

## 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 [44]: import numpy as np
```

```
In [45]: arr = np.array([3, 5, 6, 7, 8, 1, 6, 9, 2], dtype = np.int64)
```

```
In [46]: arr.ndim
```

```
Out[46]: 1
```

```
In [47]: arr.shape
```

```
Out[47]: (9,)
```

## 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 [48]: arr.mean()
```

```
Out[48]: 5.222222222222222
```

```
In [49]: np.subtract(arr, arr.mean())
```

```
Out[49]: array([-2.22222222, -0.22222222,  0.77777778,  1.77777778,  2.77777778,  
                -4.22222222,  0.77777778,  3.77777778, -3.22222222])
```

## 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 [50]: matriu = np.array([[3, 5, 6, 7, 4],  
                           [8, 1, 6, 9, 2],  
                           [2, 7, 1, 3, 4],  
                           [3, 6, 4, 2, 7],  
                           [6, 5, 3, 1, 8]])
```

```
In [51]: matriu.max()
```

```
Out[51]: 9
```

```
In [52]: matriu.max(axis = 0)
```

```
Out[52]: array([8, 7, 6, 9, 8])
```

```
In [53]: matriu.max(axis = 1)
```

```
Out[53]: array([7, 9, 7, 7, 8])
```

## 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 [54]: a = np.random.random((3, 4))
b = np.ones((3, 4))
c = np.ones((1, 4))
d = np.random.random((3, 1))
e = np.random.random((4, 4))
f = np.random.random((1, 5))
```

```
In [55]: a + b
```

```
Out[55]: array([[1.39258543, 1.24900926, 1.56618096, 1.8120609 ],
                [1.8726041 , 1.93122357, 1.94613682, 1.2115617 ],
                [1.12577811, 1.96217879, 1.90483288, 1.67603078]])
```

```
In [56]: a - d
```

```
Out[56]: array([[ 0.37251032,  0.22893415,  0.54610586,  0.7919858 ],
                [ 0.50359486,  0.56221432,  0.57712757, -0.15744755],
                [ 0.07747013,  0.9138708 ,  0.8565249 ,  0.6277228 ]])
```

```
In [57]: a * c
```

```
Out[57]: array([[0.39258543, 0.24900926, 0.56618096, 0.8120609 ],
                [0.8726041 , 0.93122357, 0.94613682, 0.2115617 ],
                [0.12577811, 0.96217879, 0.90483288, 0.67603078]])
```

```
In [58]: c / d
```

```
Out[58]: array([[49.81293444, 49.81293444, 49.81293444, 49.81293444],
                [ 2.7099592 ,  2.7099592 ,  2.7099592 ,  2.7099592 ],
                [20.70051247, 20.70051247, 20.70051247, 20.70051247]])
```

```
In [59]: a + e #donarà error perquè no tenen les mateixes dimensions.
```

ValueError

Traceback (most recent call last)

```
~\AppData\Local\Temp\ipykernel_15268\3531073975.py in <module>
----> 1 a + e #donarà error perquè no tenen les mateixes dimensions.
```

**ValueError:** operands could not be broadcast together with shapes (3,4) (4,4)

In [60]:

```
a + f #també dona error perquè tot i tenir una matriu que té un eix de
mida 1, l'altre no coincideix
```

```
-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_15268\3630075793.py in <module>
----> 1 a + f #també dona error perquè tot i tenir una matriu que té un eix de mida
      1, l'altre no coincideix
```

**ValueError:** operands could not be broadcast together with shapes (3,4) (1,5)

## Exercici 5

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

In [61]:

```
fila_1 = matriu[0, :]
print("Els valors de la fila 1 són:", fila_1)
```

Els valors de la fila 1 són: [3 5 6 7 4]

In [62]:

```
columna_2 = matriu[:, 1]
print("Els valors de la columna 2 són:", columna_2)
```

Els valors de la columna 2 són: [5 1 7 6 5]

In [63]:

```
print("La suma dels seus valors és", fila_1 + columna_2)
```

La suma dels seus valors és [ 8 6 13 13 9]

## Exercici 6

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

Això retorna una matriu de mask de la mateixa forma amb els resultats elementals del càlcul.

In [64]:

```
import numpy.ma as ma
```

In [65]:

```
mask = matriu % 4 == 0
```

In [66]:

```
mask
```

```
Out[66]: array([[False, False, False, False,  True],
               [ True, False, False, False, False],
```

```
[False, False, False, False, True],  
[False, False, True, False, False],  
[False, False, False, False, True]]])
```

## 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 [67]: matriu[mask]
```

```
Out[67]: array([4, 8, 4, 4, 8])
```

## Exercici 8

Carregareu qualsevol imatge (jpg, png ..) amb Matplotlib. adoneu-vos que les imatges RGB (Red, Green, Blue) són realment només amplades  $\times$  alçades  $\times$  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.

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

```
In [69]: img = mpimg.imread("pompompurin.jpg")
```

```
In [70]: img_edit = img[:, :, 2] + img[:, :, 0]
```

```
In [71]: mpimg.imsave("img_edit.jpg", img_edit)
```