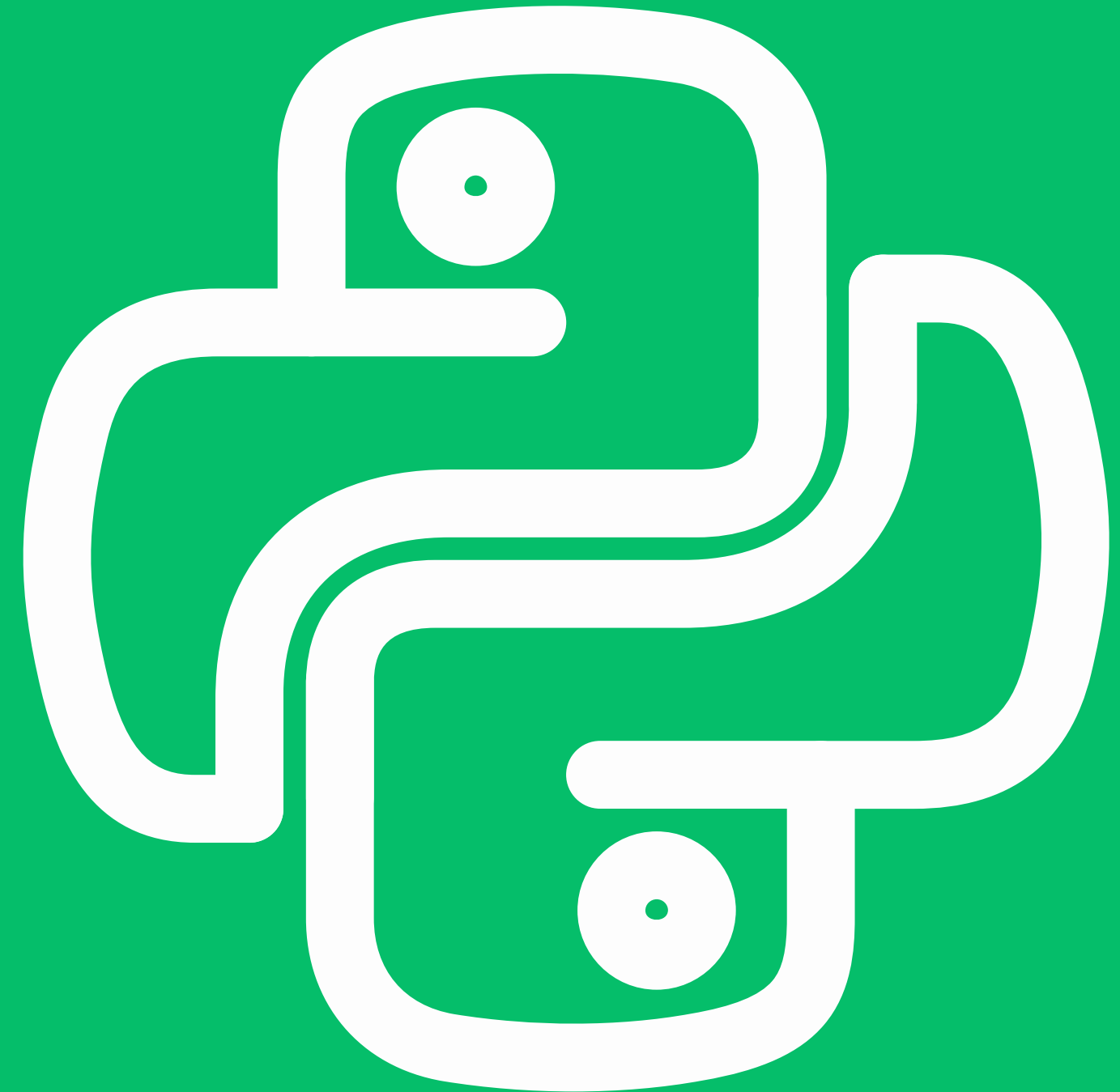


# Estructuras de Control



INSTITUTO DE INNOVACIÓN Y  
TECNOLOGÍA APLICADA

# Contenido

**01** Introducción

**02** E.C Selectivas

**03** E.C Selectivas  
Multiples

**04** E.C Repetitivas con  
while

**05** E.C Repetitivas con  
for

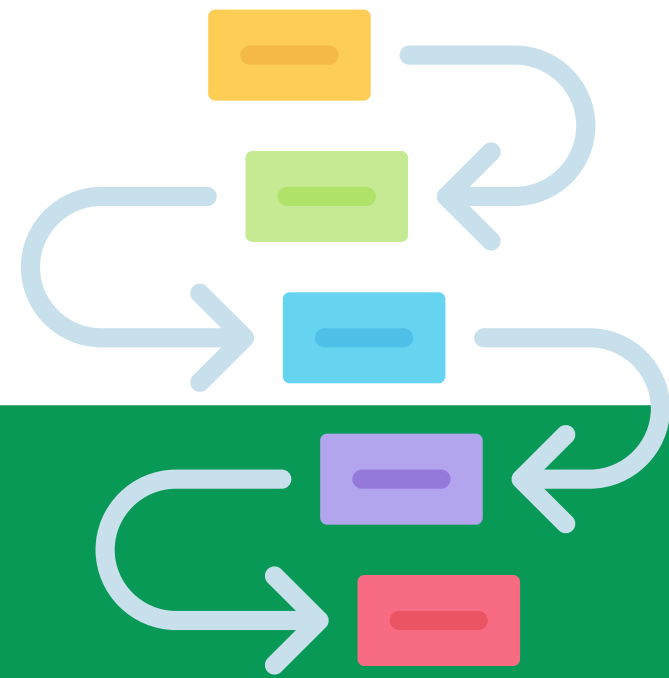
## INTRODUCCIÓN

# ¿Estructuras de Control?



- Las estructuras de control son como las "reglas" que le damos a la computadora para decidir qué hacer o cuándo repetir algo.
- En Python, las estructuras de control definen el flujo de ejecución de las tareas, determinando si un conjunto de instrucciones se ejecuta o se repite.

# Tipos de estructuras de control



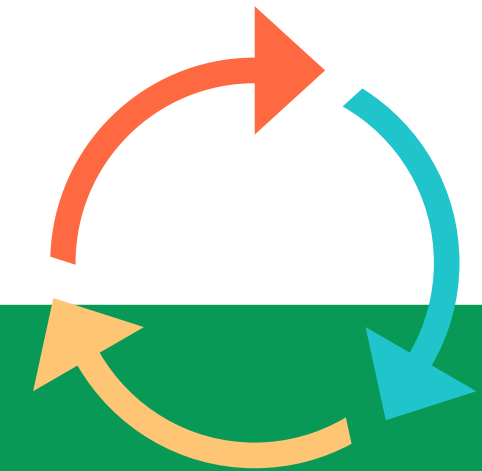
### Estructura Secuencial

La estructura secuencial es la más simple. La computadora **ejecuta las instrucciones una tras otra**, en el orden en que están escritas, como si siguieras una lista de tareas.



### Estructura Selectiva (Condicional)

La estructura selectiva o condicional permite que la computadora tome una decisión. Dependiendo de si una condición se cumple o no, hace una cosa u otra. Es como un "**si esto ocurre, haz esto; si no, haz otra cosa**".



