

# Pràctica amb programació numèrica

Descripció:

- Familiaritza't amb la Programació Numèrica a través de la llibreria NumPy.

## NIVELL 1

### Exercici 1

Crea una funció que donat un Array d'una dimensió, et faci un resum estadístic bàsic de les dades. Si detecta que l'array té més d'una dimensió, ha de mostrar un missatge d'error.

In [18]:

```
import numpy as np

array1 = np.random.randint(100, size=(20))
#array1 = np.random.randint(100, size=(20,20)) #activar aquesta línia i desactivar l'antiga

print(array1)
print(array1.ndim)

if array1.ndim >= 2:
    print("L'array d'aquest exercici ha de ser de ser d'una dimensió")
else:
    print("El número més petit de la matriu és el", array1.min())
    print("El número més gran de la matriu és el", array1.max())
    print("La suma acumulativa de tots els elements dona", array1.cumsum())
    print("La mitjana és", array1.mean())
    print("La mediana és", np.median(array1))
    print("El coeficient de correlació és", np.corrcoef(array1))
    print("La desviació estàndar és", np.std(array1))
```

```
[71 89 19 30 73 18 74 32 47  0  4 87 25 80 62 96  1 74 24 25]
1
El número més petit de la matriu és el 0
El número més gran de la matriu és el 96
La suma acumulativa de tots els elements dona [ 71 160 179 209 282 300 374 406 453 453 457
544 569 649 711 807 808 882
 906 931]
La mitjana és 46.55
La mediana és 39.5
El coeficient de correlació és 1.0
La desviació estàndar és 31.30091851687423
```

### Exercici 2

Crea una funció que et generi un quadrat NxN de nombres aleatoris entre el 0 i el 100.

In [53]:

```
n=0
m=0
contador=0

while contador >= 0:
    if n <=0:
        print("Tria un número enter positiu")
        n = int(input())
        m = n
```

```

    contador += contador
else:
    quadrat = np.random.randint(100, size=(n,m))
    break
print(quadrat)

```

```

Tria un número enter positiu
10
[[49 39 92 66 66 18  5 45 12 10]
 [31 39 17 14  0  2 78 13 57 92]
 [36 50 83 63 17 76 70 57 72 39]
 [24 40 94 38 12 75 73 26 39 95]
 [75 47 98 56 55 59 74 97  3 59]
 [86 88 47 86 83 22 55 42 94 41]
 [64 85 40 33 90  2 66 74  9 99]
 [15 28 62 64 11 70 74 25 45 14]
 [27 65 14 31 22  9 91 84 83 21]
 [93 30  7  0 50 34 31 51 11 31]]

```

## Exercici 3

Crea una funció que donada una taula de dues dimensions, et calculi els totals per fila i els totals per columna.

In [57]:

```

n=0
m=0
contador=0

while contador >= 0:
    if n <=0 or m<=0:
        print("Tria un número enter positiu per l'eix de les files")
        n = int(input())
        print("Tria un número enter positiu per l'eix de les columnes")
        m = int(input())
        contador += contador
    else:
        quadrat = np.random.randint(100, size=(n,m))
        break
print(quadrat)

print("Suma de les columnes", quadrat.sum(axis=0))      # suma total per columna
print("Suma de les files", quadrat.sum(axis=1))        # suma total per fila

```

```

Tria un número enter positiu per l'eix de les files
4
Tria un número enter positiu per l'eix de les columnes
2
[[83 65]
 [32 75]
 [18 36]
 [40 13]]
Suma de les columnes [173 189]
Suma de les files [148 107  54  53]

```

## Exercici 4

Implementa manualment una funció que calculi el coeficient de correlació. Informa't-en sobre els seus usos i interpretació.

In [ ]:

```

# https://stackabuse.com/calculating-pearson-correlation-coefficient-in-python-with-numpy/,
# https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/

```

