

★ Python 3 para impacientes ★

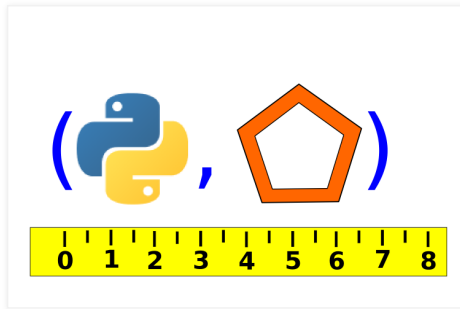


"Simple es mejor que complejo" (Tim Peters)

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

jueves, 16 de abril de 2015

Tuplas a medida con namedtuple



namedtuple del módulo **collections** se utiliza para definir subclases de la clase **tuple**, que permiten crear **tuplas** a medida a los programadores. El nombre del nuevo tipo de tupla será el que se indique en el primer argumento y constará de los campos que se incluyan en la lista del segundo argumento:

```
namedtuple(typename, field_names, verbose=False, rename=False)
```

Si el argumento **rename** es **True** los nombres de campos que no sean válidos serán sustituidos, automáticamente, por nombres posicionales ('_1', '_2', ...). También, si el argumento **verbose** es **True** la definición de la clase será impresa después de que se construya. Esta última opción que tiende a desaparecer se obtiene también mediante el atributo **_source**.

Los nombres de los campos pueden contener caracteres alfabéticos, numéricos y guiones bajos; y no pueden comenzar con caracteres numéricos o guiones bajos, ni pueden ser, lógicamente, palabras reservadas del lenguaje Python.

```
import collections

# Crear un nuevo tipo de tupla para contener datos
# de elementos químicos

elemento = collections.namedtuple('elemento',
                                   ['nombre', 'simbolo', 'numato'])

# Crear un nuevo objeto del tipo elemento,
# con datos del Azufre

elem = elemento('Azufre', 'S', 16)

# Mostrar información del objeto anterior

print(elem.nombre, elem.simbolo, elem.numato) # Azufre S 16
print(elem[0], elem[1], elem[2]) # Azufre S 16
print(elem) # elemento(nombre='Azufre', simbolo='S', numato=16)

# Conocer de qué tipo es el objeto 'elem'

print(type(elem)) #

# Mostrar la definición de la subclase 'elem'

print(elem._source)

# Mostrar Los nombres de campos de la subclase 'elem'

print(elem._fields)

# nombres de campos: ('nombre', 'simbolo', 'numato')
```

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...

[El módulo random](#)

El módulo random de la librería estándar de Python incluye un conjunto de funciones

```
# Sustituir Los valores actuales

elem = elem._replace(nombre='Carbono', simbolo='C', numato=6)

# Mostrar información

print(elem.nombre) # 'Carbono'

# Mostrar tipo de diccionario con Los campos ordenados
# como en una tupla

print(elem._asdict())

# Obtener el valor del campo indicado

print(getattr(elem, 'simbolo')) # 'C'

# Crear otro tipo nuevo para contener coordenadas de
# un plano y color

coordenada = collections.namedtuple('coordenada',
                                     ['x', 'y', 'color'])

coord = coordenada(12, 53, 'blue')

# Mostrar valores

print('x=', coord.x)
print('y=', coord.y)
print('color=', coord.color)

x, y, color = coord
print(x, y, color)
```

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

Publicado por Pherkad en [11:47](#)



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

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

que permiten obtener de distintos modos
números a...

Archivo

abril 2015 (6) ▼

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](#)