Sprint 2: Estructura d'una Matriu

Nivel 1

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 [283…
```
import numpy as np
arr = np.array([1, 22, 13, 4,45,62,71,89], dtype = np.int64)
print(arr.dtype)
print(arr)
```

```
int64
[ 1 22 13  4 45 62 71 89]
```

Exercici 2

De la matriu de l'exercici 1, calcula el valor mitjà dels valors introduïts.

In [284…
```
print("1-D array :", arr)
print("Media de arr: ", np.mean(arr))
```

```
1-D array : [ 1 22 13  4 45 62 71 89]
Media de arr:  38.375
```

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 [285…
```
import numpy as np

arr = numpy.array([[11, 2, 3, 4, 21],
                   [4, 5, 16, 45, 23],
                   [7, 81, 22, 33, 25],
                   [17, 9, 6, 13, 75],
                   [22, 44, 33, 55, 66]])

max_element = numpy.max(arr)
min_element = numpy.min(arr)

print("El elemento máximo del array es:",  max_element)
print("El elemento mínimo del array es:", min_element)
```

```
El elemento máximo del array es: 81
El elemento mínimo del array es: 2
```

In [286…
```
elemento_maximo0 = numpy.max(arr, 0)
elemento_maximo1 = numpy.max(arr, 1)
print ("Elementos máximos por columnas: ", elemento_maximo0)
print("Elementos máximos por filas: ", elemento_maximo1)
```

```
Elementos máximos por columnas:  [22 81 33 55 75]
Elementos máximos por filas:  [21 45 81 75 66]
```

Nivell 2

Treballem els conceptes de l'estructura d'una matriu, Broadcasting, indexació, Mask..

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".

""" Broadcasting es un proceso automático implementado en las operaciones de numpy para hacer más eficientes los cálculos con arreglos multidimensionales donde, siempre que no haya ambiguedad, se procede como sigue:

1. De ser necesario, toma el arreglo de menor forma (shape) y agrega 1's por la izquierda a la forma para hacerlos compatibles.
2. De ser necesario, replica el arreglo en los ejes (axes) con forma igual a 1, hasta que alcance la forma del arreglo de mayor forma. """

In [287…

```
import numpy as np

# genera 5 números enteros aleatorios distribucion uniforme
X = np.random.randint(low=1,high=10, size=(5,5))
#Y = np.random.randint(low=1,high=10, size=(5,1))
Y= X[0:1]
i = 0 #columna que queremos obtener
Z = X[:,i]

print("X.shape =", X.shape)
print("Y.shape =", Y.shape)
print("Z.shape =", Z.shape)

print("X = (matriz de 5x5)")
print(X)

print ("Y.shape = ", Y.shape)
print("Vector fila de tamaño 5:  Y[]")
print(Y)
print("Z.shape = ",Z.shape)
print("Vector columna de tamaño 5: Z[]")
print(Z)
```

```
X.shape = (5, 5)
Y.shape = (1, 5)
Z.shape = (5,)
X = (matriz de 5x5)
[[8 5 6 3 2]
 [6 5 4 8 8]
 [7 6 7 7 4]
 [8 3 6 1 2]
```

```
 [5 3 6 3 8]]
Y.shape =  (1, 5)
Vector fila de tamaño 5:  Y[]
[[8 5 6 3 2]]
Z.shape =  (5,)
Vector columna de tamaño 5: Z[]
[8 6 7 8 5]
```

Broadcast

Calcular X+Y

X es de dimensión (5,5), e Y es de dimensión (5,1) por lo que Y se replicará por columnas y se sumará: X + [Y, Y, Y] y el resultado

será una matriz de dimensión (5,5)

In [289…

```
print("Matriz suma Y + Z :")
W = X + Y
print(W)
print("W.shape = ", W.shape)
```

```
Matriz suma Y + Z :
[[16 10 12  6  4]
 [14 10 10 11 10]
 [15 11 13 10  6]
 [16  8 12  4  4]
 [13  8 12  6 10]]
W.shape =  (5, 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 [290…

```
array = [[1, 13, 6], [9, 4, 7], [19, 16, 2]]

arr = np.array(array)

print("Imprime la matriz: ")
print(arr)

print("Imprime la fila (0): ")
Y= arr[0,:]
print (Y)

print("Imprime la columna (0): ")
Z = arr[:,0]
Z = Z[:, np.newaxis]
print (Z)

W = Y + Z

print(" Matriz suma Y + Z :")
print ("W.shape: ", W.shape)
print (W)
```

```
Imprime la matriz:
[[ 1 13  6]
 [ 9  4  7]
 [19 16  2]]
Imprime la fila (0):
```

```
[ 1 13  6]
Imprime la columna (0):
[[ 1]
 [ 9]
 [19]]
 Matriz suma Y + Z :
W.shape:  (3, 3)
[[ 2 14  7]
 [10 22 15]
 [20 32 25]]
```

Exercici 6

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

In [291…

```
array = [[1, 13, 6], [9, 4, 7], [19, 16, 2]]

arr = np.array(array)

print("Imprime la matriz: ")
print(arr)