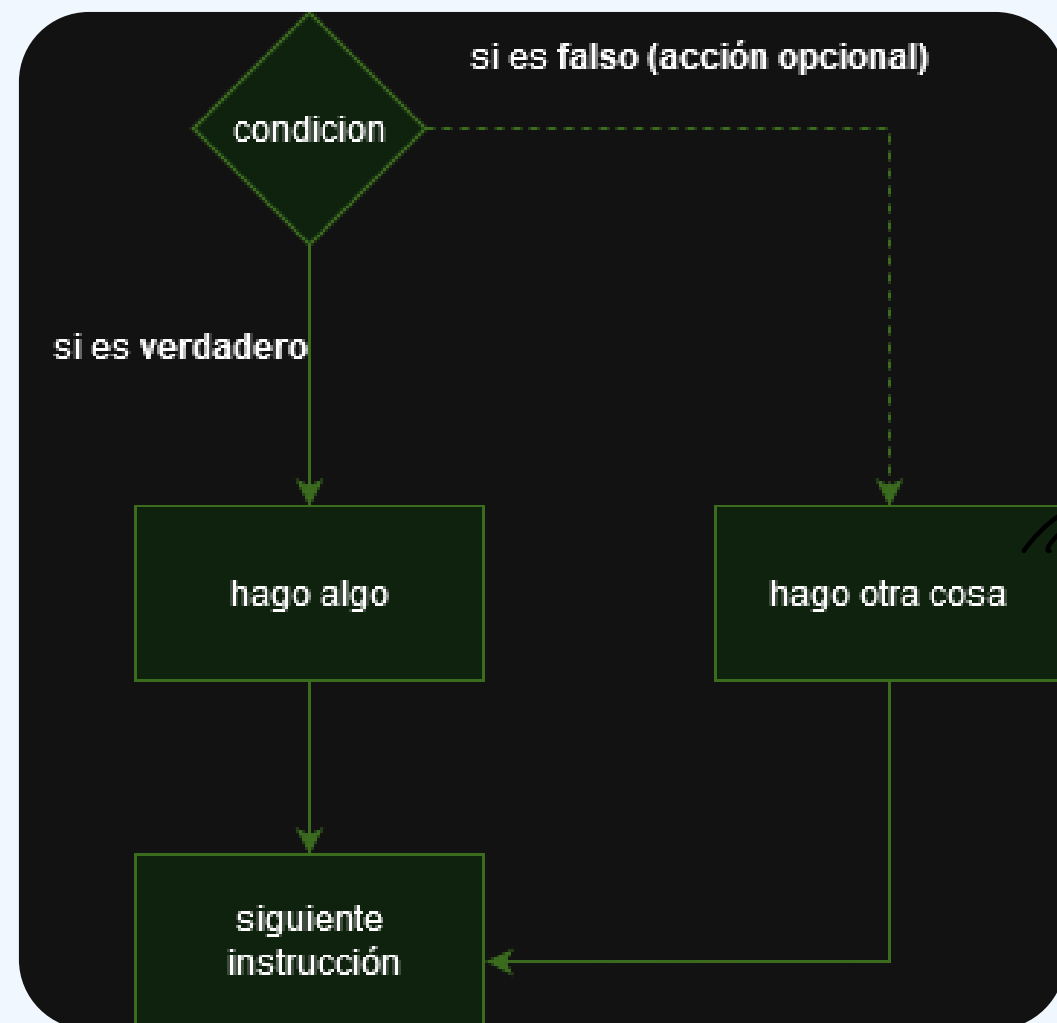
### Estructura Iterativa (Repetitiva o Bucles)

La estructura iterativa permite que la computadora **repita una acción varias veces**. Es útil cuando quieres hacer lo mismo muchas veces, por ejemplo, contar números o repetir un mensaje.



## ¿Qué es una estructura selectiva simple?

La estructura selectiva simple se usa cuando queremos que la computadora **elija entre dos opciones**. Es como decir: "**si ocurre esto, haz esto; si no ocurre, haz otra cosa**".



```
if condicion: # SI CONDICION ENTONCES:  
    # Ingrese aqui logica de IF => BLOQUE DE CODIGO IF  
else:  
    # ingrese aqui logica de ELSE => BLOQUE DE CODIGO ELSE
```

graficamente

en python



P Y T H O N

**iProblemos!**

## ¿Qué es una estructura selectiva multiple?

La estructura selectiva múltiple se usa cuando tenemos más de dos posibles opciones o caminos a seguir. elif significa "**si no se cumple lo anterior, pero sí esto otro**". Se pueden tener varios elif y una última opción con else si ninguna condición es verdadera.



```
if condicion 0:  
    # haz esto  
elif condicion 1:  
    # no se cumplio la condicion 0 pero si la condicion 1, entonces haz esto  
elif condicion 2:  
    # no se cumplio la condicion 1 pero si la condicion 2, entonces haz esto  
.  
.  
.  
else:  
    # si nada de lo anterior se cumple haz ESTO
```

en python



P Y T H O N

**iProblemos!**



## A PRACTICAR

01

Solicitar al usuario un número de cliente. Si el número es el 1000, imprimir "Ganaste un premio"

02

Solicitar al usuario que ingrese dos números y mostrar cuál de los dos es menor. No considerar el caso en que ambos números son iguales.

