



# ESTRUCTURAS DE CONTROL ITERATIVAS

CURSO PYTHON  
CILSA

# Estructuras de control iterativas o bucles

A diferencia de las estructuras de control condicionales, las **iterativas** (también llamadas cíclicas o bucles), nos permiten ejecutar un mismo código, de **manera repetida**, mientras se cumpla una **condición**.

En Python se dispone de dos estructuras cíclicas:

- El bucle **while**.
- El bucle **for**.

# Bucle while

Este bucle, se encarga de ejecutar una misma acción “mientras que” una determinada condición se cumpla:

```
anio = 2020
```

```
while anio <= 2022:
```

```
    print("Informes del Año", str(anio)) #Salida: los años desde el 2020 al 2022
```

```
    anio += 1
```

Podrás notar que, en **cada iteración, incrementamos el valor de la variable que condiciona el bucle** (anio).

Si no lo hiciéramos, esta variable siempre sería igual a 2020 y el bucle se ejecutará de **forma infinita**, ya que la condición (anio <= 2022) siempre se estaría cumpliendo.

# Bucle while con estructura condicional

Pero ¿Qué sucede si el valor que condiciona la iteración **no es numérico y no puede incrementarse**?

En ese caso, podremos utilizar una **estructura de control condicional**, anidada dentro del bucle, y frenar la ejecución cuando el condicional deje de cumplirse, con la palabra clave reservada **break**:

```
while True:
```

```
    nombre = input("Indique su nombre: ")
```

```
    if nombre:
```

```
        break
```

El bucle anterior, incluye un condicional anidado que verifica si la variable nombre es verdadera (sólo será verdadera si el usuario tipea un texto en pantalla cuando el nombre le es solicitado). Si es verdadera, el bucle para (break). Sino, seguirá ejecutándose hasta que el usuario, ingrese un texto en pantalla.

# Bucle for

El bucle **for**, en Python, es aquel que nos permitirá iterar sobre una **variable compleja**, del tipo lista o tupla:

Por cada nombre en *mi\_lista*, imprimir *nombre*:

```
mi_lista = ['Juan', 'Antonio', 'Pedro', 'Herminio']  
for nombre in mi_lista:  
    print(nombre) #Salida: Juan Antonio Pedro Herminio
```

Por cada color en *mi\_tupla*, imprimir *color*:

```
mi_tupla = ('rosa', 'verde', 'celeste', 'amarillo')  
for color in mi_tupla:  
    print(color) #Salida: rosa verde celeste amarillo
```

En los ejemplos anteriores, **nombre** y **color**, son dos variables declaradas en tiempo de ejecución (es decir, se declaran dinámicamente durante el bucle), asumiendo como valor, el de cada **elemento** de la lista (o tupla) en cada iteración.

# Bucle for emulando while

Otra forma de iterar con el bucle for, puede emular a while:

Por cada año en el rango 2020 a 2022, imprimir la frase “Informes del Año año”

```
for anio in range(2020, 2022):  
    print("Informes del Año", str(anio)) #Salida años desde el 2020 al 2022
```

Como pueden ver, en el bucle for a diferencia del while, la condición de parada es un **rango o una cantidad**.

Este se va a repetir hasta que esa cantidad se complete. En el caso de las listas, se ejecutará hasta pasar por **todos los elementos** de la misma (corta cuando la cantidad de elementos se acaba).

Es por ello que se dice que el bucle for se utiliza para “**contar**” elementos **finitos** (una cantidad determinada).