# definimos la Mask con la condición %$ ==0
mask = (arr % 4 == 0)
print ("Imprime la Máscara: ")
print(mask)
```

```
Imprime la matriz:
[[ 1 13  6]
 [ 9  4  7]
 [19 16  2]]
Imprime la Máscara:
[[False False False]
 [False  True False]
 [False  True False]]
```

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 [292…

```
print("Matiz original")
print(arr)
print("Nueva Matriz")
nueva_arr = arr[mask]
print(nueva_arr)
```

```
Matiz original
[[ 1 13  6]
 [ 9  4  7]
 [19 16  2]]
Nueva Matriz
[ 4 16]
```

Nivell 3 :Manipulació d'imatges amb Matplotlib.

Carregareu qualsevol imatge (jpg, png ..) amb Matplotlib. adoneu-vos que les imatges RGB (Red, Green, Blue)

són realment només amplades × alçades × 3 matrius (tres canals Vermell, Verd i Blau), una per cada color de nombres enters int8,

In [293…
```
%matplotlib inline
```

In [294…
```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
```

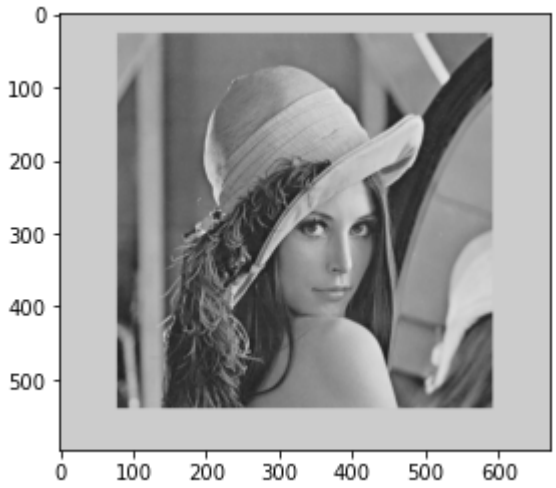
In [295…
```
img = mpimg.imread(r"C:\Users\hecto\lena.png")
print(img)
```

```
[[[0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  ...
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]]

 [[0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  ...
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]]

 [[0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  ...
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]]

 ...

 [[0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  ...
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]]

 [[0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  ...
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]]

 [[0.8 0.8 0.8]
  [0.8 0.8 0.8]
  [0.8 0.8 0.8]
  ...
```

```
[0.8 0.8 0.8]
[0.8 0.8 0.8]
[0.8 0.8 0.8]]]
```

In [296…

```
imgplot = plt.imshow(img)
```



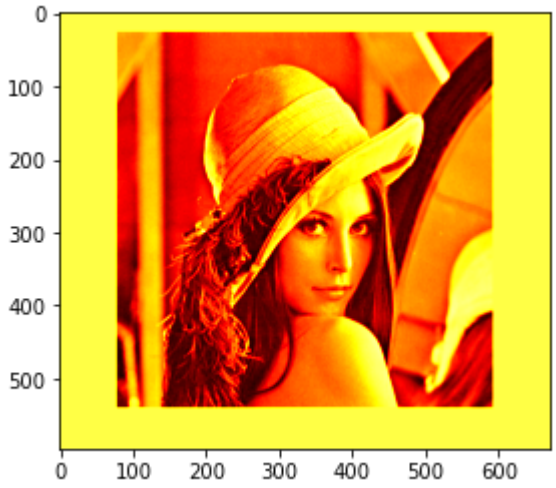
In [ ]:

In [297…

```
plt.imshow(lum_img, cmap="hot")
```

Out[297…

```
<matplotlib.image.AxesImage at 0x19a1c1dd400>
```

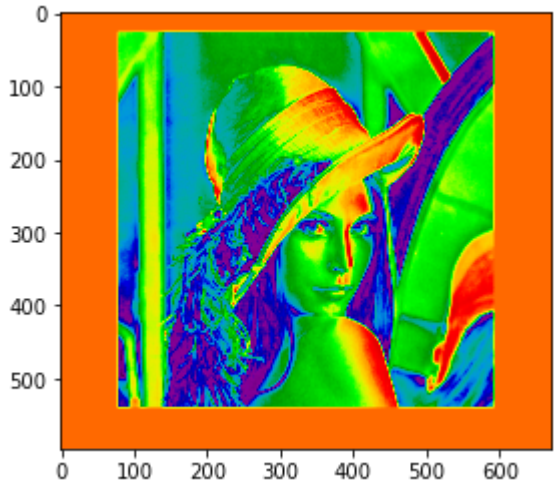


In [298…

```
## se trata de clolocar el path correcto de la imagen
```

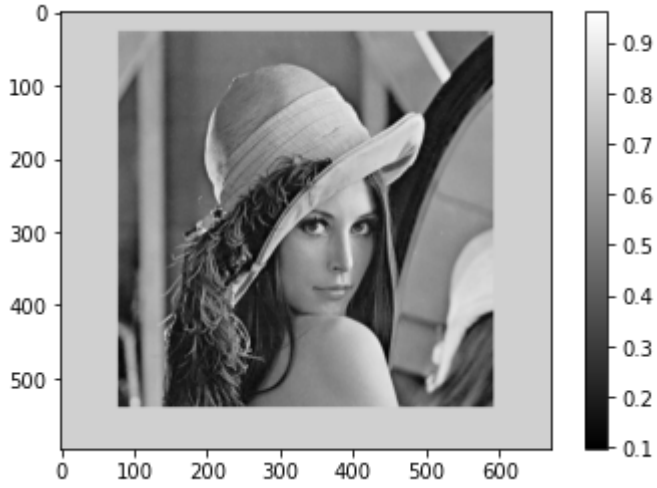
In [299…