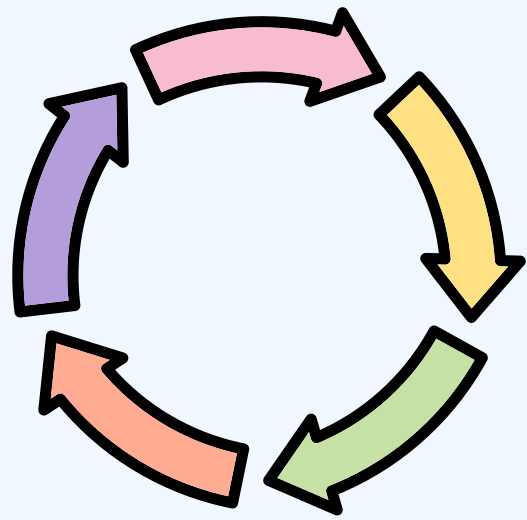
03

Solicitar al usuario que ingrese dos números y mostrar cuál de los dos es menor. Esta vez considerar el caso en que ambos números son iguales

## Ayuda Memoria

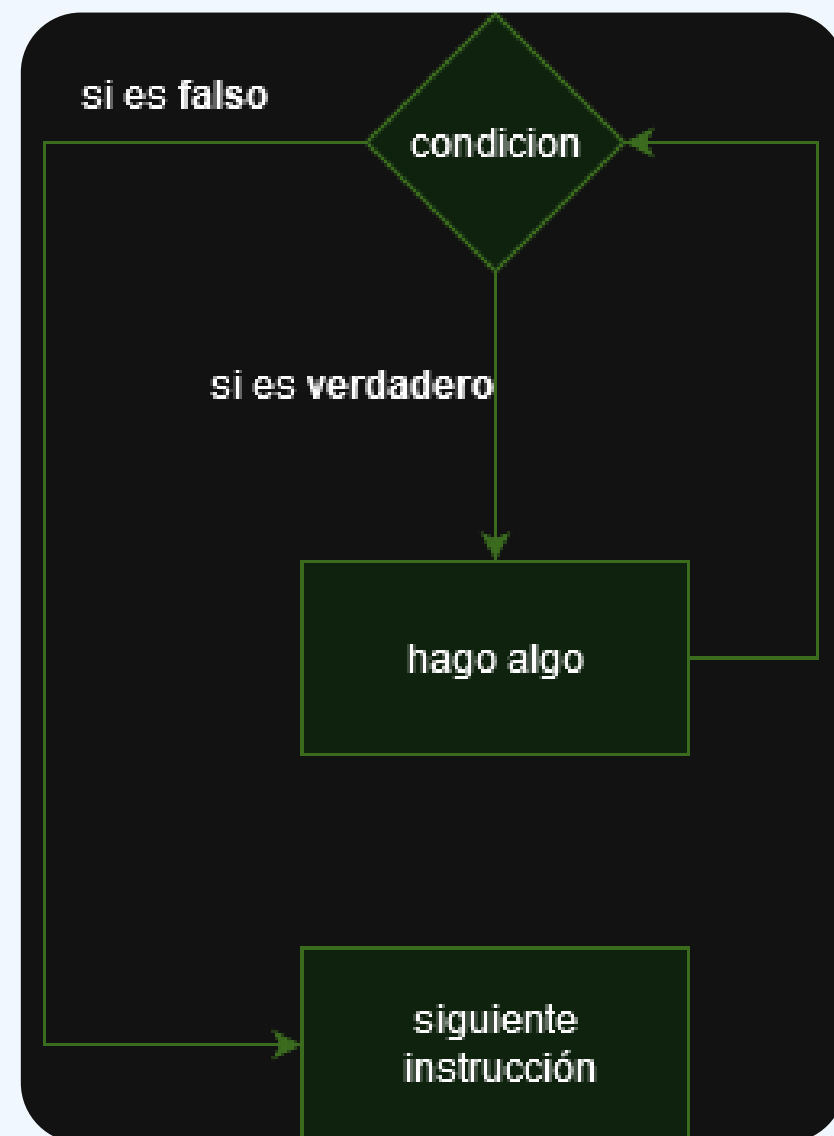
```
if condicion: # SI CONDICION ENTONCES:  
    # Ingrese aqui logica de IF => BLOQUE DE CODIGO IF  
else:  
    # ingrese aqui logica de ELSE => BLOQUE DE CODIGO ELSE
```

```
if condicion 0:  
    # haz esto  
elif condicion 1:  
    # no se cumplio la condicion 0 pero si la condicion 1, entonces haz esto  
elif condicion 2:  
    # no se cumplio la condicion 1 pero si la condicion 2, entonces haz esto  
.  
.  
.  
else:  
    # si nada de lo anterior se cumple haz ESTO
```



