

★ Python 3 para impacientes ★



"Simple es mejor que complejo" (Tim Peters)

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

domingo, 17 de mayo de 2015

Bases de datos documentales con TinyDB (y II)



En el [capítulo anterior](#) comenzamos el estudio de **TinyDB**, una base de datos documental dirigida a proyectos modestos bastante atractiva.

En dicho capítulo se expuso la forma de instalar el módulo Python necesario para trabajar con esta base de datos y se mostraron algunos ejemplos que ilustran algunas operaciones básicas.

A continuación, seguimos con más ejemplos para enseñar algunas características avanzadas de TinyDB. Para su realización recomendamos el [entorno interactivo de Python](#) o [IPython](#).

Uso avanzado de TinyDB

Insertar múltiples registros de una vez

```
from tinydb import TinyDB, where
cielo = TinyDB('cielo.db')
constelacion = cielo.table('constelacion')
constelacion.insert_multiple([{'nombre': 'Casiopea', 'abrev': 'CAS'},
                              {'nombre': 'Cefeo', 'abrev': 'CEP'}]) # [1,2]
constelacion.all()
```

```
[{'abrev': 'CAS', 'nombre': 'Casiopea'}, {'abrev': 'CEP', 'nombre': 'Cefeo'}]
```

Insertar registros con campos diferentes

```
constelacion.insert({'nombre': 'Dragón',
                    'abrev': 'DRA', 'sup': 1083}) # 3
constelacion.all()
```

```
[{'abrev': 'CAS', 'nombre': 'Casiopea'},
 {'abrev': 'CEP', 'nombre': 'Cefeo'},
 {'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083}]
```

En una base de datos TinyDB todos los registros no tienen necesariamente que tener lo mismos campos.

Insertar un nuevo campo en un registro en una actualización

```
constelacion.update({'sup': 588}, where('nombre') == 'Cefeo')
constelacion.all()
```

```
[{'abrev': 'CAS', 'nombre': 'Casiopea'},
 {'abrev': 'CEP', 'nombre': 'Cefeo', 'sup': 588},
```

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. [], [1], ... 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

```
{'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083}]
```

Insertar múltiples registros insertando un nuevo campo con un valor autoincrementado

```
constelacion.insert_multiple({'obs': 'invierno',
                             'indice': ind} for ind in range(5))
registros = constelacion.search(where('indice') != -1)
```

```
registros: [{'indice': 0, 'obs': 'invierno'},
{'indice': 1, 'obs': 'invierno'},
{'indice': 2, 'obs': 'invierno'},
{'indice': 3, 'obs': 'invierno'},
{'indice': 4, 'obs': 'invierno'}]
```

Actualizar borrando un campo de un registro que cumpla una condición

```
from tinydb.operations import delete, increment, decrement
constelacion.update(delete('indice'), where('indice') == 0)
constelacion.all()
```

```
[{'abrev': 'CAS', 'nombre': 'Casiopea'},
{'abrev': 'CEP', 'nombre': 'Cefeo', 'sup': 588},
{'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083},
{'obs': 'invierno'},
{'indice': 1, 'obs': 'invierno'},
{'indice': 2, 'obs': 'invierno'},
{'indice': 3, 'obs': 'invierno'},
{'indice': 4, 'obs': 'invierno'}]
```

Actualizar incrementando en 1 el valor numérico de un campo

```
constelacion.update(increment('indice'),
                    where('indice') != -1)
constelacion.all()
```

```
[{'abrev': 'CAS', 'nombre': 'Casiopea'},
{'abrev': 'CEP', 'nombre': 'Cefeo', 'sup': 588},
{'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083},
{'obs': 'invierno'},
{'indice': 2, 'obs': 'invierno'},
{'indice': 3, 'obs': 'invierno'},
{'indice': 4, 'obs': 'invierno'},
{'indice': 5, 'obs': 'invierno'}]
```

Actualizar decrementando en 1 el valor numérico de un campo

```
constelacion.update(decrement('indice'),
                    where('indice') != -1)
constelacion.all()
```

```
[{'abrev': 'CAS', 'nombre': 'Casiopea'},
{'abrev': 'CEP', 'nombre': 'Cefeo', 'sup': 588},
{'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083},
{'obs': 'invierno'},
{'indice': 1, 'obs': 'invierno'},
{'indice': 2, 'obs': 'invierno'},
{'indice': 3, 'obs': 'invierno'},
{'indice': 4, 'obs': 'invierno'}]
```

Obtener el número de registros que tiene una tabla

```
num_registros = len(constelacion)
num_registros # 8
```

Leer un registro que cumpla una condición (get)

```
registro = constelacion.get(where('obs') == 'invierno')
registro # {'obs': 'invierno'}
```

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

Archivo

mayo 2015 (3) ▾

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](#)

Si no existen registros que cumplan la condición la función devuelve None:

```
registro = constelacion.get(where('obs') == 'verano')
registro # None
```

Conocer si hay registros que cumplan una condición

```
hay_registros = constelacion.contains(where('sup') > 500)
hay_registros # True

hay_registros = constelacion.contains(where('sup') > 1500)
hay_registros # False
```

Contar los registros que cumplan una condición

```
num_registros = constelacion.count(where('sup') > 500)
num_registros # 2
```

Consultar un registro por su ID

```
registro = constelacion.get(eid=3)
print(registro['nombre'], registro['abrev'])
```

Dragón DRA

Obtener el ID de un registro que cumpla una condición

```
registro = constelacion.get(where('nombre') == 'Cefeo')
registro.eid # 2
```