```
imgplot = plt.imshow(lum_img)
imgplot.set_cmap('nipy_spectral')
```



In [300…
```
imgplot = plt.imshow(lum_img)
plt.colorbar()
```

Out[300…
```
<matplotlib.colorbar.Colorbar at 0x19a1c837a00>
```



Exercici 8

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

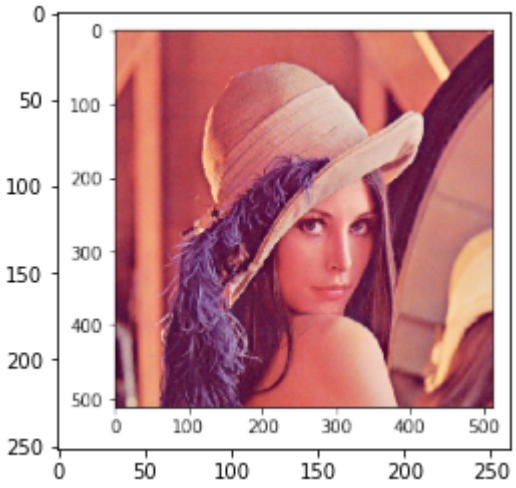
Mostreu-me a veure què passa quan eliminem el canal G Verd o B Blau. Hauries d'utilitzar la indexació per seleccionar el canal que voleu anul·lar.

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 [301…
```
img = mpimg.imread(r"C:\Users\hecto\lenacolor.png")
```

In [302…
```
plt.imshow(img)
print("Dimensiones de la imagen:")
print(img.shape)
```

```
Dimensiones de la imagen:
(252, 262, 4)
```

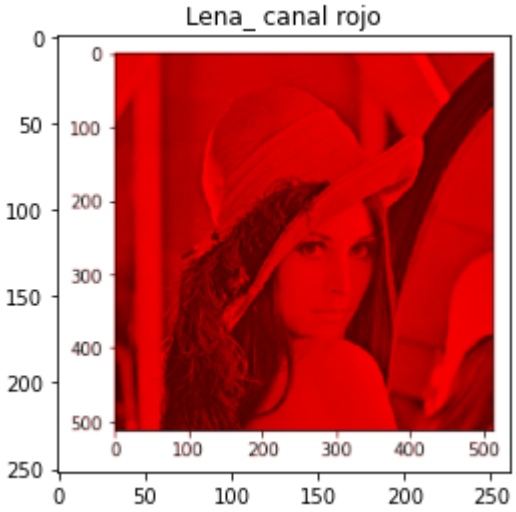


In [303…

```
lena_red=np.copy(img) #copia de la imagen para preservar la original
lena_red[:,:,1]=0
lena_red[:,:,2]=0
plt.title("Lena_ canal rojo")
plt.imshow(lena_red)
```

Out[303…

```
<matplotlib.image.AxesImage at 0x19a1dafae80>
```

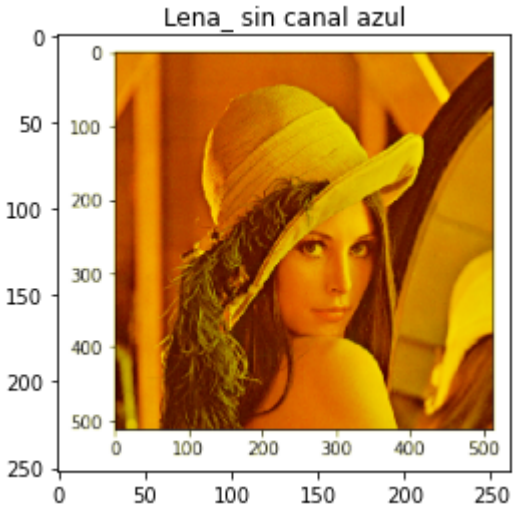


In [304…

```
lena_red_green=np.copy(img) # copia de la imagen para preservar la original
lena_red_green[:,:,2]=0
plt.title("Lena_ sin canal azul")
plt.imshow(lena_red_green)
```

Out[304…

```
<matplotlib.image.AxesImage at 0x19a1c8a8850>
```



In [305…
```
mn = lena_red_green.min()
mx = lena_red_green.max()
mx -= mn
lena_red_green = ((lena_red_green - mn)/mx) * 255
```

In [306…
```
lena_final = lena_red_green.astype(np.uint8)
```

In [307…
```
io.imsave('lena_modificada.png',lena_final)
```

In [308…
```
# se ha grabado en C:\Users\hecto\Documents\Untitled Folder\lena_modificada.png
```

In [ ]: