

★ Python 3 para impacientes ★



"Simple es mejor que complejo" (Tim Peters)

Python	IPython	EasyGUI	Tkinter	JupyterLab	Numpy
--------	---------	---------	---------	------------	-------

martes, 11 de febrero de 2014

Bases de datos SQLite3



SQLite es un sistema de gestión de base de datos de tipo relacional. Es un proyecto de dominio público creado por D. Richard Hipp que ofrece un gestor de base de datos liviano y robusto; y accesible desde programas escritos en los lenguajes de programación más populares. Algunos de ellos lo tienen embebido y no requieren la instalación de software adicional. Hay programas como Firefox, Opera, Skype, Clementine, OpenOffice y otros que utilizan SQLite.

En este blog hay una [Guía Rápida de SQLite](#) para conocer sus fundamentos.

También, en el sitio oficial de [SQLite](#) hay abundante documentación y en otros sitios hay herramientas gráficas como [SQLiteman](#) y [Sqlite Database Browser](#) que facilitan mucho el trabajo con las bases de datos.

En Python 3 el módulo **sqlite3** permite crear bases de datos y operar con tablas y registros. Para ello, es necesario importar dicho módulo:

```
import sqlite3
```

A continuación, basándonos en ejemplos, mostraremos las operaciones más comunes sobre una base de datos SQLite. Antes, crearemos la base de datos y una tabla desde la línea de comandos.

Crear base de datos

```
sqlite3 contactos.db
```

Crear tabla

```
sqlite3 CREATE TABLE agenda (ident INTEGER PRIMARY KEY, nombre VARCHAR(30)
UNIQUE, correo VARCHAR(40), telefono INT(9));
```

Iniciar y cerrar una conexión con la base de datos

```
con_bd = sqlite3.connect('contactos.db')
con_bd.close()
```

Declarar y cerrar un cursor

```
cursor_agenda = con_bd.cursor()
cursor_agenda.close()
```

Añadir registro a la tabla agenda

Buscar

Python para impacientes

[Python](#)
[IPython](#)
[EasyGUI](#)
[Tkinter](#)
[JupyterLab](#)
[Numpy](#)

Anexos

[Guía urgente de MySQL](#)
[Guía rápida de SQLite3](#)

Entradas + populares

[Dar color a las salidas en la consola](#)

En Python para dar color a las salidas en la consola (o en la terminal de texto) existen varias posibilidades. Hay un método basado ...

[Instalación de Python, paso a paso](#)

Instalación de Python 3.6 A finales de 2016 se produjo el lanzamiento de Python 3.6. El propósito de esta entrada es mostrar, pas...

[Añadir, consultar, modificar y suprimir elementos en Numpy](#)

Acceder a los elementos de un array. [], [..], ... Acceder a un elemento de un array. Para acceder a un elemento se utiliz...

[Variables de control en Tkinter](#)

Variables de control Las variables de control son objetos especiales que se asocian a los widgets para almacenar sus valore...

[Cálculo con arrays Numpy](#)

Numpy ofrece todo lo necesario para obtener un buen rendimiento cuando se trata de hacer cálculos con arrays. Por como está concebido...

[Tkinter: interfaces gráficas en Python](#)

Introducción Con Python hay muchas posibilidades para programar una interfaz gráfica de usuario (GUI) pero Tkinter es fácil d...

[Operaciones con fechas y horas. Calendarios](#)

Los módulos datetime y calendar amplían las posibilidades del módulo time que provee funciones para manipular expresiones de ti...

[Convertir, copiar, ordenar, unir y dividir arrays Numpy](#)

Esta entrada trata sobre algunos métodos que se utilizan en Numpy para convertir listas en arrays y viceversa; para copiar arrays d...

[Tkinter: Tipos de ventanas](#)

Ventanas de aplicación y de diálogos En la entrada anterior tratamos los distintos gestores de geometría que se utilizan para di...

[Threading: programación con hilos \(I\)](#)

En programación, la técnica que permite que una aplicación ejecute

```
reg = (1, "A", "a@a.a", 1)
cursor_agenda.execute("INSERT INTO agenda VALUES(?,?,?,?)", reg)
```

Completar la transacción de inserción de registro

```
con_bd.commit()
```

Consultar todos los registros de la tabla agenda

```
cursor_agenda.execute("SELECT * FROM agenda")
for registro in cursor_agenda:
    print(registro)
```

Consultar registro con parámetro

```
par = (1,)
cursor_agenda.execute("SELECT * FROM agenda WHERE ident=?", par)
for registro in cursor_agenda:
    print(registro)
```

Consultar los registros, uno a uno, con fetchone()

```
cursor_agenda.execute("SELECT * FROM agenda WHERE ident>?", par)
registro = cursor_agenda.fetchone() # Lee el primer registro
print(registro)
```

Consultar un nº de registros concretos con fetchmany()

```
par = (0,)
cursor_agenda.execute("SELECT * FROM agenda WHERE ident>?", par)
registros = cursor_agenda.fetchmany(2) # 2 registros
for registro in registros: print(registro)
```

Consultar todos los registros, con fetchall()

```
cursor_agenda.execute("SELECT * FROM agenda WHERE ident>?", par)
registros = cursor_agenda.fetchall()
for registro in registros:
    print(registro)
```

Mostrar los campos de un registro por su posición

```
par = (2,)
cursor_agenda.execute("SELECT * FROM agenda WHERE ident=?", par)
for campo in cursor_agenda:
    print("Identificador:", campo[0])
    print("Nombre.....:", campo[1])
```

Consultar registros por nombre de campos

```
con_bd = sqlite3.connect("contactos.db")
con_bd.row_factory = sqlite3.Row
cursor_agenda = con_bd.cursor()
cursor_agenda.execute("SELECT * FROM agenda WHERE ident=?", par)
registro = cursor_agenda.fetchone()
print("Nombre", registro['nombre'])
print("Correo", registro['correo'])
```

Borrar registro

```
values = (1,)
cursor_agenda.execute('DELETE FROM agenda WHERE ident=?',
                      values)
con_bd.commit()
```

simultáneamente varias operaciones en el mismo espacio de proceso se...

Archivo

febrero 2014 (17) ▾

python.org



pypi.org



Sitios

- [ActivePython](#)
- [Anaconda](#)
- [Bpython](#)
- [Django](#)
- [Flask](#)
- [Ipython](#)
- [IronPython](#)
- [Matplotlib](#)
- [MicroPython](#)
- [Numpy](#)
- [Pandas](#)
- [Pillow](#)
- [PortablePython](#)
- [PyBrain](#)
- [PyCharm](#)
- [PyDev](#)
- [PyGame](#)
- [Pypi](#)
- [PyPy](#)
- [Pyramid](#)
- [Python.org](#)
- [PyTorch](#)
- [SciPy.org](#)
- [Spyder](#)
- [Tensorflow](#)
- [TurboGears](#)

Modificar campos

```
cursor_agenda.execute("UPDATE agenda SET ecorreo='a@b' WHERE ident=3")
con_bd.commit()
```

Deshacer modificación

```
con_bd.rollback()
```

Relacionado:

- [Base de Datos SQLite3 con APSW](#)
- [Guía Rápida de SQLite3](#)

[Ir al índice del tutorial de Python](#)

Publicado por Pherkad en [14:35](#)



Etiquetas: [Python3](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

2014-2020 | Alejandro Suárez Lamadrid y Antonio Suárez Jiménez, Andalucía - España
. Tema Sencillo. Con la tecnología de [Blogger](#).