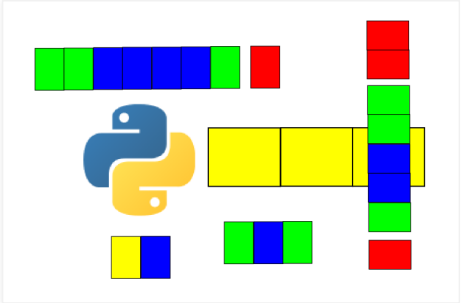
Python 3 para impacientes

"Simple es mejor que complejo" (Tim Peters)

martes, 14 de abril de 2015

Objeto deque, más que una lista



Un objeto **deque** es un contenedor de datos del módulo **collections** similar a una lista o una cola que permite añadir o suprimir elementos por sus dos extremos.

Se construye a partir de objetos *iterables* (que permiten recorrer sus elementos) como una secuencia de caracteres, una lista o una tupla. En el caso de los diccionarios el objeto **deque** se construirá con las claves existentes en el mismo.

El argumento opcional ***maxlen*** se utiliza para limitar el número de elementos que puede contener el objeto **deque**:

deque*([iterable[, maxlen]])*

A continuación, se muestra cómo crear un objeto deque a partir de una lista y se utilizan varios métodos propios de listas para consultar la longitud del objeto; añadir y borrar elementos.

```
import collections
documentos = ['doc1', 'doc3', 'doc4', 'doc5']
docs = collections.deque(documentos)
print('Deque:', docs)
print('Longitud:', len(docs))  # mostrar número de elementos deque
print('Elemento de más a la izquierda:', docs[0])
print('Elemento de más a la derecha:', docs[-1])
docs.remove('doc3')  # borrar el elemento indicado
print('remove(doc3):', docs)
docs.append('doc6')  # añadir un elemento por la derecha
print(docs)  # deque(['doc1', 'doc4', 'doc5', 'doc6'])
listadocs = list(docs)  # Obtener lista con todos
```

La ventaja principal de este tipo de contenedor está en que permite agregar o suprimir elementos por la izquierda:

```
# Para añadir un elemento por la izquierda:
docs.appendleft('doc0')

print(docs)
# deque(['doc0', 'doc1', 'doc4', 'doc5', 'doc6'])

# Para extender la lista por la izquierda con nuevos
# elementos:
docs.extendleft(['libro1', 'libro2'])

print(docs)
# deque(['libro2', 'libro1', 'doc0',
         'doc1', 'doc4', 'doc5', 'doc6'])

# Para borrar elementos por cualquiera de los extremos:
docs.pop()  # borra elemento de más a derecha: 'doc6'
```

Buscar

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...

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...

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...

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 concebid...

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...

Threading: programación con hilos (I)
En programación, la técnica que permite que una aplicación ejecute simultáneamente varias operaciones en el mismo espacio de proceso se...

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

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

```
print(docs)
# deque(['libro2', 'libro1', 'doc0',
#        'doc1', 'doc4', 'doc5'])

docs.popleft()  # borra elemento de más a izquierda: 'libro2'

print(docs)
# deque(['libro1', 'doc0', 'doc1', 'doc4', 'doc5'])

# En caso de no existir elementos para borrar se producirá
# la siguiente excepción:
# IndexError: pop from an empty deque

# Para contar el número de veces que aparece el elemento
# indicado (desde Python 3.2):
docs.count('doc4')  # 1

# Para invertir el orden de los elementos (desde Python 3.2):
docs.reverse()
print(docs)
# deque(['doc5', 'doc4', 'doc1', 'doc0', 'libro1'])

# Para rotar o pasar un número de elementos desde la derecha
# a la izquierda, o viceversa:

docs.rotate(2)  # rota 2 elementos desde derecha a izquierda
print(docs)
# deque(['doc0', 'libro1', 'doc5', 'doc4', 'doc1'])

docs.rotate(-1)  # rota 1 elemento desde izquierda a derecha
print(docs)
# deque(['libro1', 'doc5', 'doc4', 'doc1', 'doc0'])

# Para borrar todos los elementos:

docs.clear()
print(docs)  # deque([])
```

Ir al índice del tutorial de Python

Publicado por Pherkad en 12:19

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