

```
In [1]: #Ejercicio 1
# Pasar a escala de grises el color codificado en los elementos de la lista `pixel`

pixel= [0.6,0.3,0.4] # intensidades de cada canal.
#El elemento 0 es el R, el 1 el G y el 2 el B

# La intensidad en escala de grises es el promedio de la intensidad de cada canal R, G
intensidad=0 # IMPLEMENTAR
intensidad=(sum(pixel)/3)
print("La intensidad es:")
print("{0:.2f}".format(intensidad))
```

La intensidad es:
0.43

```
In [10]: #Ejercicio 2
# Pasar a blanco y negro el valor de intensidad codificado en la variable intensidad

# podemos considerar que un pixel se convierte en blanco si su intensidad en escala de
# y negro de lo contrario
print("En blanco y negro el pixel sería: (0 -> negro, 1 -> blanco)")

# blanco=1
# negro=0

bw = 0 # IMPLEMENTAR
if(intensidad > 0.5):
    bw=1
else:
    bw
print(bw)
```

En blanco y negro el pixel sería: (0 -> negro, 1 -> blanco)
0

```
In [12]: #Ejercicio 3: Escribir un for para buscar el máximo de la lista e imprimirlo
lista=[44,11,15,29,53,12,30]
maximo=0
# IMPLEMENTAR
for i in lista:
    if (i == max(lista)):
        maximo=i
    else:
        None

# debe imprimir 53
print("-El maximo es:")
print(maximo)
```

El maximo es:
53

```
In [27]: #Ejercicio 4: Escribir un for para buscar el mínimo elemento de la lista e imprimir su

lista=[44,11,15,29,53,12,30]
posicion=0
# IMPLEMENTAR
for i, j in enumerate(lista):
    if(j == min(lista)):
        posicion=i
```

```

    else:
        None
#debe imprimir 1
print("La posición del minimo es:")
print(posicion)

```

La posición del minimo es:
1

```

In [28]: # Ejercicio 5 : Ordenar la Lista de forma asendente
#IMPLEMENTAR
numeros = [10, 5, 15, 8, 100, 7, 40, 35, 65, 79]
numeros.sort()
print(numeros)

```

[5, 7, 8, 10, 15, 35, 40, 65, 79, 100]

```

In [3]: #Ejercicio 6

# Crear una funcion en donde me permita enviar como parametro el numero de elementos y
# devolver un listado de la serie fibonassi con el numero de elementos ingresado.

#IMPLEMENTAR

def fibonacci(num):
    inicial_1 = 0
    inicial_2 = 1
    lista = []
    if(num > 0):
        lista.append(inicial_1)
        lista.append(inicial_2)
        while num > len(lista):
            f = lista[-1]+lista[-2]
            lista.append(f)
        return lista
    else:
        print("Ingresa un numero mayor a 0")

#Funcion
fibonacci(20)

```

```

Out[3]: [0,
1,
1,
2,
3,
5,
8,
13,
21,
34,
55,
89,
144,
233,
377,
610,
987,
1597,
2584,
4181]

```

```

In [20]: #Ejercicio 7

```

```
# Escribir una función que reciba una lista y un valor,
#y devuelva la cantidad de veces que aparece ese valor en la lista
l=[1,4,2,3,5,1,4,2,3,6,1,7,1,3,5,1,1,5,3,2]
v=2

def ocurrencias(lista,valor):
    # IMPLEMENTAR
    cont=0
    for i in lista:
        if(valor==i):
            cont +=1
    return cont

print("La cantidad de ocurrencias es:")
print(ocurrencias(l,v))

#debe imprimir 3, la cantidad de veces que aparece el 2 en la lista

#debe imprimir 3, la cantidad de veces que aparece el 2 en la lista
```

La cantidad de ocurrencias es:
3

```
In [3]: #Ejercicio 8

# Crear un metodo de validacion de cedula Ecuatoriana, en caso de que la cedula no sea
# una excepcion, ademas de controlar que solo pueda ingresar digitos numericos por tecl

#IMPLEMENTAR

def validar_cedula(num):
    if(num != "" and len(num) == 10):
        ult_dig = num[9:]
        num = num[:9]
        l = []
        lista_pro = [2, 1, 2, 1, 2, 1, 2, 1, 2]
        for i, j in zip(num, lista_pro):

            mul = int(i) * j

            if(mul > 9):
                mul = mul - 9
                l.append(mul)
            else:
                l.append(mul)

        res = sum(l)

        mod = res % 10

        dig_ve = 10 - mod
        if (dig_ve == int(ult_dig)) or (mod == 0):
            return "cedula correcta"
        else:
            return "Cedula incorrecta"
    else:
        return "INGRESE MINIMO 10 DIGITOS"

validar_cedula(input())
```

0302295910

Out[3]: 'cedula correcta'

```
In [1]: #Ejercicio 9 : Investigar las funciones que se pueden utilizar con listas, diccionarios

lista = ["Patricia", 1.55, 30, "Cañar", True, "SYSTEM", "MOV", False, 100, 6.7]
print(lista)

#funcion append(para añadir un elemento)
a = "UPS"
lista.append(a)
print(lista)

#funcion extend(permite agregar elementos dentro de una lista
#pero a diferencia de append al momento de agregar una lista,
#cada elemento de esta lista se agrega como un elemento más dentro de la otra lista)
sub_lista = ["GIT", "IA-SE"]
lista.extend(sub_lista)
print("*****")
print(sub_lista)
print(lista)

['Patricia', 1.55, 30, 'Cañar', True, 'SYSTEM', 'MOV', False, 100, 6.7]
['Patricia', 1.55, 30, 'Cañar', True, 'SYSTEM', 'MOV', False, 100, 6.7, 'UPS']
*****
['GIT', 'IA-SE']
['Patricia', 1.55, 30, 'Cañar', True, 'SYSTEM', 'MOV', False, 100, 6.7, 'UPS', 'GIT', 'IA-SE']
```

In []:

In []: