

S03 T02: Estructura d'una Matriu

Exercici 1: Crea un np.array d'una dimensió, que inclogui l'almenys 8 nombres sencers, data type int64. Mostra la dimensió i la forma de la matriu.

```
In [37]: import numpy as np

arr = np.array([8, 3, 2, 10, 12, 6, 4, 7])

In [34]: # per comprovar que és data type int64

print(arr.dtype)

int64

In [35]: #per mostrar la dimensió de la matriu

print(arr.ndim)

1

In [36]: # per mostrar la forma de la matriu

print(arr.shape)

(8,)
```

Exercici 2: De la matriu de l'exercici 1, calcula el valor mitjà dels valors introduïts i resta la mitjana resultant de cada un dels valors de la matriu.

```
In [27]: import numpy as np

arr = np.array([8, 3, 2, 10, 12, 6, 4, 7])

In [28]: #per calcular la mitjana:

average = sum(arr)/len(arr)
print(average)

6.5

In [71]: #per restar la mitjana a cada un dels valors:

new_arr = []
for x in arr:
    x = x - average
    new_arr.append(x)
print(new_arr)

[1.5, -3.5, -4.5, 3.5, 5.5, -0.5, -2.5, 0.5]
```

Exercici 3: Crea una matriu bidimensional amb una forma de 5 x 5. Extreu el valor màxim de la matriu, i els valors màxims de cadascun dels seus eixos.

```
In [65]: import numpy as np

x = np.array([
    [1, 2, 3, 4, 5],
    [6, 7, 8, 9, 10],
    [11, 12, 13, 14, 15],
    [16, 17, 18, 19, 20],
    [21, 22, 23, 24, 25]
])

print(x.ndim)
print(x.shape)

2
(5, 5)

In [49]: # per extraure el valor màxim e la matriu:

print(a.max())

25

In [69]: # per extreure el valor màx. de cadascun dels eixos:

print(a[0].max())
print(a[1].max())
print(a[2].max())
print(a[3].max())
print(a[4].max())

5
10
15
20
25
```

Exercici 4: Mostreu-me amb exemples de diferents matrius, la regla fonamental de Broadcasting que diu : "les matrius es poden transmetre / broadcast si les seves dimensions coincideixen o si una de les matrius té una mida d'1".

```
In [80]: # En cas de matrius coincidents:

a = np.array([[1, 2], [3, 1]])
b = np.array([[2, 2], [2, 4]])
a * b

Out[80]: array([[2, 4],
               [6, 4]])

In [81]: # En cas que una de les matrius té mida 1:

c = np.array([5, 3])
a * c

Out[81]: array([[ 5,  6],
               [15,  3]])

In [82]: # En cas de matrius NO coincidents sorgeix ERROR:

d = np.array([[7, 9],[3]])
a * d

<ipython-input-82-9d752a888d3>:3: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
d = np.array([[7, 9],[3]])

Out[82]: array([[list([7, 9]), list([3, 3])],
               [list([7, 9, 7, 9, 7, 9]), list([3])]], dtype=object)
```

Exercici 5: Utilitza la Indexació per extreure els valors d'una columna i una fila de la matriu. I suma els seus valors.

```
In [140]: e = np.array([
    [3, 4, 5],
    [6, 7, 8],
    [9, 10, 11]
])

# per extreure els valors de la segona fila:

print(e[1])

[6 7 8]

In [139]: # per extreure els valors de la primera columna:

print(e[:, 0])

[3 6 9]

In [141]: # per extreure la suma dels valors anteriors:

print(e[1] + e[:, 0])

[ 9 13 17]
```

Exercici 6: Mask la matriu anterior, realitzeu un càlcul booleà vectoritzat, agafant cada element i comprovant si es divideix uniformement per quatre.

```
In [137]: # per comprovar si es divideix uniformement per 4:

f = e % 4 == 0
print(f)

[[False  True False]
 [False False  True]
 [False False False]]

In [138]: # matriu mask de la mateixa forma amb els resultats:

g = e / 4
print(g)

[[0.75 1.   1.25]
 [1.5  1.75 2.   ]
 [2.25 2.5  2.75]]
```

