resultados de la encuesta
# -----

#-----
# Variables del programa. Autor: Pedro Reina
#-----

# El archivo con la base de datos
Origen = "encuesta.db"

# El PDF para imprimir
Destino = "informe.pdf"

# Consultas a la base de datos
ConsultaPregunta = "SELECT * FROM pregunta"
ConsultaPlazo = "SELECT * FROM plazo ORDER BY limite"

# Primera linea del documento
Titulo = "Encuesta"

# Los márgenes de la página, en centímetros
MargenIzquierdo = 2
MargenSuperior = 2

# La distancia entre líneas de texto, en centímetros
Interlineado = 0.75

#-----
# Módulos necesarios
#-----

# Another Python SQLite Wrapper
import apsw

# Matemáticas
import math

# Creación de PDF con ReportLab
from reportlab.pdfgen import canvas
from reportlab.lib.units import cm
from reportlab.lib.pagesizes import A4
from reportlab.graphics.shapes import Drawing, colors
from reportlab.graphics.charts.barcharts import VerticalBarChart
from reportlab.platypus import *
from reportlab.lib.styles import getSampleStyleSheet

#-----
# Funciones
#-----

#-----
# El programa
#-----

# Informamos al usuario
print "Origen: " + Origen
print "Destino: " + Destino
print "Preparando..."
```

```

# Abrimos conexión a la base de datos y creamos el curso con la conexión
Conexion = apsw.Connection (Origen)
Cursor = Conexion.cursor()

# Conversión de medidas: todas las hemos pedido en cm
# pero se usan en puntos tipográficos
MargenIzquierdo *= cm
MargenSuperior *= cm
Interlineado *= cm

# Preparamos el PDF
Pdf = canvas.Canvas (Destino, pagesize=A4)
Pdf.setAuthor ("Pedro Reina")
Pdf.setTitle ("Informe de encuesta")

# Guardamos las dimensiones de la página
Anchura, Altura = A4

# Preparamos el estilo de los párrafos con el texto de la pregunta
ListaEstilos = getSampleStyleSheet()
Estilo = ListaEstilos['BodyText']
Estilo.fontSize = 12

# Consultamos los límites del plazo de la consulta
Registro = 1
for Fila in Cursor.execute (ConsultaPlazo):
    if ( Registro == 1 ):
        Inicio = Fila[0]
        Registro = 2
    else:
        Final = Fila[0]

# Preparamos el título
Titulo += " de " + Inicio + " a " + Final

# Leemos los registros
Registro = 1
for Fila in Cursor.execute (ConsultaPregunta):

    # Tomamos los campos del registro
    Texto = Fila[0]
    R1 = Fila[1]
    R2 = Fila[2]
    R3 = Fila[3]
    R4 = Fila[4]
    R5 = Fila[5]

    # Corrección de tags
    Texto = Texto.replace ("<B>", "<b>")
    Texto = Texto.replace ("</B>", "</b>")

    # Numeramos las preguntas
    Texto = str(Registro) + ". " + Texto

    # Cálculos estadísticos
    Maximo = max (R1, R2, R3, R4, R5)
    NumRespuestas = R1 + R2 + R3 + R4 + R5
    Media = float (R1 + 2*R2 + 3*R3 + 4*R4 + 5*R5) / NumRespuestas
    DesvTipica = math.sqrt (float (R1 + 4*R2 + 9*R3 + 16*R4 + 25*R5) \
        / NumRespuestas - Media*Media)
    Media = round (Media, 1)
    DesvTipica = round (DesvTipica, 1)

```

```

# Descripción de la estadística
Estadistica = "Respuestas: " + str (NumRespuestas) + ". " + \
              "Media: " + str (Media) + ". " + \
              "Desviación típica: " + str (DesvTipica)

# Vamos a imprimir dos preguntas por página
# Esta variable indica la posición inicial de dibujo de la
# pregunta en la página
if ( Registro % 2 == 1 ):
    Base = 0
else:
    Base = - Altura / 2

# Posición inicial de las líneas escritas
Linea = Base + Altura - MargenSuperior

# Escribimos los textos
Pdf.setFont ("Helvetica", 12)
Pdf.drawString (MargenIzquierdo, Linea, Titulo)
Linea -= Interlineado
Parrafo = Paragraph (Texto, Estilo)
AnchuraParrafo, AlturaParrafo = \
    Parrafo.wrapOn (Pdf, Anchura - 2 * MargenIzquierdo, Altura)
Linea -= AlturaParrafo
Parrafo.drawOn (Pdf, MargenIzquierdo, Linea)
Linea -= Interlineado
Pdf.drawString (MargenIzquierdo, Linea, Estadistica)

# El diagrama de barras
Dibujo = Drawing (400, 200)
Datos = [ (R1, R2, R3, R4, R5) ]
Diagrama = VerticalBarChart()
Diagrama.x = 50
Diagrama.y = 50
Diagrama.height = 140
Diagrama.width = 300
Diagrama.data = Datos
Diagrama.bars[0].fillColor = colors.gray
Diagrama.barLabelFormat = '%d'
Diagrama.barLabels.dy = 8
Diagrama.valueAxis.valueMin = 0
Diagrama.valueAxis.valueMax = max (Maximo, 1)
Diagrama.valueAxis.valueStep = max (Maximo/4, 1)
Diagrama.categoryAxis.labels.dy = -2
Diagrama.categoryAxis.categoryNames = ['1', '2', '3', '4', '5']
Dibujo.add (Diagrama)
Dibujo.drawOn (Pdf, 100, Base+520-AlturaParrafo)

# Fin de página
if ( Registro % 2 == 0 ):
    Pdf.showPage()
Registro += 1

# Escribimos el PDF
Pdf.save()

# Cerramos la conexión
Conexion.close (True)

# Informamos al usuario
print "Terminado."

```