

In []:

```
#S03 T02: Estructura de control  
#Descripció
```

In []:

```
#Nivell 1
```

In [28]:

```
#Exercici 1  
#Crea un np.array d'una dimensió, que inclogui l'almenys 8 nombres sencers, data type i  
#nt64. Mostra la dimensió i la forma de la matriu.  
import numpy as np  
  
matriu = np.array([1, 2, 3, 4, 5, 6, 7, 8])  
print(matriu)  
print("La dimensió de la matriu és",matriu.ndim)  
print("La forma de la matriu és",matriu.shape)
```

```
[1 2 3 4 5 6 7 8]  
La dimensió de la matriu és 1  
La forma de la matriu és (8,)
```

In [67]:

```
#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.  
import numpy as np  
  
matriu = np.array([1, 2, 3, 4, 5, 6, 7, 8])  
print("El valor mitjà és",matriu.mean())  
  
mitjana = np.median(matriu, axis=0)  
print("La mitjana és",mitjana)  
  
matriu_m = np.array([])  
  
for x in matriu:  
    matriu_m = np.append(x,mitjana-x)  
    print(matriu_m)
```

```
El valor mitjà és 4.5  
La mitjana és 4.5  
[0.  3.5]  
[1.  2.5]  
[2.  1.5]  
[3.  0.5]  
[ 4. -0.5]  
[ 5. -1.5]  
[ 6. -2.5]  
[ 7. -3.5]
```

In [83]:

```
#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.

import numpy as np

matriu = 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("La dimensió de la matriu és",matriu.ndim)
max_array = np.max(matriu)
max_array_h = max(np.max(matriu, axis=0))
max_array_v = max(np.max(matriu, axis=1))
print(np.max(matriu, axis=0))
print(np.max(matriu, axis=1))

print("El valor màxim de la matriu és",np.max(matriu))
print("El valor màxim -eix horitzontal- de la matriu és",max(np.max(matriu, axis=0)))
print("El valor màxim -eix vertical- de la matriu és",max(np.max(matriu, axis=1)))
```

La dimensió de la matriu és 2

[21 22 23 24 25]

[5 10 15 20 25]

El valor màxim de la matriu és 25

El valor màxim -eix horitzontal- de la matriu és 25

El valor màxim -eix vertical- de la matriu és 25

In []:

```
#Nivell 2
```

In [10]:

```
#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".

import numpy as np

#Matrius d'una dimensió
matriu_1a = np.array([1])
matriu_1b = np.array([2])
matriu_1 = matriu_1a + matriu_1b
print("La matriu d'1 dimensió",matriu_1,"compleix la regla de Broadcasting.")

#Matrius de 2 dimensions
matriu_2a = np.array([[1,2],[3,4]])
matriu_2b = np.array([[5,6],[7,8]])
matriu_2 = matriu_2a * matriu_2b
print("La matriu de 2 dimensions",matriu_2,"compleix la regla de Broadcasting.")

#Matrius de 3 dimensions
matriu_3a = np.array([[9,8,7],[6,5,4],[3,2,1]])
matriu_3b = np.array([[6,5,4],[3,2,1],[9,8,7]])
matriu_3 = matriu_3a - matriu_3b
print("La matriu de 3 dimensions",matriu_3,"compleix la regla de Broadcasting.")

#Matrius de 4 dimensions
matriu_4a = np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]])
matriu_4b = np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]])
matriu_4 = matriu_4a / matriu_4b
print("La matriu de 4 dimensions",matriu_4,"compleix la regla de Broadcasting.")
```

La matriu d'1 dimensió [3] compleix la regla de Broadcasting.

La matriu de 2 dimensions [[5 12]
[21 32]] compleix la regla de Broadcasting.

La matriu de 3 dimensions [[3 3 3]
[3 3 3]
[-6 -6 -6]] compleix la regla de Broadcasting.

[[1. 1. 1. 1.]
[1. 1. 1. 1.]
[1. 1. 1. 1.]
[1. 1. 1. 1.]]

La matriu de 4 dimensions [[1. 1. 1. 1.]
[1. 1. 1. 1.]
[1. 1. 1. 1.]
[1. 1. 1. 1.]] compleix la regla de Broadcasting.