Exercici 7: A continuació, utilitzeu aquesta màscara per indexar a la matriu de números original. Això fa que la matriu perdi la seva forma original, reduint-la a una dimensió, però encara obteniu les dades que esteu cercant.

```
In [135]: # matriu 1-D

e[f]

Out[135]: array([4, 8])
```

Exercici 8: Carregareu qualsevol imatge (jpg, png ..) amb Matplotlib. adoneu-vos que les imatges RGB (Red, Green, Blue) són realment només amplades x alçades x 3 matrius (tres canals Vermell, Verd i Blau), una per cada color de nombres enters int8,

manipuleu aquests bytes i torneu a utilitzar Matplotlib per desar la imatge modificada un cop hàgiu acabat.

Ajuda:Importeu, import matplotlib.image as mpimg. estudeu el metode mpimg.imread()

Mostreu-me a veure que passa quan eliminem el canal G Verd o B Blau.

Mostreu-me a veure què passa quan eliminem el canal G Verd o B Blau. Hauries d'utilitzar la indexació per seleccionar el canal que voleu anul·lar.

Utilitzar el mètode, mpimg.imsave () de la llibreria importada, per guardar les imatges modificades i que haureu de pujar al vostre repositori a github.

```
In [2]: import matplotlib.pyplot as plt
import matplotlib.image as mpimg

tree = mpimg.imread('/Users/deliagonzalezmata/Downloads/tree.jpeg')
print (tree)
imgplot = plt.imshow(tree)

[[[ 53  64  30]
 [ 84  97  53]
 [ 86 102  40]
 ...
 [ 88 111  23]
 [ 98 121  33]
 [109 132  44]]

 [[ 75  86  52]
 [ 53  66  22]
 [ 61  77  15]
 ...
 [ 81 102  25]
 [ 89 110  33]
 [ 98 119  42]]

 [[ 94 105  73]
 [ 68  81  38]
 [ 69  85  23]
 ...
 [ 71  89  27]
 [ 81  99  37]
 [ 93 111  49]]

 ...

 [[ 66  55  51]
 [ 84  73  69]
 [ 82  71  67]
 ...
 [ 57  50  42]
 [ 60  53  45]
 [ 55  48  40]]

 [[ 61  50  46]
 [ 54  43  39]
 [ 56  45  41]
 ...
 [ 52  45  39]
 [ 62  55  49]
 [ 46  39  33]]

 [[ 66  55  51]
 [ 62  51  47]
 [ 70  59  55]
 ...
 [ 35  27  25]
 [ 49  41  39]
 [ 25  17  15]]]
0
25
50
75
100
125
150
175
0 50 100 150 200 250

In [3]: print(tree.ndim)
print(tree.shape)
print(tree.dtype)

3
(183, 275, 3)
uint8

In [7]: # Quan eliminem Verd:

tree_new = tree.copy()
for i in range (0, tree.shape[0]):
    for y in range (0, tree.shape[1]):
        r,g,b = tree_red[i,y]
        tree_new[i,y] = r,0,b

imgplot = plt.imshow(tree_new)

mpimg.imsave ('/Users/deliagonzalezmata/Downloads/tree_new.jpeg', tree_new)

0
25
50
75
100
125
150
175
0 50 100 150 200 250

In [8]: # Quan eliminem verd i blau:

tree_red = tree.copy()
for i in range (0, tree.shape[0]):
    for y in range (0, tree.shape[1]):
        r,g,b = tree_red[i,y]
        tree_red[i,y] = r,0,0

imgplot = plt.imshow(tree_red)

mpimg.imsave ('/Users/deliagonzalezmata/Downloads/tree_red.jpeg', tree_red)

0
25
50
75
100
125
150
175
0 50 100 150 200 250
```