Tasca2A

May 27, 2021

1 IT Academy - Data Science amb Python

1.1 Tasca 2-A: Estructures d'una matriu

[1]:
```
import numpy as np
import numpy.random as random
```

[3]:
```
rng = random.default_rng()
```

1.1.1 Exercici 1

[4]:
```
arr1 = rng.integers(0, 100, 10)
arr1
```

[4]:
```
array([30, 40, 21, 38, 38, 64, 47, 44, 28, 61], dtype=int64)
```

[5]:
```
print("Nombre de dimensions:", arr1.ndim)
print("Forma de la matriu:", arr1.shape)
```

```
Nombre de dimensions: 1
Forma de la matriu: (10,)
```

1.1.2 Exercici 2

[6]:
```
arr_mean = np.mean(arr1)
print("Valor mitjà de la matriu:", arr_mean)
```

```
Valor mitjà de la matriu: 41.1
```

[7]:
```
arr1 = np.subtract(arr1, arr_mean)
arr1
```

[7]:
```
array([-11.1,  -1.1, -20.1,  -3.1,  -3.1,  22.9,   5.9,   2.9, -13.1,
        19.9])
```

1.1.3 Exercici 3

[8]:
```
arr2 = rng.integers(0, 100, 25).reshape(5, 5)
arr2
```

[8]:
```
array([[84, 29, 83, 99, 32],
       [32, 64, 88, 22, 81],
       [65, 24, 59,  8, 55],
       [37,  2, 99, 42, 45],
       [49, 62, 97,  8, 88]], dtype=int64)
```

[9]:
```
print("Valor màxim de la matriu:", np.amax(arr2))
```

```
Valor màxim de la matriu: 99
```

[10]:
```
print("Valor màxim cada columna (eix vertical):", np.amax(arr2, axis=0))
```

```
Valor màxim cada columna (eix vertical): [84 64 99 99 88]
```

[11]:
```
print("Valor màxim de cada fila (eix horitzontal):",np.amax(arr2, axis=1))
```

```
Valor màxim de cada fila (eix horitzontal): [99 88 65 99 97]
```

1.1.4 Exercici 4

Extracte de la documentació de NumPy:

> In order to broadcast, the size of the trailing axes for both arrays in an operation must either be the same size or one of them must be one.

Exemple A: Broadcasting de matrius amb eixos de la mateixa forma (3,4)

[12]:
```
arr3 = rng.integers(0, 100, 12).reshape(3, 4)
arr3
```

[12]:
```
array([[10, 23, 43, 28],
       [61, 32, 80, 13],
       [52, 34, 91, 75]], dtype=int64)
```

[13]:
```
arr4 = rng.random(12).reshape(3, 4)
arr4
```

[13]:
```
array([[0.65672579, 0.82312965, 0.47257884, 0.01718973],
       [0.25356224, 0.50219775, 0.55413272, 0.78491038],
       [0.05630364, 0.30828536, 0.37451479, 0.32795766]])
```

[14]:
```
arr3 * arr4
```

[14]:
```
array([[ 6.56725791, 18.93198199, 20.32089012,  0.48131253],
       [15.46729666, 16.07032789, 44.33061796, 10.20383498],
       [ 2.92778952, 10.48170214, 34.08084622, 24.59682451]])
```

Exemple B: Broadcasting d'una matriu 3D (3, 4, 2) amb un vector.

[15]:
```
arr5 = rng.integers(0, 100, 24).reshape(3, 4, 2)
arr5
```

[15]:
```
array([[[53, 83],
        [64, 71],
        [14, 54],
        [51, 59]],

       [[74, 32],
        [19, 21],
        [13, 22],
        [59,  8]],

       [[93, 84],
        [63, 42],
        [48, 95],
        [52, 31]]], dtype=int64)
```

[16]:
```
arr5 % 2
```

[16]:
```
array([[[1, 1],
        [0, 1],
        [0, 0],
        [1, 1]],

       [[0, 0],
        [1, 1],
        [1, 0],
        [1, 0]],

       [[1, 0],
        [1, 0],
        [0, 1],
        [0, 1]]], dtype=int64)
```

Exemple C: Broadcasting d'una matriu 2D (3, 4) amb una matriu 1D (4,)

[17]:
```
arr3
```

[17]:
```
array([[10, 23, 43, 28],
       [61, 32, 80, 13],
       [52, 34, 91, 75]], dtype=int64)
```

[18]:
```
arr6 = rng.integers(0, 100, 4)
arr6
```

[18]:
```
array([84, 92, 60, 57], dtype=int64)
```

[19]:
```
arr3 + arr6
```

[19]:
```
array([[ 94, 115, 103,  85],
       [145, 124, 140,  70],
       [136, 126, 151, 132]], dtype=int64)
```

Exemple D: Broadcasting d'una matriu 1D (4,) amb una matriu originalment 1D (3,) a la que se l'ha insertat un nou eix per forçar la compatibilitat (3,1).

