

Bucle **for** (2) - Bucles anidados



Bucles anidados

- Bucles anidados
- Variables independientes
- Variables dependientes

En esta lección se tratan los bucles **for** anidados:

Bucles anidados

Se habla de bucles anidados cuando un bucle se encuentra en el bloque de instrucciones de otro bloque.

Al bucle que se encuentra dentro del otro se le puede denominar bucle interior o bucle interno. El otro bucle sería el bucle exterior o bucle externo.

Los bucles pueden tener cualquier nivel de anidamiento (un bucle dentro de otro bucle dentro de un tercero, etc.).

Aunque en Python no es necesario, se recomienda que los nombres de las variables de control de los bucles anidados no coincidan, para evitar ambigüedades.

Bucles anidados (variables independientes)

Se dice que las variables de los bucles son **independientes** cuando los valores que toma la variable de control del bucle interno **no** dependen del valor de la variable de control del bucle externo. Por ejemplo:

Ejemplo de bucle anidado (variables independientes)

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 0 y j vale 0
```

```
i vale 0 y j vale 1
```

```
i vale 1 y j vale 0
```

```
i vale 1 y j vale 1
```

```
i vale 2 y j vale 0
```

i vale 2 y j vale 1

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 anidado (variables independientes) - Paso 1

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

Se ejecuta el bucle externo.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es i y toma el valor 0.



Ejemplo de bucle anidado (variables independientes) - Paso 2

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

A continuación se ejecutan las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 0.



Ejemplo de bucle anidado (variables independientes) - Paso 3

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")  
i vale 0 y j vale 0
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (0 y 0).



Ejemplo de bucle anidado (variables independientes) - Paso 4

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")  
i vale 0 y j vale 0
```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 1.



Ejemplo de bucle anidado (variables independientes) - Paso 5

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")  
i vale 0 y j vale 0  
i vale 0 y j vale 1
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (0 y 1).



Ejemplo de bucle anidado (variables independientes) - Paso 6

```
for i in [0, 1, 2]:  
    for j in [0, 1]:
```

```
print(f"i vale {i} y j vale {j}")
```

```
i vale 0 y j vale 0
```

```
i vale 0 y j vale 1
```

Una vez terminado el bucle interno, se retoma el bucle externo.

La variable de control toma el segundo valor de la lista. En este caso, la variable de control es i y toma el valor 1.



Ejemplo de bucle anidado (variables independientes) - Paso 7

```
for i in [0, 1, 2]:
```

```
    for j in [0, 1]:
```

```
        print(f"i vale {i} y j vale {j}")
```

```
i vale 0 y j vale 0
```

```
i vale 0 y j vale 1
```

A continuación se ejecutan por segunda vez las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 0.



Ejemplo de bucle anidado (variables independientes) - Paso 8

```
for i in [0, 1, 2]:
```

```
    for j in [0, 1]:
```

```
        print(f"i vale {i} y j vale {j}")
```

```
i vale 0 y j vale 0
```

```
i vale 0 y j vale 1
```

```
i vale 1 y j vale 0
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (1 y 0).



Ejemplo de bucle anidado (variables independientes) - Paso 9

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

i vale 0 y j vale 0

i vale 0 y j vale 1

i vale 1 y j vale 0

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 1.



Ejemplo de bucle anidado (variables independientes) - Paso 10

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

i vale 0 y j vale 0

i vale 0 y j vale 1

i vale 1 y j vale 0

i vale 1 y j vale 1

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (1 y 1).



Ejemplo de bucle anidado (variables independientes) - Paso 11

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

i vale 0 y j vale 0

i vale 0 y j vale 1

i vale 1 y j vale 0

i vale 1 y j vale 1

Una vez terminado el bucle, se retoma el bucle externo.

La variable de control toma el tercer y último valor de la lista. En este caso, la variable de control es i y toma el valor 2.



Ejemplo de bucle anidado (variables independientes) - Paso 12

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

i vale 0 y j vale 0

i vale 0 y j vale 1

i vale 1 y j vale 0

i vale 1 y j vale 1

A continuación se ejecutan las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 0.



Ejemplo de bucle anidado (variables independientes) - Paso 13

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

i vale 0 y j vale 0

i vale 0 y j vale 1

i vale 1 y j vale 0

i vale 1 y j vale 1

i vale 2 y j vale 0

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (2 y 0).



Ejemplo de bucle anidado (variables independientes) - Paso 14

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")
```

i vale 0 y j vale 0

i vale 0 y j vale 1

i vale 1 y j vale 0

i vale 1 y j vale 1

i vale 2 y j vale 0

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 1.



