¿Qué es el bucle while en Python?

El bucle while en Python es una estructura de control. Las estructuras de control determinan qué ruta de código se sigue en el momento de la ejecución. Los bucles se utilizan generalmente para **ejecutar repetidamente un mismo bloque de código**. A continuación, te mostramos un resumen de las estructuras de control más importantes en Python:

| Estructura de control en Python | Explicación |
| --- | --- |
| Bifurcación if else | Ejecuta el bloque de código una vez si la condición es verdadera. |
| Bucle while de Python | Ejecuta el bloque de código repetidamente mientras la condición sea verdadera. |
| Bucle For de Python | Itera sobre los elementos de una colección, ejecutando un bloque de código para cada elemento. |

**Los programadores menos experimentados no suelen conocer los bucles** e intentan emular su funcionamiento. En lugar de definir una estructura de datos con varios elementos y procesarla con un, Los  bucle, definen una variable independiente para cada conjunto de datos.

En esta lección se tratan los bucles while.

## El 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

La ejecución de esta estructura de control while es la siguiente:

* Python evalúa la condición:
  + si el resultado es True se 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.
  + si el resultado es False, el cuerpo del bucle no se ejecuta y continúa 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 antes del bucle while y modificarse en el bucle while.

## Diagrama de flujo

El diagrama de flujo siguiente muestra la ejecución de un bucle while ...:

DIAGRAMA DE FLUJO: BUCLE WHILEHaga clic en los botones de avance y retroceso para ver la ejecución paso a paso El programa llega al bucle while.INICIOEvalúa la condición ...CONDICIÓNSi la condición es True, ejecuta el bloque ...TrueBLOQUEy vuelve a evaluar la condición.Si la condición es False,termina el bucle while ..FalseFIN

Si incluimos en este esquema la definición y modificación de las variables de control que intervienen en la condición, el diagrama de flujo sería el siguiente:

DIAGRAMA DE FLUJO: BUCLE WHILEEl programa llega al bucle while.INICIOEvalúa la condición ...CONDICIÓNSi la condición es True,ejecuta el bloque ...TrueBLOQUE... efectúa el cambio ...CAMBIOy vuelve a evaluar la condición.CONDICIÓNSi la condición es False,termina el bucle while..FalseFINEfectúa la asignación.ASIGNACIÓN

## Ejemplos de bucles while

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

**Ejemplo de bucle while 1**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

2

3

Programa terminado

Puede ver la ejecución paso a paso de este programa utilizando los iconos de avance y retroceso situados abajo a la derecha.

**Ejemplo de bucle while 1 - Paso 1**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

Se ejecuta la primera instrucción del programa.

En este caso, da el valor 1 a la variable i. La variable se utilizará como variable de control en el bucle while posterior.

**Ejemplo de bucle while 1 - Paso 2**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

A continuación se ejecuta el bucle. Un bucle está controlado por la condición. Si la condición se cumple, se ejecutan las instrucciones del cuerpo del bucle.

En este caso la condición es que la variable de control sea inferior a 3. Como i vale 1, la condición se cumple, así que se pasa a ejecutar las instrucciones del bucle, en este caso dos instrucciones.

**Ejemplo de bucle while 1 - Paso 3**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción imprime el valor de i, es decir, 1.

**Ejemplo de bucle while 1 - Paso 4**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

A continuación se añade 1 a la variable de control i.

En este caso, como la variable de control i valía 1, ahora toma el valor 2.

**Ejemplo de bucle while 1 - Paso 5**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 3. Como i vale 2, la condición se cumple, así que se ejecutan de nuevo las instrucciones del cuerpo del bucle.

**Ejemplo de bucle while 1 - Paso 6**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

2

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción imprime el valor de i, es decir, 2.

**Ejemplo de bucle while 1 - Paso 7**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

2

A continuación se añade 1 a la variable de control i.

En este caso, como la variable de control i valía 2, ahora toma el valor 3.

**Ejemplo de bucle while 1 - Paso 8**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

2

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 3. Como i vale 3, la condición se cumple, así que se ejecutan de nuevo las instrucciones del cuerpo del bucle.

**Ejemplo de bucle while 1 - Paso 9**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

2

3

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción imprime el valor de i, es decir, 3.

**Ejemplo de bucle while 1 - Paso 10**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

2

3

A continuación se añade 1 a la variable de control i.

En este caso, como la variable de control i valía 3, ahora toma el valor 4.

**Ejemplo de bucle while 1 - Paso 11**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

2

3

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 3. Como i vale 4, la condición ya no se cumple, por lo que el bucle se termina.

**Ejemplo de bucle while 1 - Paso 12**

i = 1

while i <= 3:

print(i)

i += 1

print("Programa terminado")

1

2

3

Programa terminado

La última instrucción del programa imprime el texto "Programa terminado".

El ejemplo anterior se podría haber programado con un bucle for. La ventaja de un bucle while es que la variable de control se puede modificar con mayor flexibilidad, como en el ejemplo siguiente:

**Ejemplo de bucle while 2**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

13

40

Programa terminado

Puede ver la ejecución paso a paso de este programa utilizando los iconos de avance y retroceso situados abajo a la derecha.

**Ejemplo de bucle while 2 - Paso 1**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

Se ejecuta la primera instrucción del programa.

En este caso, da el valor 1 a la variable i. La variable se utilizará como variable de control en el bucle while posterior.

**Ejemplo de bucle while 2 - Paso 2**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

A continuación se ejecuta el bucle. Un bucle está controlado por la condición. Si la condición se cumple, se ejecutan las instrucciones del cuerpo del bucle.

En este caso la condición es que la variable de control sea inferior a 50. Como i vale 1, la condición se cumple, así que se pasa a ejecutar las instrucciones del bucle, en este caso dos instrucciones.

