

# ★ Python 3 para impacientes ★

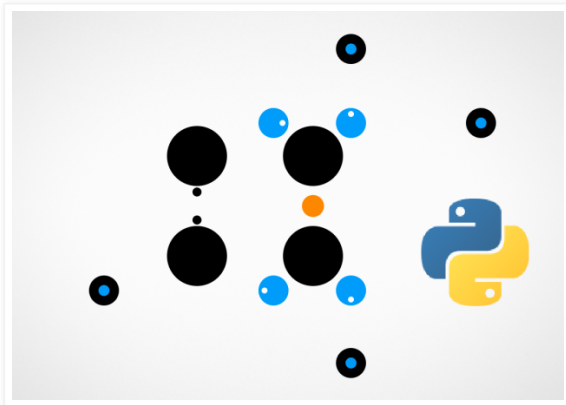


"Simple es mejor que complejo" (Tim Peters)

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

lunes, 13 de abril de 2015

## Counter, el contador de Python



El módulo **collections** provee nuevos tipos de datos mejorados que derivan, según el caso, de [una lista](#), [una tupla](#), [un diccionario](#) o [conjunto de datos \(set\)](#). En este artículo vamos a estudiar el objeto **Counter**.

Un objeto **Counter** es un contenedor del módulo **collections** que se utiliza para contar las veces que aparece un valor en una secuencia de caracteres, en una lista, un diccionario o una lista de nombres de argumentos con asignaciones. Este tipo de objeto es de una subclase que está basada en la clase **dict** (diccionario).

A continuación, se muestran distintas formas de construir objetos **Counter** y se realizan algunas operaciones que demuestran su afinidad con la clase **dict**.

```
import collections
lista1 = ['Granada', 'Huelva', 'Sevilla', 'Granada',
          'Granada', 'Sevilla', 'Sevilla']

# Crear un objeto Counter a partir de la lista lista1
cuenta1 = collections.Counter(lista1)

# En el objeto Counter las claves son los
# elementos (sin repetir) de la lista y los valores el
# número de veces que aparece cada elemento en la lista.
print(cuenta1)
# Counter({'Granada': 3, 'Sevilla': 3, 'Huelva': 1})

lista1.append('Huelva') # Añade un nuevo elemento
cuenta1 = collections.Counter(lista1)
print(cuenta1)
# Counter({'Granada': 3, 'Sevilla': 3, 'Huelva': 2})

# Creación de un objeto Counter a partir de un diccionario.
dicc1 = {'Granada': 3, 'Sevilla': 3, 'Huelva': 1}
cuenta2 = collections.Counter(dicc1)
cuenta3 = collections.Counter(Granada=3, Sevilla=3, Huelva=1)

# Un objeto contador se puede construir vacío y después
# actualizar con el método update().

cuenta4 = collections.Counter()
print(cuenta4)
cuenta4.update('xxxyzzzyxyx')
print(cuenta4)
# Counter({'x': 6, 'y': 3, 'z': 2})
```

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 valores...

#### [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

```

cuenta4.update({'m':1, 'n':2})
print(cuenta4)
# Counter({'x': 6, 'y': 3, 'n': 2, 'z': 2, 'm': 1})

# La forma de acceder al valor de algún elemento del Contador
# es igual que en un diccionario:
print(cuenta4['x']) # 6
print(cuenta4['a']) # 0

# Para obtener una Lista con Los elementos de un objeto Counter:
lista4 = list(cuenta4.elements())
print(lista4)
# ['m', 'y', 'y', 'y', 'n', 'n', 'x',
#  'x', 'x', 'x', 'x', 'x', 'z', 'z']

# Los elementos de La Lista no estarán necesariamente ordenados.
# Para ordenar una Lista utilizaremos el método sort().
lista4.sort()
print(lista4)
# ['m', 'n', 'n', 'x', 'x', 'x', 'x',
#  'x', 'x', 'y', 'y', 'y', 'z', 'z']

# Para obtener una Lista de tuplas con Los elementos más
# comunes de un Contador.
print(cuenta4.most_common(1))
# Obtiene el elemento que más se repite: [('x', 6)]

print(cuenta4.most_common(3))
# 3 elementos más repetidos: [('x', 6), ('y', 3), ('n', 2)]

```

Para contar las palabras de una frase podemos crear una lista utilizando el método **split()** para dividir la frase en tantos trozos como palabras existan. Por defecto, **split()** utiliza el espacio en blanco como separador:

```

sagan = 'La ausencia de prueba no es prueba de ausencia'
lista5 = sagan.split()
cuenta5 = collections.Counter(lista5)

for clave, valor in cuenta5.items():
    print(clave,':',valor)
# no : 1, ausencia : 2, La : 1, de : 2, es : 1, prueba : 2

```

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

Publicado por Pherkad en [15:10](#)



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](#)