Ejemplo de bucle anidado (variables independientes) - Paso 15

```
for i in [0, 1, 2]:
```

```
for j in [0, 1]:  
    print(f"i vale {i} y j vale {j}")  
i vale 0 y j vale 0  
i vale 0 y j vale 1  
i vale 1 y j vale 0  
i vale 1 y j vale 1  
i vale 2 y j vale 0  
i vale 2 y j vale 1
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (2 y 1).



Ejemplo de bucle anidado (variables independientes) - Paso 16

```
for i in [0, 1, 2]:  
    for j in [0, 1]:  
        print(f"i vale {i} y j vale {j}")  
i vale 0 y j vale 0  
i vale 0 y j vale 1  
i vale 1 y j vale 0  
i vale 1 y j vale 1  
i vale 2 y j vale 0  
i vale 2 y j vale 1
```

Una vez completado el bucle externo, el programa termina.



En el ejemplo anterior, el bucle externo (el controlado por i) se ejecuta 3 veces y el bucle interno (el controlado por j) se ejecuta dos veces por cada valor de i. Por ello la instrucción `print()` se ejecuta en total 6 veces (3 veces que se ejecuta el bucle externo x 2 veces que se ejecuta cada vez el bucle interno = 6 veces).

En general, el número de veces que se ejecuta el bloque de instrucciones del bucle interno es el producto de las veces que se ejecuta cada bucle.

En el ejemplo anterior se han utilizado listas para facilitar la comprensión del funcionamiento del bucle pero, si es posible hacerlo, se recomienda utilizar tipos `range()`, entre otros motivos porque durante la ejecución del programa ocupan menos memoria en el ordenador y se pueden hacer depender del desarrollo del programa.

El siguiente programa es equivalente al ejemplo anterior:

```
for i in range(3):  
    for j in range(2):  
        print(f"i vale {i} y j vale {j}")  
  
i vale 0 y j vale 0  
i vale 0 y j vale 1  
i vale 1 y j vale 0  
i vale 1 y j vale 1  
i vale 2 y j vale 0  
i vale 2 y j vale 1
```

Al escribir bucles anidados, hay que prestar atención al sangrado de las instrucciones, ya que ese sangrado indica a Python si una instrucción forma parte de un bloque u otro. En los tres siguientes programas la única diferencia es el sangrado de la última instrucción:

Ejemplo de sangrado en bucle anidado (1)

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")  
  
11 1 12 1 11 2 12 2 11 3 12 3
```

En este caso, la última instrucción forma parte del cuerpo del bucle interno. Por tanto el valor de `i` se escribe cada vez que se ejecuta el bucle interno.

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



Ejemplo de sangrado en bucle anidado (1) - Paso 1

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")
```

Se ejecuta el bucle externo.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es i y toma el valor 1.



Ejemplo de sangrado en bucle anidado (1) - Paso 2

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")
```

A continuación se ejecutan las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 11.



Ejemplo de sangrado en bucle anidado (1) - Paso 3

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")
```

11

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de dos instrucciones. La primera imprime el valor de j (11).



Ejemplo de sangrado en bucle anidado (1) - Paso 4

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 1
```

La segunda instrucción del bloque imprime el valor de i (1).



Ejemplo de sangrado en bucle anidado (1) - Paso 5

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")  
11 1
```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 12.



Ejemplo de sangrado en bucle anidado (1) - Paso 6

```
for i in [1, 2, 3]:  
    for j in [11, 12]:
```

```
print(j, end=" ")
print(i, end=" ")
11 1 12
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de dos instrucciones. La primera imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (1) - Paso 7

```
for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 1 12 1
```

La segunda instrucción del bloque imprime el valor de i (1).



Ejemplo de sangrado en bucle anidado (1) - Paso 8

```
for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 1 12 1
```

Una vez terminado el bucle interno, se retoma el bucle externo.