[20]:
```
arr6
```

[20]:
```
array([84, 92, 60, 57], dtype=int64)
```

[21]:
```
arr7 = rng.integers(0, 100, 3)
arr7
```

[21]:
```
array([42, 56, 38], dtype=int64)
```

[22]:
```
arr7 = arr7[:, np.newaxis]
arr7
```

[22]:
```
array([[42],
       [56],
       [38]], dtype=int64)
```

[23]:
```
arr8 = arr6 + arr7
arr8
```

[23]:
```
array([[126, 134, 102,  99],
       [140, 148, 116, 113],
       [122, 130,  98,  95]], dtype=int64)
```

1.1.5 Exercici 5

[24]:
```
arr2
```

[24]:
```
array([[84, 29, 83, 99, 32],
       [32, 64, 88, 22, 81],
       [65, 24, 59,  8, 55],
       [37,  2, 99, 42, 45],
       [49, 62, 97,  8, 88]], dtype=int64)
```

[25]:
```
first_column = arr2[:, 0]
first_column
```

[25]:
```
array([84, 32, 65, 37, 49], dtype=int64)
```

[26]:
```
last_row = arr2[-1, :]
last_row
```

[26]:
```
array([49, 62, 97,  8, 88], dtype=int64)
```

[27]:
```
first_column + last_row
```

[27]:
```
array([133,  94, 162,  45, 137], dtype=int64)
```

1.1.6 Exercici 6

[28]:
```
arr2
```

[28]:
```
array([[84, 29, 83, 99, 32],
       [32, 64, 88, 22, 81],
       [65, 24, 59,  8, 55],
       [37,  2, 99, 42, 45],
       [49, 62, 97,  8, 88]], dtype=int64)
```

[29]:
```
mask = arr2 % 4 == 0
mask
```

[29]:
```
array([[ True, False, False, False,  True],
       [ True,  True,  True, False, False],
       [False,  True, False,  True, False],
       [False, False, False, False, False],
       [False, False, False,  True,  True]])
```

