

INDICE

- [Creare array che contengono dtype custom](#)
 - [esempio 1 : rec-arrays](#)
 - [esempio 2: rec-array da array, indici in due dimensioni](#)
 - [esempio 3: come sopra ma in 3 dimensioni](#)
 - [Lettura e scrittura di rec-array dal disco](#)
 - [esempio di tofile/fromfile](#)
 - [esempio di loadtxt](#)
 - [Per ulteriori approfondimenti su record-arrays \(tipi custom\) vedi:](#)

```
In [6]: import numpy as np
```

Il seguente non è un argomento di base ma ha a che vedere con la creazione degli array e quindi questo inciso è stato messo insieme alla lezione sulle basi.

- [HDF5](#)
-

Creare array che contengono dtype custom

Oltre che creare degli array con tipi predefiniti, posso creare anche degli array con tipi dtype custom.

Per creare un dtype uso la funzione [dtype](#)

esempio 1 : rec-arrays

voglio creare un array che contiene una serie di elementi caratterizzati da 3 campi dati:

- una data (stringa di 10 caratteri in formato AAAA-MM-GG)
- un'ora (stringa di 8 caratteri in formato hh:mm:ss)
- una misura rappresentata da un numero float64

creo un tipo custom:

```
In [ ]: from datetime import date,time

tstype = [('data', 'S10'), ('ora', 'S8'), ('misura', np.float64)]
```

creo un array che contiene elementi del tipo appena definito:

```
In [7]: ts = np.array([('2012-01-01', '00:00:00', 0.0),
 ('2012-01-02', '00:00:00', 2.0),
 ('2012-01-03', '00:00:00', 1.0),
 ('2012-01-04', '00:00:00', 2.0),
 ('2012-01-05', '00:00:00', 2.0),
 ('2012-01-05', '12:00:00', 2.0),
 ('2012-01-06', '00:00:00', 2.0),
 ('2012-01-07', '00:00:00', 2.0)], dtype=tstype)
```

```
In [8]: ts
```

```
Out[8]: array([(b'2012-01-01', b'00:00:00', 0.0),
 (b'2012-01-02', b'00:00:00', 2.0),
 (b'2012-01-03', b'00:00:00', 1.0),
```

```
(b'2012-01-04', b'00:00:00', 2.0),
(b'2012-01-05', b'00:00:00', 2.0),
(b'2012-01-05', b'12:00:00', 2.0),
(b'2012-01-06', b'00:00:00', 2.0), (b'2012-01-07', b'00:00:00', 2.0)],
dtype=[('data', '<S10'), ('ora', '<S8'), ('misura', '<f8')])
```

```
In [9]: ts.shape , ts.dtype.itemsize
```

```
Out[9]: ((8,), 26)
```

posso accedere a ciascuno dei componenti del dtype come ad un ndarray a sé

```
In [11]: ts['data']
```

```
Out[11]: array([b'2012-01-01', b'2012-01-02', b'2012-01-03', b'2012-01-04',
                b'2012-01-05', b'2012-01-05', b'2012-01-06', b'2012-01-07'],
                dtype='<S10')
```

```
In [12]: ts['ora']
```

```
Out[12]: array([b'00:00:00', b'00:00:00', b'00:00:00', b'00:00:00', b'00:00:00',
                b'12:00:00', b'00:00:00', b'00:00:00'],
                dtype='<S8')
```

```
In [13]: ts['misura']
```

```
Out[13]: array([ 0.,  2.,  1.,  2.,  2.,  2.,  2.,  2.]
```

posso accedere agli elementi dell array in modo tradizionale:

```
In [10]: ts[2]
```

```
Out[10]: (b'2012-01-03', b'00:00:00', 1.0)
```

esempio 2: rec-array da array, indici in due dimensioni

vediamo un esempio che mira ad ottenere un array che contiene come elemento una 'tupla' che rappresenta l'indice dell'elemento nell'array stesso:

uso la funzione [np.indices](#) che restituisce un array che rappresenta gli indici degli elementi di un array di shape data:

```
In [14]: nx=4
ny=5
#
i = np.indices((nx,ny))
print('ishape=',i.shape)
print('\ni[0] (irow)=')
print(i[0])
print('\ni[1] (icol)=')
print(i[1])
```

```
ishape= (2, 4, 5)
```

```
i[0] (irow)=
[[0 0 0 0]
 [1 1 1 1]
 [2 2 2 2]
 [3 3 3 3]]
```

```

i[1] (icol)=
[[0 1 2 3 4]
 [0 1 2 3 4]
 [0 1 2 3 4]
 [0 1 2 3 4]]

