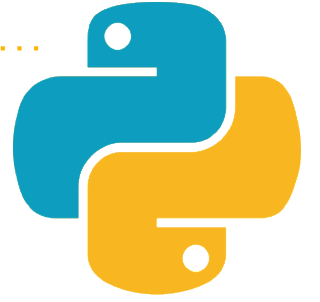


# Unidad No.2 Lenguajes Altamente Dinámicos/débilmente Tipados

TI2011 – Introducción a la Programación



# Ciclos en Python

Python posee herramientas para poder ejecutar el mismo código un determinado número de veces.  
Esto se hace a través de:

- ✓ **while** loops
- ✓ **for** loops

# While Loop



# While loops

Se evalúa al "entrar" al ciclo, es decir, si la operación lógica se cumple, se ejecuta el ciclo por primera vez.

Si la operación lógica es falsa, entonces el ciclo termina.

Si la operación lógica se cumple, o es verdadera, el código se repite.



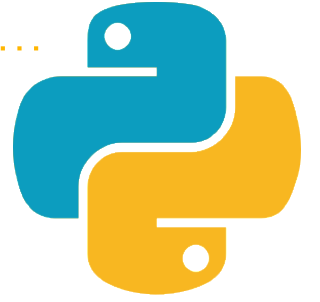
# While loops

La estructura general de un while es la siguiente:

```
while operación lógica:  
    comando 1
```

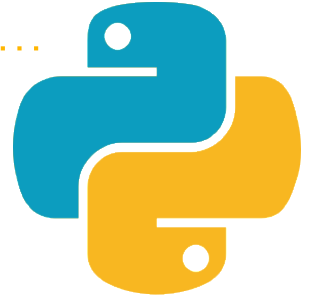
Ejemplo:

```
i = 1  
while i < 6:  
    print(i)  
    i += 1  
print(6, 'no es mayor que 6')
```



Ejemplo While con else:

```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print(i , " no es mayor que 6")
```



Ejemplo 2:

```
while True:  
    print("Hola \n")
```

1

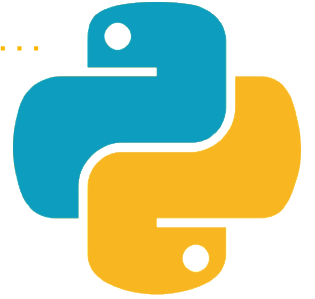
**¿Se va a ejecutar el ciclo?**

**Sí**, porque la operación lógica es verdadera

2

**¿Cuántas veces se ejecutará?**

**Infinitas veces**, porque la operación lógica siempre será verdadera



Ejemplo 3:

```
while False:  
    print("Hola \n")
```

1

**¿Se va a ejecutar el ciclo?**

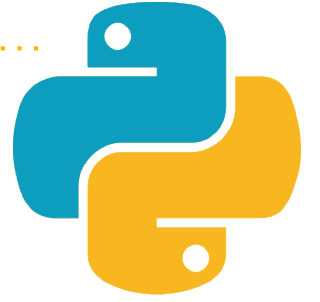
**No**, porque la operación lógica es verdadera

2

**¿Cuántas veces se ejecutará?**

**0 veces**, ya que la operación lógica es falsa y nunca pasará a ser verdadera

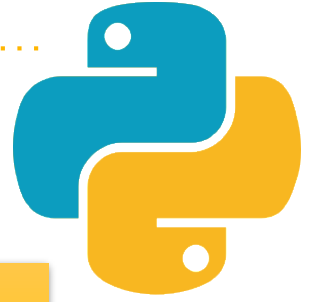




Ejemplo 4:

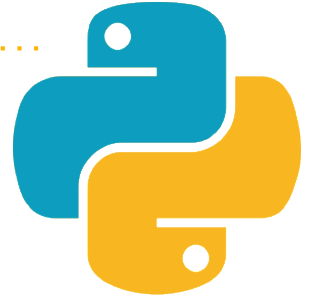
```
contador = 5
while contador != 0:
    print("Dentro del ciclo: ", contador)
    contador -= 1
print("Fuera del ciclo", contador)
```

```
contador = 5
while contador != 0:
    print("Dentro del ciclo: ", contador)
    contador -= 1
print("Fuera del ciclo", contador)
```



**Sentencia break:** interrumpe o cortar los ciclos

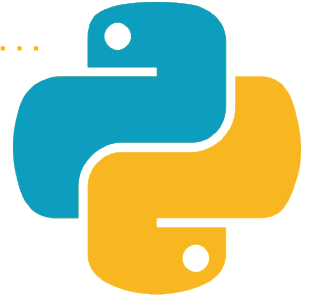
```
var=10
while var>0:
    print ("Valor de variable:",var)
    var-=1
    if var == 6:
        break
```



**Sentencia continue:** hace que pase de nuevo al principio del bucle aunque no se haya terminado de ejecutar el ciclo anterior

```
var=10
while var>0:
    var-=1
    if var == 6:
        continue
    print ("Valor de variable:",var)
```

## Ejercicios:



- a. Diseña un programa en Python que reciba un número mayor que 10 e imprima todos los números pares entre 0 y el número.
- b. Diseña un programa que imprima 5 veces la palabra “hola”, debe considerar un salto de línea por cada palabra mostrada.
- c. Diseñe un programa en Python que reciba dos números a y b, para luego saber cual es el mayor. Sin embargo los números deben ser mayores a 0. El programa saldrá si al menos un número ingresado es menor a 0.

## Ejercicios:



- d. Diseñe un programa en Python que cuente los números pares e impares ingresados por un usuario de forma no definida, (se pueden ingresar n cantidad de números para evaluar).  
Para finalizar el programa y obtener el resultado deberá presionar la letra n, para seguir ingresando números debe presionar cualquier tecla.
- e. Al programa anterior (letra d) añadir la posibilidad de que el algoritmo entregue el resultado de la suma de los pares e impares ingresados y la totalidad de los ciclos ejecutados.