La variable de control toma el segundo valor de la lista. En este caso, la variable de control es i y toma el valor 2.



Ejemplo de sangrado en bucle anidado (1) - Paso 9

```
for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
        print(i, end=" ")

11 1 12 1
```

A continuación se ejecutan por segunda vez las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 11.



Ejemplo de sangrado en bucle anidado (1) - Paso 10

```
for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
        print(i, end=" ")

11 1 12 1 11
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de dos instrucciones. La primera imprime el valor de j (11).



Ejemplo de sangrado en bucle anidado (1) - Paso 11

```
for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
        print(i, end=" ")

11 1 12 1 11 2
```

La segunda instrucción del bloque imprime el valor de i (2).



Ejemplo de sangrado en bucle anidado (1) - Paso 12

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")  
  
11 1 12 1 11 2
```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 12.



Ejemplo de sangrado en bucle anidado (1) - Paso 13

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")  
  
11 1 12 1 11 2 12
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de dos instrucciones. La primera imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (1) - Paso 14

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")
```

```
11 1 12 1 11 2 12 2
```

La segunda instrucción del bloque imprime el valor de i (2).



Ejemplo de sangrado en bucle anidado (1) - Paso 15

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")
```

```
11 1 12 1 11 2 12 2
```

Una vez terminado el bucle, se retoma el bucle externo.

La variable de control toma el tercer y último valor de la lista. En este caso, la variable de control es i y toma el valor 3.



Ejemplo de sangrado en bucle anidado (1) - Paso 16

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
        print(i, end=" ")
```

```
11 1 12 1 11 2 12 2
```

A continuación se ejecutan las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 11.



Ejemplo de sangrado en bucle anidado (1) - Paso 17

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 1 12 1 11 2 12 2 11

```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de dos instrucciones. La primera imprime el valor de j (11).



Ejemplo de sangrado en bucle anidado (1) - Paso 18

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 1 12 1 11 2 12 2 11 3

```

La segunda instrucción del bloque imprime el valor de i (3).



Ejemplo de sangrado en bucle anidado (1) - Paso 19

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 1 12 1 11 2 12 2 11 3

```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 12.



Ejemplo de sangrado en bucle anidado (1) - Paso 20

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 1 12 1 11 2 12 2 11 3 12
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de dos instrucciones. La primera imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (1) - Paso 21

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 1 12 1 11 2 12 2 11 3 12 3
```

La segunda instrucción del bloque imprime el valor de i (3).



Ejemplo de sangrado en bucle anidado (1) - Paso 22

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 1 12 1 11 2 12 2 11 3 12 3
```

Una vez completado el bucle externo, el programa termina.



Ejemplo de sangrado en bucle anidado (2)

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1 11 12 2 11 12 3
```

En este caso, la última instrucción forma parte del cuerpo del bucle externo, pero no del interno. Por tanto el valor de *i* se escribe cada vez que se ha terminado de ejecutar el bucle interno.

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



Ejemplo de sangrado en bucle anidado (2) - Paso 1

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")
```

Se ejecuta el bucle externo.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es *i* y toma el valor 1.



Ejemplo de sangrado en bucle anidado (2) - Paso 2

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")
```

```
print(i, end=" ")
```

A continuación se ejecuta el bloque de instrucciones del bucle externo, formado en este caso por el bucle interno y una instrucción `print`.

Se ejecuta primero el bucle interno. La variable de control toma el primer valor de la lista. En este caso, la variable de control es `j` y toma el valor 11.



Ejemplo de sangrado en bucle anidado (2) - Paso 3

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")
```

11

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de `j` (11).



Ejemplo de sangrado en bucle anidado (2) - Paso 4

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")
```

11

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es `j` y toma el valor 12.



Ejemplo de sangrado en bucle anidado (2) - Paso 5

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (2) - Paso 6

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1
```

Una vez completado el bucle interno, se ejecuta la siguiente instrucción, que imprime el valor de i (1).



