

★ Python 3 para impacientes ★

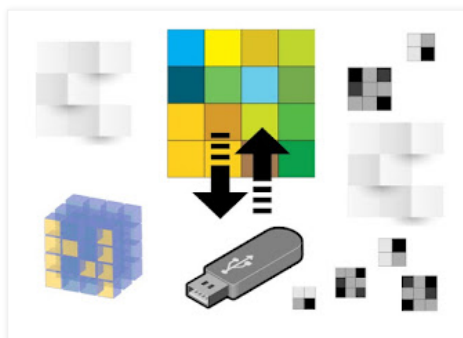


"Simple es mejor que complejo" (Tim Peters)

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

sábado, 25 de enero de 2020

Leer y escribir arrays Numpy en archivos



Para guardar de manera permanente los datos de memoria de un array Numpy para su posterior uso existen distintas opciones según sea el archivo de tipo binario, texto y CSV (Valores separados por comas).

Leer y escribir arrays Numpy en archivos de texto

loadtxt() y savetxt()

Leer y escribir los datos de arrays en archivos de texto.

```
# Escribir en un archivo de texto un array (10x3) generado
# con números aleatorios del 1 al 4:

a = np.random.randint(1, 5, (10,3), dtype=np.int8)
print(a)

# [[4 3 4]
#  [3 2 2]
#  [3 3 1]
#  [3 1 3]
#  [3 3 1]
#  [4 1 3]
#  [4 2 3]
#  [4 1 3]
#  [3 4 3]
#  [4 4 3]]

np.savetxt('datos.txt', a)

# Leer los datos de un archivo de texto y declarar un array:

a = np.loadtxt('datos.txt')
print(a)

# [[4. 3. 4.]
#  [3. 2. 2.]
#  [3. 3. 1.]
#  [3. 1. 3.]
#  [3. 3. 1.]
#  [4. 1. 3.]
#  [4. 2. 3.]
#  [4. 1. 3.]
#  [3. 4. 3.]
#  [4. 4. 3.]]
```

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 concebido...

[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...

[Tkinter: interfaces gráficas en Python](#)

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

[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

Leer y escribir arrays Numpy en archivos CSV

loadtxt() y savetxt()

Leer y escribir los datos de arrays en archivos CSV (valores separados por comas).

```
# Escribir en un archivo CSV un array (10x3) generado con
# números aleatorios del 1 al 4:

a = np.random.randint(1, 5, (10,3))
print(a)
np.savetxt("datos.csv", a, delimiter=",")

# [[1 2 2]
#  [2 1 4]
#  [3 1 1]
#  [3 3 1]
#  [3 3 3]
#  [4 2 3]
#  [2 2 1]
#  [3 2 4]
#  [1 2 4]
#  [3 4 2]]

# Leer los datos de un archivo CSV y declarar un array:

a = np.loadtxt('datos.csv', dtype=np.int8, delimiter=',', skiprows=0)
print(a)

# [[1 2 2]
#  [2 1 4]
#  [3 1 1]
#  [3 3 1]
#  [3 3 3]
#  [4 2 3]
#  [2 2 1]
#  [3 2 4]
#  [1 2 4]
#  [3 4 2]]
```

Leer y escribir los datos de arrays en archivos binarios

fromfile() y tofile()

Leer y escribir los datos de arrays en archivos binarios.

```
# Escribir en un archivo binario un array (10x3) generado
# con números aleatorios del 1 al 4:

a = np.random.randint(1, 5, (10,3), dtype=np.int8)
print(a)

# [[4 3 4]
#  [3 2 2]
#  [3 3 1]
#  [3 1 3]
#  [3 3 1]
#  [4 1 3]
#  [4 2 3]
#  [4 1 3]
#  [3 4 3]
#  [4 4 3]]

a.tofile('datos.dat')

# Leer los datos del archivo binario y declarar un array:

# Cuando se leen los datos con fromfile() se crea un vector
# con 30 elementos porque los datos binarios se guardan de
# forma consecutiva. Para obtener el array bidimensional
# original después de leer hay que convertir el vector en un
# array 2D (10x3) con el método reshape().
```

que permiten obtener de distintos modos
números a...

Archivo

enero 2020 (1) ▼

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](#)

```
a = np.fromfile('datos.dat', dtype=np.int8)
a = a.reshape(10, 3)
print(a)

# [[4 3 4]
#  [3 2 2]
#  [3 3 1]
#  [3 1 3]
#  [3 3 1]
#  [4 1 3]
#  [4 2 3]
#  [4 1 3]
#  [3 4 3]
#  [4 4 3]]
```

Leer y escribir cadenas de texto en archivos binarios

fromfile() y write()

Leer y escribir cadenas de texto en archivos.

```
# Para escribir una cadena en un archivo binario:

cadena = b'Python para impacientes'
archivo = open('datos.dat', 'wb')
archivo.write(cadena)
archivo.close()

# Para Leer la cadena y declarar el array:

a = np.fromfile('datos.dat', count=6, dtype='S1')
print(a)

# [b'P' b'y' b't' b'h' b'o' b'n']

# Consultar otros argumentos admitidos: sep y offset
```

Leer y escribir listas en archivos binarios

pickle.load(), pickle.dump()

Leer y escribir listas Python en archivos binarios. Después de leer una lista de un archivo para convertir a array utilizar el método asarray().

```
# Escribir una lista en un archivo binario (con pickle):

import pickle
lista = [0.1, 0.2, 0.3, 0.4, 0.5]
archivo = open('datos.dat', 'wb')
pickle.dump(lista, archivo)
archivo.close()

# Leer la lista y declarar el array:

archivo = open('datos.dat', 'rb')
lista = pickle.load(archivo)
a = np.asarray(lista)
archivo.close()

print(a)

# [0.1 0.2 0.3 0.4 0.5])
```

Publicado por Pherkad en [7:46](#)



Etiquetas: [Numpy](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

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