

Bucles. While

Bucle while

Un bucle while permite repetir la ejecución de un grupo de instrucciones mientras se cumpla una condición (es decir, mientras la condición tenga el valor True) .

La sintaxis del bucle while es la siguiente:

```
while condicion:
    cuerpo del bucle
```

Cuando la secuencia del programa llega a un bucle while, Python **evalúa la condición** y, si es cierta, ejecuta el cuerpo del bucle. Una vez ejecutado el cuerpo del bucle, se repite el proceso (se evalúa de nuevo la condición

y, si es cierta, se ejecuta de nuevo el cuerpo del bucle) una y otra vez mientras la condición sea cierta. Únicamente cuando la condición sea **falsa**, el cuerpo del bucle no se ejecutará y continuará la ejecución del resto del programa.

La variable o las variables que aparezcan en la condición se suelen llamar **variables de control**. Las variables de control deben definirse (inicializarse) antes del bucle while y **modificarse** en el bucle while.

Por ejemplo, el siguiente programa escribe los números del 1 al 3:

```
i = 1
while i <= 3:
    print(i)
    i += 1
print("Programa terminado")
```

Puedes ejecutar paso a paso el programa en el [siguiente enlace](#)

El resultado sería:

```
1
2
3
Programa terminado
```

El ejemplo anterior se podría haber programado con un bucle **for** de la siguiente forma

```
for i in range (1, 4)
    print(i)
print("Programa terminado")
```

La ventaja de un bucle while es que la variable de control se puede **modificar con mayor flexibilidad**, como en el ejemplo siguiente:

```
i = 1
while i <= 50:
    print(i)
    i = 3*i + 1
print("Programa terminado")
```

La salida sería:

```
1
4
13
40
Programa terminado
```

Hacer el mismo programa utilizando **for** es posible, pero sería algo más complejo.

Otra ventaja del bucle while es que el número de iteraciones no está definida antes de empezar el bucle, por ejemplo porque los datos los proporciona el usuario.

Por ejemplo, el siguiente ejemplo pide un número positivo al usuario una y otra vez hasta que el usuario lo haga correctamente:

```
numero = int(input("Escriba un número positivo: "))
while numero < 0:
    print(";Ha escrito un número negativo! Inténtelo de nuevo")
    numero = int(input("Escriba un número positivo: "))
print("Gracias por su colaboración")
```

Ejemplo de ejecución:

```
Escriba un número positivo: -4
;Ha escrito un número negativo! Inténtelo de nuevo
Escriba un número positivo: -8
;Ha escrito un número negativo! Inténtelo de nuevo
Escriba un número positivo: 9
Gracias por su colaboración
```

for vs while

Ahora que conocemos los dos tipos de bucles es importante entender cuando debemos usar uno u otro. Aunque ambos son similares en el hecho de que repiten un bloque de código se usan en diferentes situaciones

Los bucles `for` se usan cuando se sabe exactamente cuantas veces se va a ejecutar el bucle. El número de veces se define, o bien dentro de la función `range()`, o bien dentro de algún **iterable** (los veremos más adelante). Por el contrario, los bucles `while` se ejecutan mientras se cumpla una condición y no siempre está claro cuantas veces se repetirá el bloque de código.

Bucles infinitos

Si la condición del bucle se cumple siempre, el bucle no terminará nunca de ejecutarse y tendremos lo que se denomina un **bucle infinito**.

Aunque a veces es necesario utilizar bucles infinitos en un programa, normalmente se deben a errores que se deben corregir.

Los bucles infinitos no intencionados deben evitarse pues significan perder el control del programa.

Para interrumpir un bucle infinito, hay que pulsar la combinación de teclas `Ctrl+C`. Al interrumpir un programa se mostrará un mensaje de error similar a éste:

```
Traceback (most recent call last):
  File "ejemplo.py", line 3, in <module>
    print(i)
KeyboardInterrupt
```

Por desgracia, es fácil programar involuntariamente un bucle infinito, por lo que es inevitable hacerlo de vez en cuando, sobre todo cuando se está aprendiendo a programar.

Estos algunos ejemplos de bucles infinitos:

El programador ha olvidado modificar la variable de control dentro del bucle

El siguiente programa hace que se impriman números 1 indefinidamente:

```
i = 1
while i <= 10:
    print(i, "", end="")
```

La salida sería:

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
```

El programador ha escrito una condición que se cumplirá siempre

Por ejemplo, en el siguiente programa se mostrarán números consecutivos indefinidamente:

```
i = 1
while i > 0:
    print(i, "", end="")
    i += 1
```

```
1 2 3 4 5 6 7 8 9 10 11 ...
```

Se aconseja expresar las condiciones como **desigualdades** en vez de comparar valores. En el ejemplo siguiente, el programador ha escrito una condición que se cumplirá siempre y el programa imprimirá números consecutivos indefinidamente:

```
i = 1
while i != 10:
    print(i, "",end="")
    i += 2
```

La salida sería:

```
1 3 5 7 9 11 13 15 ...
```

Referencias

- [CSP Python - The while loop](#)
- Apuntes generados a partir del curso [Introducción a la programación con Python](#) que se distribuye bajo una [Licencia Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional \(CC BY-SA 4.0\)](#).



tags: `pro` `utl` `python` `bucles` `for`