**Ejemplo de bucle while 2 - Paso 3**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción imprime el valor de i, es decir, 1.

**Ejemplo de bucle while 2 - Paso 4**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

A continuación se modifica la variable de control i, multiplicando su valor por 3 y sumando 1.

En este caso, como la variable de control i valía 1, ahora toma el valor 4 (3\*1 + 1).

**Ejemplo de bucle while 2 - Paso 5**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 50. Como i vale 4, la condición se cumple, así que se ejecutan de nuevo las instrucciones del cuerpo del bucle.

**Ejemplo de bucle while 2 - Paso 6**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción imprime el valor de i, es decir, 4.

**Ejemplo de bucle while 2 - Paso 7**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

A continuación se modifica la variable de control i, multiplicando su valor por 3 y sumando 1.

En este caso, como la variable de control i valía 4, ahora toma el valor 13 (3\*4 + 1).

**Ejemplo de bucle while 2 - Paso 8**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 50. Como i vale 13, la condición se cumple, así que se ejecutan de nuevo las instrucciones del cuerpo del bucle.

**Ejemplo de bucle while 2 - Paso 9**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

13

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción imprime el valor de i, es decir, 13.

**Ejemplo de bucle while 2 - Paso 10**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

13

A continuación se modifica la variable de control i, multiplicando su valor por 3 y sumando 1.

En este caso, como la variable de control i valía 13, ahora toma el valor 40 (3\*13 + 1).

**Ejemplo de bucle while 2 - Paso 11**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

13

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 50. Como i vale 40, la condición se cumple, así que se ejecutan de nuevo las instrucciones del cuerpo del bucle.

**Ejemplo de bucle while 2 - Paso 12**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

13

40

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción imprime el valor de i, es decir, 40.

**Ejemplo de bucle while 2 - Paso 13**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

13

40

A continuación se modifica la variable de control i, multiplicando su valor por 3 y sumando 1.

En este caso, como la variable de control i valía 40, ahora toma el valor 121 (3\*40 + 1).

**Ejemplo de bucle while 2 - Paso 14**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

13

40

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 50. Como i vale 121, la condición ya no se cumple, por lo que el bucle se termina.

**Ejemplo de bucle while 2 - Paso 15**

i = 1

while i <= 50:

print(i)

i = 3 \* i + 1

print("Programa terminado")

1

4

13

40

Programa terminado

La última instrucción del programa imprime el texto "Programa terminado".

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:

**Ejemplo de bucle while 3**

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")

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

Puede ver la ejecución paso a paso de este programa utilizando los iconos de avance y retroceso situados abajo a la derecha.

**Ejemplo de bucle while 3 - Paso 1**

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")

Escriba un número positivo: -4

Se ejecuta la primera instrucción del programa.

En este caso, se pide al usuario que escriba un número positivo. En el ejemplo, el usuario escribe un número negativo (-4) y ese número se guarda en la variable de control "numero".

**Ejemplo de bucle while 3 - Paso 2**

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")

Escriba un número positivo: -4

A continuación se ejecuta el bucle. Un bucle está controlado por la condición. Si la condición se cumple, se ejecutan las instrucciones del cuerpo del bucle.

En este caso la condición es que la variable de control sea inferior a 0. Como "numero" vale -4, la condición se cumple, así que se pasa a ejecutar las instrucciones del bucle, en este caso dos instrucciones.

**Ejemplo de bucle while 3 - Paso 3**

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")

Escriba un número positivo: -4

¡Ha escrito un número negativo! Inténtelo de nuevo

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción avisa que se ha escrito un número negativo.

**Ejemplo de bucle while 3 - Paso 4**

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")

Escriba un número positivo: -4

¡Ha escrito un número negativo! Inténtelo de nuevo

Escriba un número positivo: -8

A continuación, se vuelve a pedir al usuario que escriba un número positivo. En el ejemplo, el usuario escribe un número negativo (-8) y ese número se guarda en la variable de control "numero".

En este caso, la variable de control "numero" toma el valor -8.

**Ejemplo de bucle while 3 - Paso 5**

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")

Escriba un número positivo: -4

¡Ha escrito un número negativo! Inténtelo de nuevo

Escriba un número positivo: -8

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 0. Como "numero" vale -8, la condición se cumple, así que se pasa a ejecutar las instrucciones del bucle, en este caso dos instrucciones.

**Ejemplo de bucle while 3 - Paso 6**

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")

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

A continuación se ejecutan las instrucciones del bloque.

La primera instrucción avisa que se ha escrito un número negativo.

**Ejemplo de bucle while 3 - Paso 7**

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")

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

A continuación, se vuelve a pedir al usuario que escriba un número positivo.

En el ejemplo, el usuario escribe por fin un número positivo (9) y ese número se guarda en la variable de control "numero". En este caso, la variable de control "numero" toma el valor 9.

**Ejemplo de bucle while 3 - Paso 8**

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")

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

A continuación se vuelve a evaluar la condición del bucle.

Se comprueba si la variable de control es inferior a 0. Como "numero" vale 9, la condición no se cumple, así que se pasa a ejecutar las instrucciones del bucle, por lo que el bucle se termina.

**Ejemplo de bucle while 3 - Paso 9**

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")

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

La última instrucción del programa agradece al usuario su colaboración.

## 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 y el programa imprimirá números 1 indefinidamente:
* i = 1
* while i <= 10:
* print(i, end=" ")

1 1 1 1 1 1 1 1 ...

* El programador ha escrito una condición que se cumplirá siempre y el programa imprimirá 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 != 100:
* print(i, end=" ")
* i += 2

1 3 5 7 9 11 ... 97 99 101 ...