

In [11]: *# 1. Tamaño grande: dada una lista, escriba una función que cambie todos los números positivos de la lista a "big"*
Ejemplo: biggie_size ([- 1, 3, 5, -5]) devuelve la misma lista, pero cuyos valores son ahora [-1, "big", "big"]

```
def big(arr):  
    for i in range(len(arr)):  
        if arr[i]>0:  
            arr[i]="big"  
    return arr
```

```
print(biggie_size([-1,3,5,-5]))
```

```
[-1, 'big', 'big', -5]
```

In [19]: *# 2. Contar positivos : dada una lista de números, cree una función para reemplazar el último valor con el número*
positivos. (Tenga en cuenta que cero no se considera un número positivo).
Ejemplo: count_positives([- 1, 1, 1,1]) cambia la lista original a [-1, 1, 1, 3] y la devuelve
Ejemplo: count_positives([1, 6, -4, -2, -7, -2]) cambia la lista a [1, 6, -4, -2, -7, 2] y la devuelve

```
def contarPositivos(arr):  
    contador=0  
    for i in range(len(arr)):  
        if arr[i]>0:  
            contador+=1  
  
    arr[len(arr)-1]=contador  
    return arr
```

```
print(countarPositivos([-1,1,1,1]))
```

```
[-1, 1, 1, 3]
```

In [21]: *# 3. Suma total : crea una función que toma una lista y devuelve la suma de todos los valores de la matriz.*
Ejemplo: sum_total ([1,2,3,4]) debería devolver 10
Ejemplo: sum_total ([6,3, -2]) debería devolver 7

```
def sumaTotal(arr):  
    suma=0  
    for i in range(len(arr)):  
        suma+=arr[i]  
    return suma  
  
print(sumaTotal([1,2,3,4]))  
print(sumaTotal([6,3,-2]))
```

10
7

In [25]: *# 4. Promedio : crea una función que toma una lista y devuelve el promedio de todos los valores.*
Ejemplo: el promedio ([1,2,3,4]) debería devolver 2.5

```
def promedio(arr):  
    suma=0  
    for i in range(len(arr)):  
        suma+=arr[i]  
    return suma/len(arr)  
  
print(promedio([1,2,3,4]))
```

2.5

In [27]: *# 5. Longitud : crea una función que toma una lista y devuelve la longitud de la lista.*
Ejemplo: la longitud ([37,2,1, -9]) debería devolver 4
Ejemplo: longitud ([]) debería devolver 0

```
def longitud(arr):  
    return len(arr)  
  
print(longitud([37,2,1,-9]))  
print(longitud([]))
```

4
0

In [46]: *# 6. Mínimo : crea una función que tome una lista de números y devuelva el valor mínimo en la lista. Si la lista
haga que la función devuelva False.
Ejemplo: mínimo ([37,2,1, -9]) debería devolver -9
Ejemplo: mínimo ([]) debería devolver False*

```
def valorMinimo(arr):  
    minimo=0  
    if len(arr)==0:  
        return False  
    else:  
        for i in range (len(arr)):  
            if minimo>arr[i]:  
                minimo=arr[i]  
        return minimo  
  
print(valorMinimo([37,2,1,-9]))  
print(valorMinimo([]))
```

-9
False

In [48]: *# 7. Máximo : crea una función que toma una lista y devuelve el valor máximo en la matriz. Si la lista está vacía
función devuelva False.
Ejemplo: máximo ([37,2,1, -9]) debería devolver 37
Ejemplo: máximo ([]) debería devolver False*

```
def valorMaximo(arr):  
    if len(arr)==0:  
        return False  
    maximo=arr[0]  
    for i in range (len(arr)):  
        if maximo<arr[i]:  
            maximo=arr[i]  
    return maximo  
  
print(valorMaximo([37,2,1,-9]))  
print(valorMaximo([]))
```

37
False

```
In [ ]: # 8. Análisis final : crea una función que tome una lista y devuelva un diccionario que tenga la suma total, pro  
# máximo y longitud de la lista.  
# Ejemplo: ultimate_analysis ([37,2,1, -9]) debería devolver {'total': 31, 'promedio': 7.75, 'minimo': -9, 'maxi  
# 'longitud': 4}
```

In [57]:

```
def analisisFinal(arr):  
    diccionario = {}  
  
    diccionario["total"]=sumaTotal(arr)  
    diccionario["promedio"]=promedio(arr)  
    diccionario["minimo"]=valorMinimo(arr)  
    diccionario["maximo"]=valorMaximo(arr)  
    diccionario["longitud"]=longitud(arr)  
  
    return diccionario  
  
print(analisisFinal([37,2,1, -9]))  
{'total': 31, 'promedio': 7.75, 'minimo': -9, 'maximo': 37, 'longitud': 4}
```

```
In [59]: def analisisFinal(arr):
    diccionario = {'total': 0, 'promedio': 0, 'minimo': 0, 'maximo': 0, 'longitud': 0}
    diccionario['longitud'] = len(arr)
    sum = 0
    min = arr[0]
    max = arr[0]
    for i in range(len(arr)):
        sum = sum + arr[i]
        if arr[i] < min:
            min = arr[i]
        if arr[i] > max:
            max = arr[i]
    diccionario['total'] = sum
    diccionario['promedio'] = sum/len(arr)
    diccionario['minimo'] = min
    diccionario['maximo'] = max
    return diccionario

print(analisisFinal([37,2,1, -9]))
```

```
{'total': 31, 'promedio': 7.75, 'minimo': -9, 'maximo': 37, 'longitud': 4}
```

In [64]: *# 9. Lista inversa : crea una función que tome una lista y la devuelva con los valores invertidos. Haz esto sin
segunda lista. (Se sabe que este desafío aparece durante las entrevistas técnicas básicas).
Ejemplo: reverse_list ([37,2,1, -9]) debería devolver [-9,1,2,37]*

```
def listaInversa(arr):
    for i in range (len(arr)//2):
        variable=arr[i]
        arr[i]=arr[(len(arr)-i-1)]
        arr[len(arr)-i-1]=variable
    return arr

print(listaInversa([37,2,1,-9,20,3,6]))
```

```
[6, 3, 20, -9, 1, 2, 37]
```

```
In [63]: def invertir(lista):  
    for i in range(len(lista)//2):  
        variable = lista[i] # Se guarda Los primeros elementos de La lista  
        lista[i] = lista[len(lista)-i-1] # 4-0-1 #Se reemplaza Los primeros elementos por Los últimos  
        lista[len(lista)-i-1] = variable #Se reemplaza Los últimos elementos por Los primeros  
    return lista  
  
print(invertir([37,2,1, -9,20,3,6]))  
  
[6, 3, 20, -9, 1, 2, 37]
```

In []:

In []: