

Ejemplo de bucles simples con For

Ahora veremos como usar el segundo tipo de bucles en Python.

Los bucles For son muy parecidos a los bucles While, con la diferencia que la cantidad de elementos que se repetirán depende de una secuencia de elementos sobre la que la sentencia For realiza su repetición. Si esto no se entiende veamos la diferencia con un algoritmo simple, como crear una secuencia de numeros del 1 al 10. Veamos primero con un While.

In [1]:

```
cont=1
while cont <= 10:
    print(cont, end= ' ')
    cont = cont + 1
```

1 2 3 4 5 6 7 8 9 10

Com se explicó en el la pagina anterior. El While repite conforme a que se cumpla la condición. Mientras esta sea valida se repetirá el bucle tantas veces se mantenga la condición como verdadera (valor True). Por ello siempre es necesaria una sentencia que invalide la condición sino nuestro bucle será infinito. En el algoritmo anterior la ultima linea es la que llegado un momento tendrá un valor mayor a 10 y esa variable cont es la que invalidará la condición.

Veamos como funciona con For:

In [2]:

```
for cont in range(1, 11):
    print(cont, end= ' ')
```

1 2 3 4 5 6 7 8 9 10

Note que el bucle con for es más sencillo que el bucle While, no necesitamos una variable cont inicializada en un valor, ni tener que incrementar la variable para invalidar alguna condición. Simplemente definimos en los parametro de la función range(inicio,fin) los valores para que se genere la secuencia. Entonces aquí la función range genera la secuencia de elementos usando los siguientes parametros: inicio, fin, incremento. Esos tres valores permiten generar distintas formas de secuencias de numeros por lo que es una forma potente de realizar bucles.

Ahora veamos distintas formas de secuencias generadas con For.

In [3]:

```
#For de secuencia de pares
for i in range(2, 21, 2):
    print(i, end= ' ')
```

2 4 6 8 10 12 14 16 18 20

In [4]:

```
#secuencia invertida de impares desde 11 hasta 1.  
for i in range(11, 0, -2):  
    print(i, end= ' ')
```

11 9 7 5 3 1

In [5]:

```
#secuencia de numeros enteros desde 1 hasta N. Con N solicitado por e  
l usuario  
n = int(input())  
for i in range(1, n+1):  
    print(i, end= ' ')
```

22

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

In [3]:

```
#solución a) suma de pares de 2 hasta N. Si N es par hasta N sino has  
ta N-1.  
N = int(input())  
for x in range(2, N+1,2):  
    print(x, end= ' ')
```

20

2 4 6 8 10 12 14 16 18 20

In [5]:

```
#solución b) suma de pares de 2 hasta N. Si N es par hasta N sino has  
ta N-1.  
N = int(input())  
for x in range(2, N+1):  
    if x % 2 == 0:  
        print(x, end= ' ')
```

23

2 4 6 8 10 12 14 16 18 20 22

In []:

```
#Suma de numeros de 1 hasta N
```

In [6]:

```
suma = 0
N = int(input())
for i in range(1, N+1):
    suma = suma + i
print(suma)
```

10

55

In [7]:

```
# el mismo programa con pregunta y respuesta con texto.
suma = 0
N = int(input('ingrese el valor de N:'))
for i in range(1, N+1):
    suma = suma + i
print('la suma es:', suma)
```

ingrese el valor de N:11

la suma es: 66

In [8]:

```
#la suma de cuadrados de 1 hasta N.
n = int(input())
suma = 0
for i in range(1, n+1):
    suma = suma + i**2
print(suma)
```

10

385

In [9]:

```
# l asuma de cubos de 1 hasta N
n = int(input())
suma = 0
for i in range(1, n+1):
    suma = suma + i**3
print(suma)
```

10

3025

In [10]:

```
# 1 asuma de cubos de raices cuadradas
import math
n = int(input())
suma = 0
for i in range(1, n+1):
    suma = suma + math.sqrt(i)
print(suma)
```

10

22.4682781862041

In [11]:

```
# 1 asuma de cubos de raices cuadradas ahora con precisión a 3 decimales.
import math
n = int(input())
suma = 0
for i in range(1, n+1):
    suma = suma + math.sqrt(i)
print(round(suma,3))
```

10

22.468

Ahora trabajemos ejemplos usando bucles combinados o anidados.

vamos a trabajar una matriz de simbolos +

In [13]:

```
# imprimir una tabla de simbolos + de acuerdo a un valor N
N = int(input())
for i in range(1, N+1):
    for j in range(1, N+1):
        print('+ ', end='')
    print()
```

10

```
+++++++
+++++++
+++++++
+++++++
+++++++
+++++++
+++++++
+++++++
+++++++
+++++++
```

Observe que hace cada bucle. El bucle interno imprime filas de simbolos +, por el parametro end=" a la función print. Note como la función print() sin parametros imprime un salto de linea. La combinación de print(valor, end=") y print() permiten construir la tabla, el paramero end=" obliga print imprimir todo en la misma linea sin realizar saltos. print() no imprime nada pero si los saltos de linea.

Ahora en base a esa matriz comencemos a crear distintas formas.

In [15]:

```
#tringulo de simbolos +
N = int(input())
for i in range(1, N+1):
    for j in range(1, i+1):
        print('+', end='')
    print()
```

```
10
+
++
+++
++++
+++++
++++++
+++++++
+++++++
+++++++
+++++++
+++++++
```

In [17]:

```
#tringulo invertido simbolos +
N = int(input())
for i in range(1, N+1):
    for j in range(1, N-i+2):
        print('+', end='')
    print()
```

```
10
+++++++
+++++++
+++++++
+++++++
+++++++
+++++++
+++++
+++++
+++++
+++
++
+
```

In [1]:

```
#tringulo numeros secuenciales.  
N = int(input())  
cont = 1  
  
for i in range(1, N+1):  
    for j in range(1, i+1):  
        print(cont, end=' ')  
        cont = cont + 1  
    print()
```

```
10  
1  
2 3  
4 5 6  
7 8 9 10  
11 12 13 14 15  
16 17 18 19 20 21  
22 23 24 25 26 27 28  
29 30 31 32 33 34 35 36  
37 38 39 40 41 42 43 44 45  
46 47 48 49 50 51 52 53 54 55
```