















































# Bucles o Ciclos

# Estructuras de control iterativas

Las Estructuras de Control while (mientras) y for (para) nos van a permitir repetir una instrucción o conjunto de instrucciones mientras una expresión booleana resulte verdadera. Se las suele denominar Bucle While y Bucle For con el objetivo de destacar su naturaleza iterativa.

## For

El bucle for nos permitirá establecer cuántas veces queremos o tenemos que iterar, sus componentes son:

* Expresión inicial
* Condición
* Incremento o paso

En pseudocódigo: Para i = 1 Hasta i <= 10 Con Paso 1 Hacer

En Python con todos sus componentes:

**for** expresion **in** range(valor\_inicial,condicion,paso):

instrucciones

Para poder usar este bucle solo es necesario poner la condición de paro, por defecto el valor\_inicial = 0 y el paso = 1

*# siempre la condicion es menor que: i < condición*

**for** i **in** range(condicion):

instrucciones

*# si el paso es -1 es mayor que: i>condición*

**for** i **in** range(valor\_inicial,condicion, **-**1):

instrucciones

En Python, ésta Estructura de Control Iterativa se caracteriza por iterar para cada uno de los elementos de una lista (o Tupla) sin que sea necesario modificar ninguna condición dentro del conjunto de instrucciones que dicho bucle contiene. Es decir, que nos mantendremos dentro del bucle siempre y cuando la lista siga teniendo elementos.

Este bucle será de gran utilidad cuando iteremos sobre las filas y columnas de matrices.

**for** i **in** range(n):

instrucciones

In [ ]:

*#Ejemplo, i tomara valores de 0 a 9, el valor incia es 0 por defecto*

*# i tomara valores menores a 10*

**for** i **in** range(10):

print(i)

In [ ]:

*# Ejemplo con todos sus componentes*

**for** i **in** range(6,20,2):

print(i)

In [ ]:

*# Ejemplo con el paso -1*

**for** i **in** range(10,0,**-**1):

print(i)

## While

Esta Estructura de Control Iterativa es muy útil cuando se requiere que un algoritmo se mantenga iterando una cantidad de n-veces hasta que deje de ser verdadera una cierta expresión booleana.

**while**(condicion):

instrucciones

In [ ]:

*# Ejemplo*

continuar **=** 1

**while**(continuar):

continuar **=** int(input("Introduce 1 o 0 "))

print("Hola mundo")

In [ ]:

continuar **=** "y"

**while**(continuar **==** "y"):

continuar **=** input("Quieres continuar y/n:")

print("Hola mundo")

In [ ]:

cont **=** 0

**while**(cont**<**10):

print("Hola mundo")

cont**+=**1

## Do while

Se usa cuando el problema a resolver requiere que se ejecute por lo menos una vez el ciclo. Se compone por una condicional cuya expresión se evalúa después de ejecutar el bloque de instrucciones.

Para crear un bucle do while en Python, necesitas modificar el bucle while un con el propósito de obtener un comportamiento similar al de los bucles do while en otros lenguajes.

**while**(**True**):

instrucciones

**if** condicion:

**break**

In [ ]:

*# Ejemplo*

palabra\_secreta **=** "python"

**while**(**True**):

dame\_palabra **=** input("Ingresa la palabra secreta: ")

**if** dame\_palabra **==** palabra\_secreta:

**break**

print("Palabra secreta: incorrecta")

print("Palabra secreta: correcta")

In [ ]:

*# Ejemplo menú*

**while**(**True**):

print("1.- opción 1")

print("2.- opción 2")

print("3.- opción 3")

opt **=** int(input("Ingresa una opción: "))

**if** 0 **<** opt **and** opt **<** 4:

**break**

**else**:

print("Error")

print(f"elegiste la opción: {opt}")

# Estructuras de control condicionales

Es la ejecución condicional de una instrucción o conjunto de instrucciones de acuerdo al valor que adopte una expresión booleana. Su implementación se logra con las denominadas Estructuras de Control Condicionales.

## Simples

Ejecuta un conjunto de líneas de código si es cierta una expresión booleana.

**if**(condicion):

instrucciones

In [ ]:

*# Ejemplo*

print("¡Hola mundo!")

print("¡Hola mundo!")

print("¡Hola mundo!")

binario **=** int(input("Ingresa 1 o 0"))

**if**(binario):

print("¡Hola mundo desde if")

print("¡Hola mundo desde if")

print("¡Hola mundo desde if")

print("¡Hola mundo!")

print("¡Hola mundo!")

print("¡Adios mundo!")

In [ ]:

*# Ejemplo con operador relacional*

print("¡Hola mundo!")

print("¡Hola mundo!")

print("¡Hola mundo!")

pregunta **=** input("¿Quieres pasar por if? y/n")

**if**(pregunta **==** "y"): *# operador relacional*

print("¡Hola mundo desde if")

print("¡Hola mundo desde if")

print("¡Hola mundo desde if")

print("¡Hola mundo!")

print("¡Hola mundo!")

print("¡Adios mundo!")

In [ ]:

*# Ejemplo con operador relacional*

print("¡Hola mundo!")

print("¡Hola mundo!")

print("¡Hola mundo!")

pregunta **=** input("¿Quieres pasar por if? y/n")

pregunta2**=** input("¿Estas seguro? y/n")

**if**(pregunta **==** "y" **and** pregunta2 **==** "y"): *# operador relacional*

print("¡Hola mundo desde if")

print("¡Hola mundo desde if")

print("¡Hola mundo desde if")

print("¡Hola mundo!")

print("¡Hola mundo!")

print("¡Adios mundo!")

## Bicondicional

También podemos incorporar un segundo conjunto de instrucciones cuya ejecución se produzca cuando la expresión booleana resulte falsa. Para ello se usa la instrucción else:

**if**(condicion):

instrucciones

**else**:

instruccines

In [ ]:

*# Ejemplo*

print("automovil: en marcha")

print("automovil: en marcha")

print("automovil: en marcha")

print("automovil: en marcha")

semaforo **=** input("Semaforo esta en verde y/n")

**if**(semaforo **==** "y"):

print("automovil: continua...")

**else**:

print("automovil: detenido")

print("termina")

## Múltiple

Es común que querramos contar con más de dos puntos de salida de nuestra Estructura de Control. Para ello, se intercala la instrucción elif -abreviatura de else if- que funciona de modo similar a else pero evalúa también una expresión booleana como lo hace if:

**if**(condicion):

instrucciones

**elif**(condicion2):

instrucciones

**.**

**.**

**elif**(condicionN):

instrucciones

**else**:

instrucciones

In [ ]:

dia **=** int(input("Ingresa un número del 1 al 7"))

**if**(dia**==**1):

print("Lunes")

**elif**(dia**==**2):

print("Martes")

**elif**(dia**==**3):

print("Miercoles")

**elif**(dia**==**4):

print("Jueves")

**elif**(dia**==**5):

print("Viernes")

**elif**(dia**==**6):

print("Sábadp")

**elif**(dia**==**7):

print("Domingo")

**else**:

print("Error")