Actualizar un campo de varios registros por sus ID

```
constelacion.update({'sup': 0}, eids=[7, 8])
constelacion.all()
```

```
[{'abrev': 'CAS', 'nombre': 'Casiopea'},
 {'abrev': 'CEP', 'nombre': 'Cefeo', 'sup': 588},
 {'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083},
 {'obs': 'invierno'},
 {'indice': 1, 'obs': 'invierno'},
 {'indice': 2, 'obs': 'invierno'},
 {'indice': 3, 'obs': 'invierno', 'sup': 0},
 {'indice': 4, 'obs': 'invierno', 'sup': 0}]
```

Borrar varios registros por su ID

```
constelacion.remove(eids=[4, 5, 6, 7, 8])
constelacion.all()
```

```
[{'abrev': 'CAS', 'nombre': 'Casiopea'},
 {'abrev': 'CEP', 'nombre': 'Cefeo', 'sup': 588},
 {'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083}]
```

Consultar registros que no cumplan condición (NOT ~)

```
registros = constelacion.search(~(where('nombre') == 'Casiopea'))
```

```
registros: [{'abrev': 'CEP', 'nombre': 'Cefeo', 'sup': 588},
 {'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083},
 {'obs': 'invierno'},
 {'indice': 1, 'obs': 'invierno'},
```

```
{'indice': 2, 'obs': 'invierno'},
{'indice': 3, 'obs': 'invierno', 'sup': 0},
{'indice': 4, 'obs': 'invierno', 'sup': 0}]
```

Se obtienen todos los registros con nombre distinto a 'Casiopea' y aquellos que no tienen el campo *nombre*.

Consultar registros que cumplan una condición u otras (OR "|")

```
registros = constelacion.search((where('abrev') == 'CAS') \
| (where('abrev') == 'CEP'))
```

```
registros: [{'abrev': 'CAS', 'nombre': 'Casiopea'},
{'abrev': 'CEP', 'nombre': 'Cefeo', 'sup': 588}]
```

Cada condición debe escribirse delimitada por paréntesis.

Consultar registros que cumplan más de una condición (AND "&")

```
registros = constelacion.search((where('nombre') != 'Cefeo') \
& (where('sup') > 500))
```

```
registros: [{'abrev': 'DRA', 'nombre': 'Dragón', 'sup': 1083}]
```

Cada condición debe escribirse delimitada por paréntesis.

Consultar registros utilizando expresiones regulares

```
registros = constelacion.search(where('nombre').matches('^C'))
print(registros)
# [{'abrev': 'CAS', 'nombre': 'Casiopea'},
# {'abrev': 'CEP', 'sup': 588, 'nombre': 'Cefeo'}]

registros = constelacion.search(where('nombre').contains('C+'))
print(registros)
# [{'abrev': 'CAS', 'nombre': 'Casiopea'},
# {'abrev': 'CEP', 'sup': 588, 'nombre': 'Cefeo'}]
```

Consultar registros filtrando con una función

```
supmayora1000 = lambda superficie: superficie >= 1000
registros = constelacion.search(where('sup').test(supmayora1000))
```

```
registros: [{'abrev': 'DRA', 'sup': 1083, 'nombre': 'Dragón'}]
```

Insertar registros con campos anidados (diccionarios)

```
constelacion.purge()
# suprimimos todos los registros existentes

constelacion.insert({'nombre': {'esp': 'Tauro', 'latin': 'Taurus'},
'caract': {'abrev': 'TAR', 'sup': 797}})
constelacion.insert({'nombre': {'esp': 'Cuervo', 'latin': 'Corvus'},
'caract': {'abrev': 'CRV', 'sup': 184}})
```

Consultar registros con campos anidados (diccionarios)

```
registro = constelacion.search(where('nombre').has('latin') \
== 'Corvus')
```

```
registro: [{'nombre': {'latin': 'Corvus', 'esp': 'Cuervo'}, 'caract': {'abrev': 'CRV', 'sup': 184}}]
```

Insertar registros con campos anidados (listas)

```
# suprimimos antes todos los registros existentes
constelacion.purge()
```

```
constelacion.insert({'nombre': 'Tauro',
                    'magnitud': [{'val': 0.87 },
                                 {'val': 1.68 },
                                 {'val': 3.5}]}))

constelacion.insert({'nombre': 'Cuervo',
                    'magnitud': [{'val': 4.02 },
                                 {'val': 2.65 },
                                 {'val': 2.58}]}))
```

Consultar registros con campos anidados (listas) que al menos uno cumpla la condición

```
registros = constelacion.search(where('magnitud').any(where('val') \
> 4))
```

```
registros: [{'magnitud': [{'val': 4.02}, {'val': 2.65}, {'val': 2.58}],
'nombre': 'Cuervo']]
```

Consultar registros con campos anidados (listas) que todos cumplan la condición

```
registros = constelacion.search(where('magnitud').all(where('val') \
> 2))
```

```
registros: [{'magnitud': [{'val': 4.02}, {'val': 2.65}, {'val': 2.58}],
'nombre': 'Cuervo']]
```

Consultar registros con campos anidados (listas) que contengan algún valor de la lista

```
registros = constelacion.search(where('magnitud').any([{'val': 0.87},
                                                         {'val': 1.67}])))
```

```
registros: [{'magnitudes': [{'val': 0.87}, {'val': 1.68}, {'val': 3.5}],
'nombre': 'Tauro']]
```

Crear base de datos utilizando la memoria convencional como soporte de almacenamiento

```
from tinydb.storages import MemoryStorage
cielo = TinyDB(storage=MemoryStorage)
constelacion = cielo.table('constelacion')
constelacion.insert({'nombre': 'Casiopea', 'abrev': 'CAS'})
constelacion.all()
```

```
[{'nombre': 'Casiopea', 'abrev': 'CAS'}]
```

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

Publicado por Pherkad en [4:31](#)



Etiquetas: [Python3](#), [TinyDB](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)