

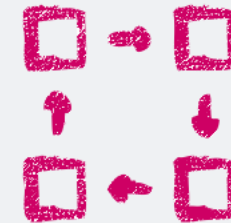
NIVEL 3

INSTRUCCIONES ITERATIVAS – WHILE



¿CUÁNDO USAMOS INSTRUCCIONES ITERATIVAS?

- ✓ Cuando necesitamos repetir varias veces un conjunto de instrucciones
- ✓ Puede ser un número fijo de veces o mientras se cumpla una condición



INSTRUCCIÓN WHILE

Mientras se cumple la **condición lógica** (booleana)

while *condición:*

acción

acción

• • •

acción

Se ejecutan estas **acciones**.
No hay límite en la cantidad de instrucciones que componen el cuerpo de un while.

EJEMPLO: WHILE

i empieza con el valor cero

MIENTRAS i sea menor que 3

```
EjemploContador.py ✕  
1 i = 0  
2 while i < 3:  
3     print(i)  
4     i += 1  
5 print("FIN")
```

Se imprime el valor de i
Se incrementa el valor de i

Resultado de la ejecución

```
Terminal 1/A ✕  
  
In [15]: runfile('C:  
Desktop/IP/N3-C1')  
0  
1  
2  
FIN
```

CUIDADO CON LOS CICLOS INFINITOS!!



Si dentro del ciclo, no se incrementa la variable `i`, la condición **NUNCA** será falsa y el ciclo se ejecuta infinitamente. Decimos que se «queda en un ciclo o loop infinito»

```
EjemploCicloInfinito.py ✕  
1 i = 0  
2 while i < 10:  
3     print(i)  
4 print("FIN")  
5
```

EJEMPLO: CÁLCULO DE UNA SUMATORIA

•

```

1 sumatoria = 0
2 i = 1
3 iteraciones = 0
4 while i <= 1000:
5     iteraciones+=1
6     sumatoria += i
7     i += 1
8 print("El resultado de la sumatoria es: ", sumatoria)
9 print("El número de iteraciones fue: ", iteraciones)
10 print("El valor de i es: ", i)

```

•

```

13 sumatoria = 0
14 i = 0
15 iteraciones = 0
16 while i < 1000:
17     iteraciones+=1
18     sumatoria += i
19     i += 1
20 print("El resultado de la sumatoria es: ", sumatoria)
21 print("El número de iteraciones fue: ", iteraciones)
22 print("El valor de i es: ", i)

```

$$\sum_{i=1}^n i$$

¿Cuál es el ciclo que implementa exactamente la sumatoria?

EJEMPLO: CÁLCULO DE UNA SUMATORIA

```
13 sumatoria = 0
14 i = 0
15 iteraciones = 0
16 while i < 1000:
17     iteraciones+=1
18     sumatoria += i
19     i += 1
20 print("El resultado de la sumatoria es: ", sumatoria)
21 print("El número de iteraciones fue: ", iteraciones)
22 print("El valor de i es: ", i)
```

Resultado de la ejecución

El resultado de la sumatoria es: 499500
El número de iteraciones fue: 1000
El valor de i es: 1000

Resultado de la ejecución

El resultado de la sumatoria es: 500500
El número de iteraciones fue: 1000
El valor de i es: 1001

```
1 sumatoria = 0
2 i = 1
3 iteraciones = 0
4 while i <= 1000:
5     iteraciones+=1
6     sumatoria += i
7     i += 1
8 print("El resultado de la sumatoria es: ", sumatoria)
9 print("El número de iteraciones fue: ", iteraciones)
10 print("El valor de i es: ", i)
```

Modifique el siguiente programa que calcula la raíz cuadrada de un número, para que obligue al usuario a teclear un número positivo. Es decir, que mientras el usuario digite números negativos, el programa le informe del error y le solicite un número diferente



```
EjemploRaiz.py ✕  
1 from math import sqrt  
2  
3 x = float(input("Digite un número positivo: "))  
4 print ("La raíz cuadrada de {0} es {1}".format(x, sqrt(x)))  
5
```