1.1.7 Exercici 7

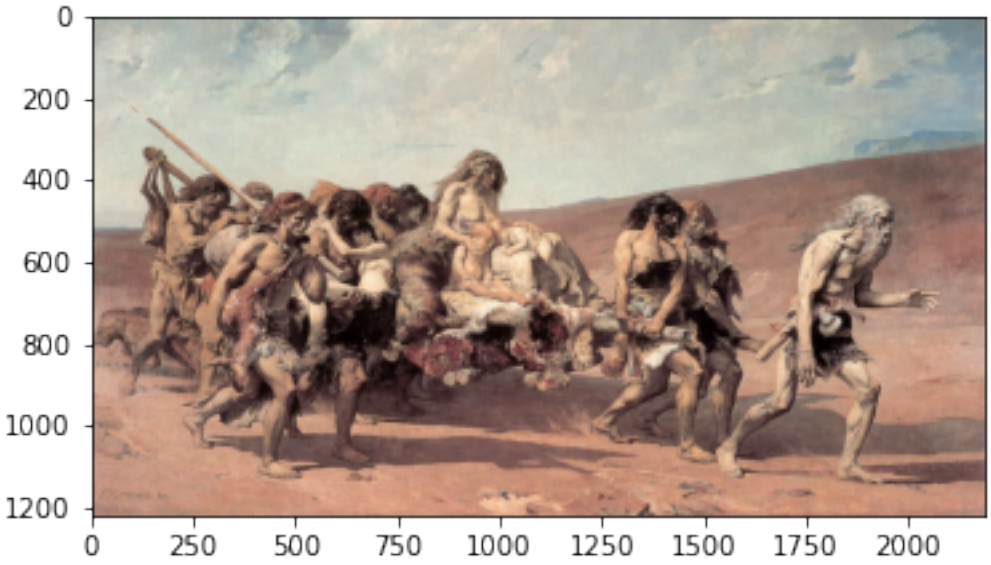
[30]:
```
arr2[mask]
```

[30]:
```
array([84, 32, 32, 64, 88, 24,  8,  8, 88], dtype=int64)
```

1.1.8 Exercici 8

[31]:
```
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
```

[32]:
```
img = mpimg.imread("Cain.jpg")
plt.imshow(img)
plt.show()
```



[33]:
```
print(img)
```

```
[[[182 191 170]
  [186 203 184]
  [171 181 170]
  …
  [123 142 149]
  [145 143 157]
  [172 185 194]]

 [[169 184 179]
  [180 192 188]
  [179 181 178]
  …
  [118 136 148]
  [127 147 158]
  [170 162 183]]

 [[178 183 186]
  [184 185 187]
  [179 183 182]
  …
  [125 141 140]
  [132 137 156]
  [163 166 185]]

 …
```

```
[[180 138 126]
 [179 145 136]
 [187 146 124]
 …
 [180 138 126]
 [169 124 119]
 [171 136 117]]

[[189 148 120]
 [179 142 133]
 [177 133 122]
 …
 [156 116 104]
 [165 127 116]
 [162 133 119]]

[[208 156 119]
 [199 152 134]
 [199 144 124]
 …
 [161 121 109]
 [164 113 112]
 [179 138 120]]]
```

[34]:
```
img.flags # img no és modificable, només de llectura
```

[34]:
```
  C_CONTIGUOUS : True
  F_CONTIGUOUS : False
  OWNDATA : False
  WRITEABLE : False
  ALIGNED : True
  WRITEBACKIFCOPY : False
  UPDATEIFCOPY : False
```

[35]:
```
img2 = np.array(img) # fem una copia de l'arxiu
img2.flags
```

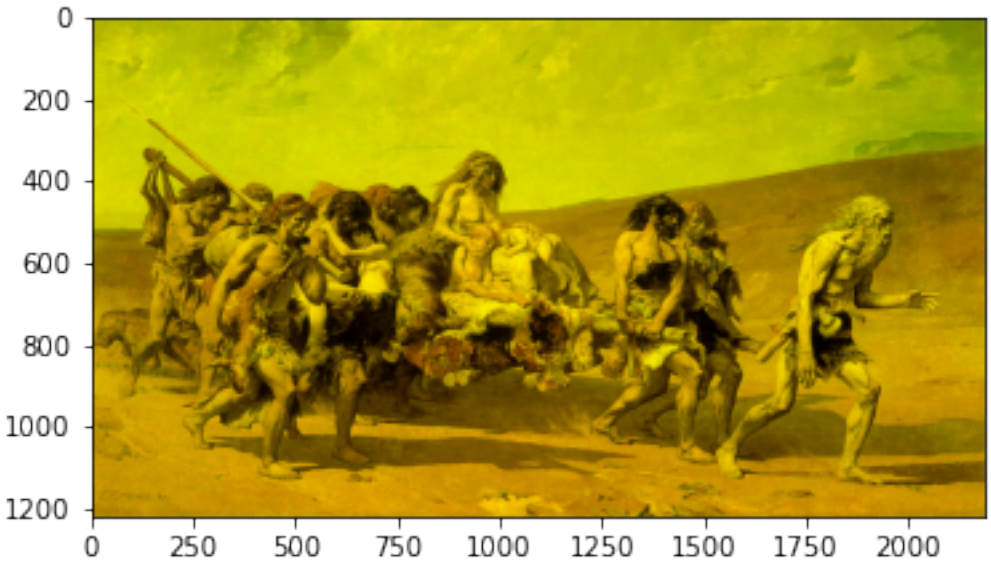
[35]:
```
  C_CONTIGUOUS : True
  F_CONTIGUOUS : False
  OWNDATA : True
  WRITEABLE : True
  ALIGNED : True
  WRITEBACKIFCOPY : False
  UPDATEIFCOPY : False
```

[36]:
```
img2[:, :, 2] = 0
print(img2)
```

```
[[[182 191   0]
  [186 203   0]
  [171 181   0]
  …
  [123 142   0]
  [145 143   0]
  [172 185   0]]

 [[169 184   0]
  [180 192   0]
  [179 181   0]
  …
  [118 136   0]
  [127 147   0]
  [170 162   0]]

 [[178 183   0]
  [184 185   0]
  [179 183   0]
  …
  [125 141   0]
  [132 137   0]
  [163 166   0]]

 …

 [[180 138   0]
  [179 145   0]
  [187 146   0]
  …
  [180 138   0]
  [169 124   0]
  [171 136   0]]

 [[189 148   0]
  [179 142   0]
  [177 133   0]
  …
  [156 116   0]
  [165 127   0]
  [162 133   0]]

 [[208 156   0]
  [199 152   0]
  [199 144   0]
  …
  [161 121   0]
  [164 113   0]
```

```
[179 138   0]]]
```

[37]:
```
plt.imshow(img2)
plt.show()
```



[38]:
```
mpimg.imsave("Cain2.jpg", img2)
```