**CICLOS EN PYTHON**

WHILE

El ciclo while en Python cuenta con un **else** que se ejecuta cuando NO se cumple la condición

contador = 0

while contador < 11:

    print(contador)

    contador += 1

else:

    print('fin del ciclo while')

este código daría como resultado un listado de los números del 0 al 10 y finalmente 'fin del ciclo while’

FOR

Las cadenas o string en Python son tratadas como arrays de caracteres, por tanto, si se hace un **foreach** a un string, se divide en cada carácter el string

Al igual que el WHILE, FOR cuenta con un else que se ejecuta cuando finaliza el ciclo.

BREAK

La instrucción **break** se usa para interrumpir el flujo de un bucle, generalmente cuando se cumple una condición

Este es un ejemplo:

palabra = "RARISIMOJO"

for letra in palabra:

    if(letra == 'O'):

        break

    print(letra)

print('Continua el programa')

este programa imprime cada letra de la variable **palabra** pero se detiene al encontrarse con una O

el break interrumple el flujo del ciclo for, pero no el del programa en general. Si se quisiese lograr dicho resultado, sería necesario utilizar en lugar de break la instrucción **exit(1)**

Cabe resaltar que este **break** tambien detiene las operaciones por defecto en el bloque else correspondiente al for

CONTINUE

La palabra continue funciona como break, solo que no finaliza el bloque de código entero.

min = int(input('introduce el primer numero del rango '))

max = int(input('introduce el segundo numero del rango '))

while(min < max):

    if(min % 2 == 0):

        print(min)

        min += 1

        continue

    min += 1

else:

    print('fin de la iteracion')

en este código el continue hace que se finalice el bloque y “se vuelva a comprobar la condición”