Ejemplo de sangrado en bucle anidado (2) - Paso 7

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1
```

Una vez terminado el bloque de instrucciones que forman el bucle externo, se retoma el bucle externo.

La variable de control toma el segundo valor de la lista. En este caso, la variable de control es i y toma el valor 2.



Ejemplo de sangrado en bucle anidado (2) - Paso 8

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1
```

A continuación se ejecuta por segunda vez el bloque de instrucciones del bucle externo, formado en este caso por el bucle interno y una instrucción `print`.

Se ejecuta primero el bucle interno. La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 11.



Ejemplo de sangrado en bucle anidado (2) - Paso 9

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1 11
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (11).



Ejemplo de sangrado en bucle anidado (2) - Paso 10

```
for i in [1, 2, 3]:
```

```

for j in [11, 12]:
    print(j, end=" ")
    print(i, end=" ")

```

11 12 1 11

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 12.



Ejemplo de sangrado en bucle anidado (2) - Paso 11

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")

```

11 12 1 11 12

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (2) - Paso 12

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")

```

11 12 1 11 12 2

Una vez completado el bucle interno, se ejecuta la siguiente instrucción, que imprime el valor de i (2).



Ejemplo de sangrado en bucle anidado (2) - Paso 13

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1 11 12 2
```

Una vez terminado el bloque de instrucciones que forman el bucle externo, se retoma el bucle externo.

La variable de control toma el tercer y último valor de la lista. En este caso, la variable de control es i y toma el valor 3.



Ejemplo de sangrado en bucle anidado (2) - Paso 14

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1 11 12 2
```

A continuación se ejecuta por tercera vez el bloque de instrucciones del bucle externo, formado en este caso por el bucle interno y una instrucción **print**.

Se ejecuta primero el bucle interno. La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 11.



Ejemplo de sangrado en bucle anidado (2) - Paso 15

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")
```

```
11 12 1 11 12 2 11
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (11).



Ejemplo de sangrado en bucle anidado (2) - Paso 16

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1 11 12 2 11
```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 12.



Ejemplo de sangrado en bucle anidado (2) - Paso 17

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 1 11 12 2 11 12
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (2) - Paso 18


```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 12 1 11 12 2 11 12 3

```

Una vez completado el bucle interno, se ejecuta la siguiente instrucción, que imprime el valor de i (3).



Ejemplo de sangrado en bucle anidado (2) - Paso 19

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 12 1 11 12 2 11 12 3

```

Una vez completado el bucle externo, el programa termina.



Ejemplo de sangrado en bucle anidado (3)

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 12 11 12 11 12 3

```

En este caso, la última instrucción no forma parte de ningún bucle. Por tanto el valor de i se escribe una sola vez, al terminarse de ejecutar el bucle externo.

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



Ejemplo de sangrado en bucle anidado (3) - Paso 1

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")
```

Se ejecuta el bucle externo.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es i y toma el valor 1.



Ejemplo de sangrado en bucle anidado (3) - Paso 2

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")
```

A continuación se ejecuta el bloque de instrucciones del bucle externo, formado en este caso por el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 11.



Ejemplo de sangrado en bucle anidado (3) - Paso 3

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (11).



Ejemplo de sangrado en bucle anidado (3) - Paso 4

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
print(i, end=" ")  
11
```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 12.



Ejemplo de sangrado en bucle anidado (3) - Paso 5

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
print(i, end=" ")  
11 12
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (3) - Paso 6

```
for i in [1, 2, 3]:
```

```

for j in [11, 12]:
    print(j, end=" ")
print(i, end=" ")

```

11 12

Una vez terminado el bloque de instrucciones que forman el bucle externo, se retoma el bucle externo.

La variable de control toma el segundo valor de la lista. En este caso, la variable de control es i y toma el valor 2.



Ejemplo de sangrado en bucle anidado (3) - Paso 7

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")

```

11 12

A continuación se ejecuta por segunda vez el bloque de instrucciones del bucle externo, formado en este caso por el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 11.



Ejemplo de sangrado en bucle anidado (3) - Paso 8

```

for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")

```

11 12 11

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (11).



Ejemplo de sangrado en bucle anidado (3) - Paso 9

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 11
```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 12.