```

uso la funzione [np.rec.fromarrays](#) che crea un rec array a partire da array standard:

```

In [15]: #
mi=np.rec.fromarrays([i[0],i[1]],names='ix,iy')
print('\n\nmi=')
print(mi)
print('\nmi.shape=',mi.shape)
print('mi.dtype=',mi.dtype)

```

```

mi=
[[ (0, 0) (0, 1) (0, 2) (0, 3) (0, 4)]
 [ (1, 0) (1, 1) (1, 2) (1, 3) (1, 4)]
 [ (2, 0) (2, 1) (2, 2) (2, 3) (2, 4)]
 [ (3, 0) (3, 1) (3, 2) (3, 3) (3, 4)]]

mi.shape= (4, 5)
mi.dtype= [('ix', '<i4'), ('iy', '<i4')]

```

esempio 3: come sopra ma in 3 dimensioni

```

In [16]: nx=4
ny=5
nz=6
#
i = np.indices((nx,ny,nz))
print('i.shape=',i.shape)
#print('\ni[0] (irow)=')
#print(i[0])
#print('\ni[1] (icol)=')
#print(i[1])
#print('\ni[2] (icol)=')
#print(i[2])
#
mi=np.rec.fromarrays([i[0],i[1],i[2]],names='ix,iy,iz')
print('\n\nmi=')
print(mi)
print('\nmi.shape=',mi.shape)
print('mi.dtype=',mi.dtype)

```

```

i.shape= (3, 4, 5, 6)

```

```

mi=
[[[(0, 0, 0) (0, 0, 1) (0, 0, 2) (0, 0, 3) (0, 0, 4) (0, 0, 5)]
  [(0, 1, 0) (0, 1, 1) (0, 1, 2) (0, 1, 3) (0, 1, 4) (0, 1, 5)]
  [(0, 2, 0) (0, 2, 1) (0, 2, 2) (0, 2, 3) (0, 2, 4) (0, 2, 5)]
  [(0, 3, 0) (0, 3, 1) (0, 3, 2) (0, 3, 3) (0, 3, 4) (0, 3, 5)]
  [(0, 4, 0) (0, 4, 1) (0, 4, 2) (0, 4, 3) (0, 4, 4) (0, 4, 5)]]

[[[(1, 0, 0) (1, 0, 1) (1, 0, 2) (1, 0, 3) (1, 0, 4) (1, 0, 5)]
  [(1, 1, 0) (1, 1, 1) (1, 1, 2) (1, 1, 3) (1, 1, 4) (1, 1, 5)]
  [(1, 2, 0) (1, 2, 1) (1, 2, 2) (1, 2, 3) (1, 2, 4) (1, 2, 5)]
  [(1, 3, 0) (1, 3, 1) (1, 3, 2) (1, 3, 3) (1, 3, 4) (1, 3, 5)]
  [(1, 4, 0) (1, 4, 1) (1, 4, 2) (1, 4, 3) (1, 4, 4) (1, 4, 5)]]

```

```

[[ (2, 0, 0) (2, 0, 1) (2, 0, 2) (2, 0, 3) (2, 0, 4) (2, 0, 5) ]
 [ (2, 1, 0) (2, 1, 1) (2, 1, 2) (2, 1, 3) (2, 1, 4) (2, 1, 5) ]
 [ (2, 2, 0) (2, 2, 1) (2, 2, 2) (2, 2, 3) (2, 2, 4) (2, 2, 5) ]
 [ (2, 3, 0) (2, 3, 1) (2, 3, 2) (2, 3, 3) (2, 3, 4) (2, 3, 5) ]
 [ (2, 4, 0) (2, 4, 1) (2, 4, 2) (2, 4, 3) (2, 4, 4) (2, 4, 5) ]

[[ (3, 0, 0) (3, 0, 1) (3, 0, 2) (3, 0, 3) (3, 0, 4) (3, 0, 5) ]
 [ (3, 1, 0) (3, 1, 1) (3, 1, 2) (3, 1, 3) (3, 1, 4) (3, 1, 5) ]
 [ (3, 2, 0) (3, 2, 1) (3, 2, 2) (3, 2, 3) (3, 2, 4) (3, 2, 5) ]
 [ (3, 3, 0) (3, 3, 1) (3, 3, 2) (3, 3, 3) (3, 3, 4) (3, 3, 5) ]
 [ (3, 4, 0) (3, 4, 1) (3, 4, 2) (3, 4, 3) (3, 4, 4) (3, 4, 5) ]]]