## ¿Qué es una estructura repetitiva?

La estructura iterativa permite que la computadora repita una acción varias veces. Es útil cuando quieres hacer lo mismo muchas veces, por ejemplo, contar números o repetir un mensaje.



graficamente

```
while condicion:  
    # hace algo
```

en python

## Tipos de estructuras iterativas



### Sin condicion de corte (Infinito)

Una estructura iterativa infinita es aquella que **nunca se detiene**, porque no tiene una condición que diga cuándo parar. Esto puede ser útil en algunos casos, pero es peligroso si no se controla, ya que el programa se quedará repitiendo la acción para siempre.



### Con condicion de corte al inicio

En esta estructura, la computadora **verifica la condición al principio del ciclo**. Si la condición es verdadera, repite las instrucciones. Si es falsa, se detiene.



### Con condicion de corte al final

En este caso, la computadora **siempre ejecuta las instrucciones al menos una vez** y luego, **al final de cada ciclo, verifica si debe seguir o no**. Esto se puede simular en Python usando un ciclo while con una condición inicial que permita al menos una ejecución.



P Y T H O N

**iProblemos!**

## A PRACTICAR

**01** Cree un bucle que muestre los números del 100 al 200.

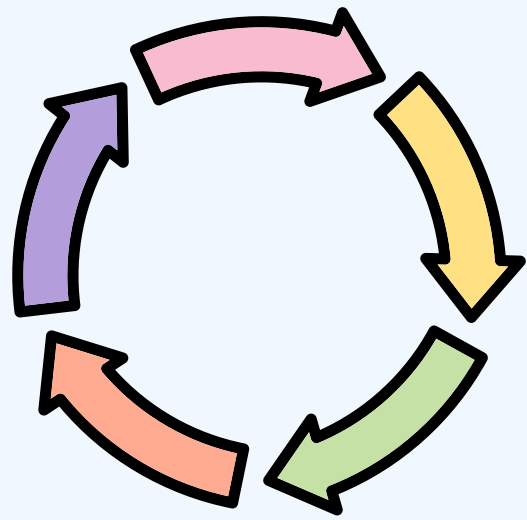
**02** Crear un ciclo infinito, es decir, un ciclo que nunca termina (las instrucciones dentro del ciclo quedan a criterio de cada uno)

**03** Escriba un programa que pregunte al usuario si quiere finalizar la ejecución del programa. Si este escribe "sí" el programa finaliza