Ejemplo de sangrado en bucle anidado (3) - Paso 10

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 11 12
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (3) - Paso 11

```
for i in [1, 2, 3]:  
    for j in [11, 12]:
```

```
print(j, end=" ")
print(i, end=" ")
11 12 11 12
```

Una vez terminado el bloque de instrucciones que forman el bucle externo, se retoma el bucle externo.

La variable de control toma el tercer y último valor de la lista. En este caso, la variable de control es i y toma el valor 3.



Ejemplo de sangrado en bucle anidado (3) - Paso 12

```
for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 12 11 12
```

A continuación se ejecuta por tercera vez el bloque de instrucciones del bucle externo, formado en este caso por el bucle interno.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 11.



Ejemplo de sangrado en bucle anidado (3) - Paso 13

```
for i in [1, 2, 3]:
    for j in [11, 12]:
        print(j, end=" ")
    print(i, end=" ")
11 12 11 12 11
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (11).



Ejemplo de sangrado en bucle anidado (3) - Paso 14

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 11 12 11
```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 12.



Ejemplo de sangrado en bucle anidado (3) - Paso 15

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")  
11 12 11 12 11 12
```

A continuación se ejecutan por segunda vez las instrucciones del bloque.

En este caso el bloque consta de una instrucción, que imprime el valor de j (12).



Ejemplo de sangrado en bucle anidado (3) - Paso 16

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
    print(i, end=" ")
```

```
11 12 11 12 11 12 3
```

Una vez completado el bucle interno, se ejecuta la siguiente instrucción, que imprime el valor de `i` (3).



Ejemplo de sangrado en bucle anidado (3) - Paso 17

```
for i in [1, 2, 3]:  
    for j in [11, 12]:  
        print(j, end=" ")  
print(i, end=" ")
```

```
11 12 11 12 11 12 3
```

Una vez ejecutada la última instrucción, el programa termina.



La costumbre más extendida es utilizar la letra "i" como nombre de la variable de control del bucle externo y la letra "j" como nombre de la variable de control del bucle interno (o "k" si hay un tercer nivel de anidamiento), pero se puede utilizar cualquier otro nombre válido.

En Python se puede incluso utilizar la misma variable en los dos bucles anidados porque Python las trata como si fueran dos variables distintas. Por ejemplo:

```
for i in range(3):  
    print(f"i (externa) vale {i}")  
    for i in range(2):  
        print(f"i (interna) vale {i}")
```

```
i (externa) vale 0
```

```
i (interna) vale 0
```

```
i (interna) vale 1
```

```
i (externa) vale 1
```

```
i (interna) vale 0
```

```
i (interna) vale 1
```



```
i (externa) vale 2
```

```
i (interna) vale 0
```

```
i (interna) vale 1
```

De todas formas, este uso no se recomienda porque da lugar a programas difíciles de interpretar y además no es habitual en otros lenguajes de programación. Se aconseja utilizar siempre nombres de variables distintos.

Bucles anidados (variables dependientes)

Se dice que las variables de los bucles son **dependientes** cuando los valores que toma la variable de control del bucle interno dependen del valor de la variable de control del bucle externo. Por ejemplo:

Ejemplo de bucle anidado (variables dependientes) 1

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

```
i vale 2 y j vale 0
```

```
i vale 2 y j vale 1
```

```
i vale 3 y j vale 0
```

```
i vale 3 y j vale 1
```

```
i vale 3 y j vale 2
```

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 anidado (variables dependientes) 1 - Paso 1

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

Se ejecuta el bucle externo.

La variable de control toma el primer valor de la lista. En este caso, la variable de control es i y toma el valor 1.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 2

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

A continuación se ejecutan las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma los valores de la lista `range(i)`. Como i tiene el valor 1, la lista `range(i)` es `range(1)`, es decir [0]

La variable de control toma el primer y único valor de la lista. En este caso, la variable de control es j y toma el valor 0.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 3

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")  
i vale 1 y j vale 0
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (1 y 0).



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 4

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

Una vez terminado el bucle interno, se retoma el bucle externo.