```

```

mi.shape= (4, 5, 6)
mi.dtype= [('ix', '<i4'), ('iy', '<i4'), ('iz', '<i4')]

```

```
In [17]: mi['ix']
```

```

Out[17]: array([[0, 0, 0, 0, 0, 0],
                [0, 0, 0, 0, 0, 0],
                [0, 0, 0, 0, 0, 0],
                [0, 0, 0, 0, 0, 0],
                [0, 0, 0, 0, 0, 0]],

              [[1, 1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1, 1]],

              [[2, 2, 2, 2, 2, 2],
               [2, 2, 2, 2, 2, 2],
               [2, 2, 2, 2, 2, 2],
               [2, 2, 2, 2, 2, 2],
               [2, 2, 2, 2, 2, 2]],

              [[3, 3, 3, 3, 3, 3],
               [3, 3, 3, 3, 3, 3],
               [3, 3, 3, 3, 3, 3],
               [3, 3, 3, 3, 3, 3],
               [3, 3, 3, 3, 3, 3]])

```

```
In [18]: mi['iy']
```

```

Out[18]: array([[0, 0, 0, 0, 0, 0],
                [1, 1, 1, 1, 1, 1],
                [2, 2, 2, 2, 2, 2],
                [3, 3, 3, 3, 3, 3],
                [4, 4, 4, 4, 4, 4]],

              [[0, 0, 0, 0, 0, 0],
               [1, 1, 1, 1, 1, 1],
               [2, 2, 2, 2, 2, 2],
               [3, 3, 3, 3, 3, 3],
               [4, 4, 4, 4, 4, 4]],

              [[0, 0, 0, 0, 0, 0],
               [1, 1, 1, 1, 1, 1],
               [2, 2, 2, 2, 2, 2],
               [3, 3, 3, 3, 3, 3],
               [4, 4, 4, 4, 4, 4]],

              [[0, 0, 0, 0, 0, 0],
               [1, 1, 1, 1, 1, 1],
               [2, 2, 2, 2, 2, 2],
               [3, 3, 3, 3, 3, 3],
               [4, 4, 4, 4, 4, 4]]])

```

```
[1, 1, 1, 1, 1, 1],
[2, 2, 2, 2, 2, 2],
[3, 3, 3, 3, 3, 3],
[4, 4, 4, 4, 4, 4]]])
```

Lettura e scrittura di rec-array dal disco

esempio di tofile/fromfile

Prima di leggere un array da un file dobbiamo prima scrivere il file stesso [tofile](#) (ATTENZIONE ALLE NOTE)

```
In [ ]: ts.tofile('ts.bin')
```

lo leggo con fromfile specificando il tipo:

```
In [ ]: ts2 = np.fromfile('ts.bin',dtype=tstype)
```

```
In [ ]: ts2
```

```
In [ ]: (ts==ts2).any()
```

esempio di loadtxt

genero un file di testo adatto

```
In [ ]: out=open('ts.txt','wt')
        for i in range(100):
            d=datetime.date(random.randint(2000,2012),random.randint(1,12),random.randint(1,28)).isoformat()
            h=datetime.time(random.randint(0,23),random.randint(0,59),random.randint(0,59)).isoformat()
            m=random.random()*100
            out.write(d+' '+h+' '+str(m)+'\n')
        out.close()
```

lo leggo

```
In [ ]: a=np.loadtxt('ts.txt',dtype=tstype)
        a
```

Per ulteriori approfondimenti su record-arrays (tipi custom) vedi:

Per approfondimenti vedi anche

- [specifying-and-constructing-data-types](#)
- [basics.rec](#)
- [Cookbook-Recarray](#)