## Ayuda Memoria

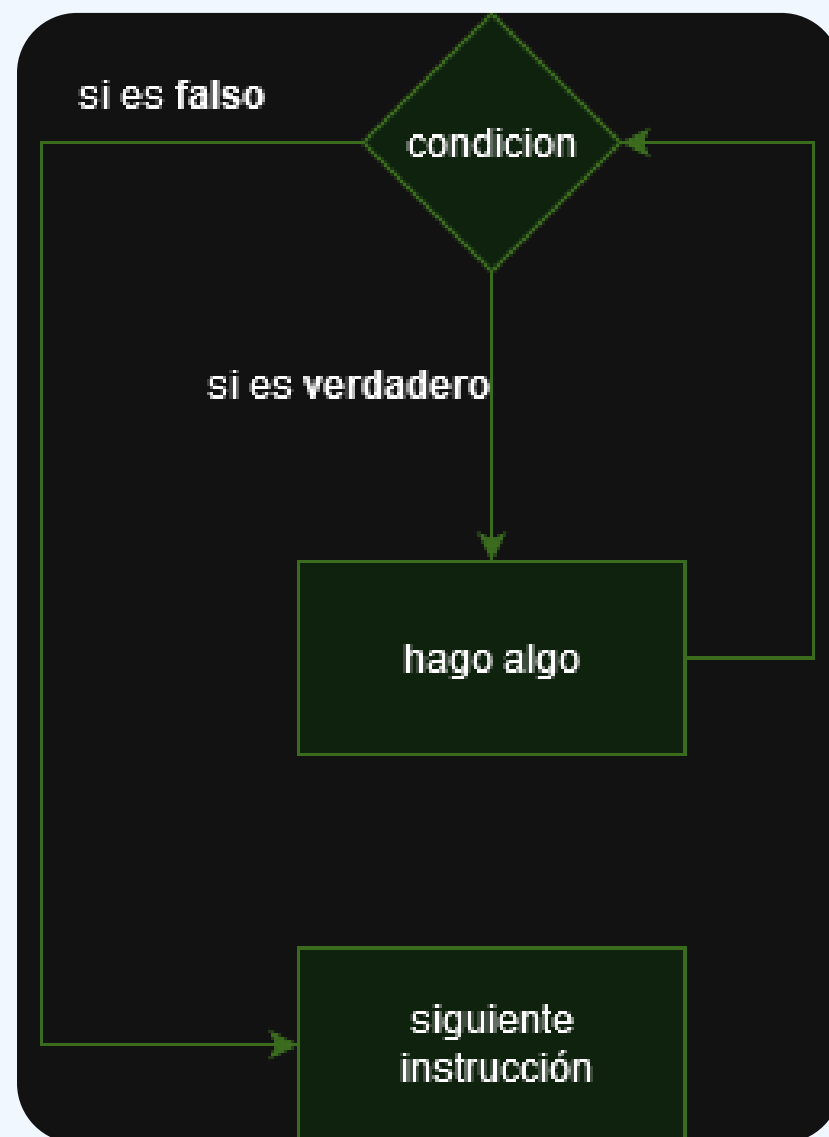
```
while condicion:  
    # hace algo
```

```
while True:  
    # hace algo  
    if condicion:  
        break
```



## ¿Qué es una estructura repetitiva?

La estructura iterativa permite que la computadora repita una acción varias veces. Es útil cuando quieres hacer lo mismo muchas veces, por ejemplo, contar números o repetir un mensaje.



graficamente

```
for [variable] in range([limite superior]):  
    # logica del for
```

```
for [variable] in range([limite inferior],[limite superior],[paso]):  
    # logica del for
```

en python

01

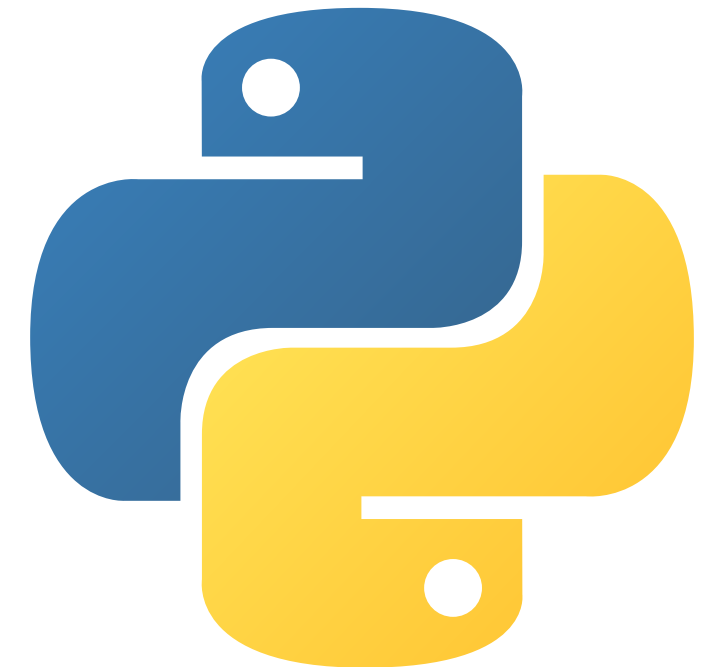
Pide al usuario un numero. Muestra su tabla de multiplicar del 1 al 10 utilizando el ciclo 'for'.

02

Cree un bucle que sume los números del 100 al 200

## Ayuda Memoria

```
for [variable] in range([limite superior]):  
    # logica del for
```



```
for [variable] in range([limite_Inferior],[limite_Superior],[Salto]):  
    #Logica del for
```

I I T A

# ¡Muchas Gracias!

Por su atención