La variable de control toma el segundo valor de la lista. En este caso, la variable de control es i y toma el valor 2.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 5

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

A continuación se ejecutan las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma los valores de la lista `range(i)`. Como i tiene el valor 2, la lista `range(i)` es `range(2)`, es decir [0, 1]

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 0.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 6

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

```
i vale 2 y j vale 0
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (2 y 0).



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 7

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

```
i vale 2 y j vale 0
```

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo y último valor de la lista. En este caso, la variable de control es j y toma el valor 1.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 8

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

```
i vale 2 y j vale 0
```

```
i vale 2 y j vale 1
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (2 y 1).



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 9

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

```
i vale 2 y j vale 0
```

```
i vale 2 y j vale 1
```

Una vez terminado el bucle interno, se retoma el bucle externo.

La variable de control toma el tercer y último valor de la lista. En este caso, la variable de control es i y toma el valor 3.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 10

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

```
i vale 2 y j vale 0
```

```
i vale 2 y j vale 1
```

A continuación se ejecutan las instrucciones del bloque, en este caso, el bucle interno.

La variable de control toma los valores de la lista `range(i)`. Como i tiene el valor 3, la lista `range(i)` es `range(3)`, es decir [0, 1, 2]

La variable de control toma el primer valor de la lista. En este caso, la variable de control es j y toma el valor 0.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 11

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

```
i vale 2 y j vale 0
```

```
i vale 2 y j vale 2
```

```
i vale 3 y j vale 0
```

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (3 y 0).



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 12

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

i vale 1 y j vale 0

i vale 2 y j vale 0

i vale 2 y j vale 2

i vale 3 y j vale 0

A continuación continúa la ejecución del bucle interno.

La variable de control toma el segundo valor de la lista. En este caso, la variable de control es j y toma el valor 1.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 13

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

i vale 1 y j vale 0

i vale 2 y j vale 0

i vale 2 y j vale 2

i vale 3 y j vale 0

i vale 3 y j vale 1

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (3 y 1).



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 14

```
for i in [1, 2, 3]:  
    for j in range(i) :  
        print(f"i vale {i} y j vale {j}")
```

i vale 1 y j vale 0

i vale 2 y j vale 0

i vale 2 y j vale 2

i vale 3 y j vale 0

i vale 3 y j vale 1

A continuación continúa la ejecución del bucle interno.

La variable de control toma el tercer y último valor de la lista. En este caso, la variable de control es j y toma el valor 2.



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 15

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

i vale 1 y j vale 0

i vale 2 y j vale 0

i vale 2 y j vale 1

i vale 3 y j vale 0

i vale 3 y j vale 1

i vale 3 y j vale 2

A continuación se ejecutan las instrucciones del bloque.

En este caso el bloque consta de una sola instrucción que imprime los valores de i y j (3 y 2).



Ejemplo de bucle anidado (variables dependientes) 1 - Paso 16

```
for i in [1, 2, 3]:  
    for j in range(i):  
        print(f"i vale {i} y j vale {j}")
```

```
i vale 1 y j vale 0
```

```
i vale 2 y j vale 0
```

```
i vale 2 y j vale 1
```

```
i vale 3 y j vale 0
```

```
i vale 3 y j vale 1
```

```
i vale 3 y j vale 2
```

Una vez completado el bucle externo, el programa termina.



En el ejemplo anterior, el bucle externo (el controlado por *i*) se ejecuta 3 veces y el bucle interno (el controlado por *j*) se ejecuta 1, 2 y 3 veces. Por ello la instrucción `print()` se ejecuta en total 6 veces.

La variable *i* toma los valores de 1 a 3 y la variable *j* toma los valores de 0 a *i*, por lo que cada vez el bucle interno se ejecuta un número diferente de veces:

- Cuando *i* vale 1, `range(i)` devuelve la lista [0] y por tanto el bucle interno se ejecuta una sola vez y el programa escribe una sola línea en la que *i* vale 1 (y *j* vale 0).
 - Cuando *i* vale 2, `range(i)` devuelve la lista [0, 1] y por tanto el bucle interno se ejecuta dos veces y el programa escribe dos líneas en la que *i* vale 2 (y *j* vale 0 o 1 en cada una de ellas).
 - Cuando *i* vale 3, `range(i)` devuelve la lista [0, 1, 2] y por tanto el bucle interno se ejecuta tres veces y el programa escribe tres líneas en la que *i* vale 3 (y *j* vale 0, 1 o 2 en cada una de ellas).
-