In [52]:

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

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

columna = matriu[:,0]
fila = matriu[2]
matriu_r = []

for i in columna:
    for j in fila:
        matriu_r.append(i+j)

print(matriu_r)
```

[12, 11, 10, 9, 8, 7, 6, 5, 4]

In [17]:

```
#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 c
àlcul.

import numpy as np
import numpy.ma as ma

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

matriu_mask = []

count = 0
for i in matriu:
    fila = matriu[count]
    count += 1
    for j in fila:
        if (j % 4 == 0):
            matriu_mask.append(True)
        else:
            matriu_mask.append(False)
print(matriu_mask)
```

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

In [16]:

```
#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 en  
cara obteniu les dades que esteu cercant.  
  
#Si tens la màscara la matriu mask i la matriu a arr1 simplement és fer arr1[mask]  
  
import numpy as np  
import numpy.ma as ma  
  
arr1 = np.array([[9,8,7],  
                [6,5,4],  
                [3,2,1]])  
  
mask = np.array([[False, True, False],  
                [False, False, True],  
                [False, False, False]])  
  
arr1[mask]
```

Out[16]:

```
array([8, 4])
```

In []:

```
#Nivell 3
```

In [48]:

```
#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 × alçada
s × 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()

#Exercici 8

import os
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

dir = os.getcwd()
image = mpimg.imread('ironman.jpg')
plt.imshow(image)
plt.show()

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

image_v = np.copy(image)

canal_r=image_v[:, :, 0]
canal_g=image_v[:, :, 1]
canal_b=image_v[:, :, 2]

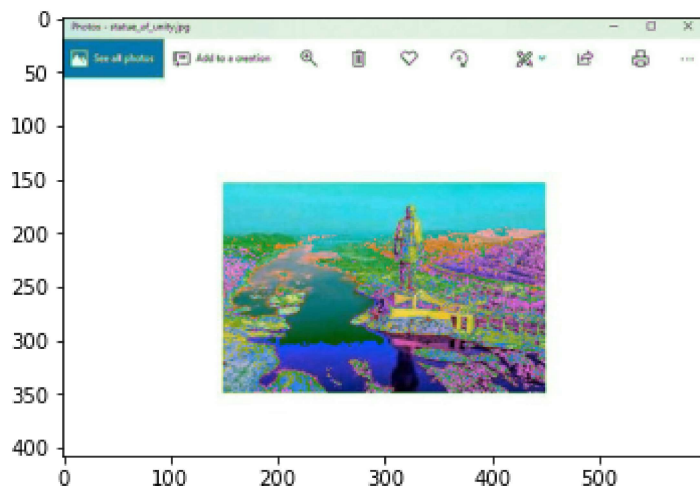
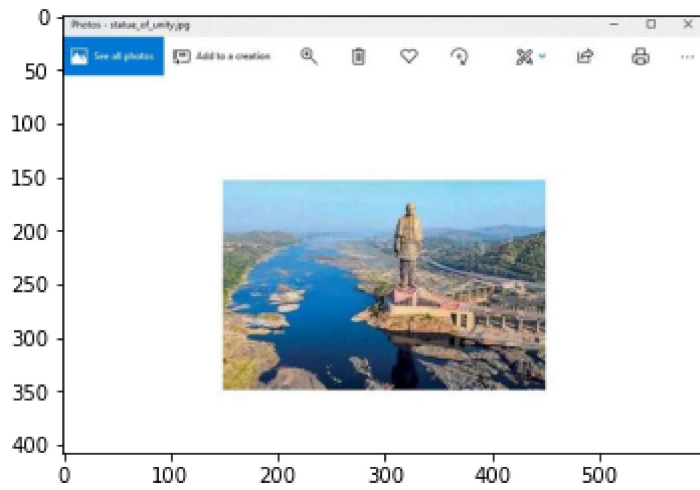
canal_rb=np.copy(image_v)
canal_rb[:, :, 0]=canal_rb[:, :, 0] + canal_r
canal_rb[:, :, 2]=canal_rb[:, :, 2] + canal_b

plt.figure(3)
plt.axis('on')
plt.imshow(canal_rb)
plt.show()

#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.
#canal G Verd
#canal B Blau

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

mpimg.imsave(os.path.join(dir, "ironman_v.jpg"), canal_rb)
```



In []:

In []:

In []:

In []: