

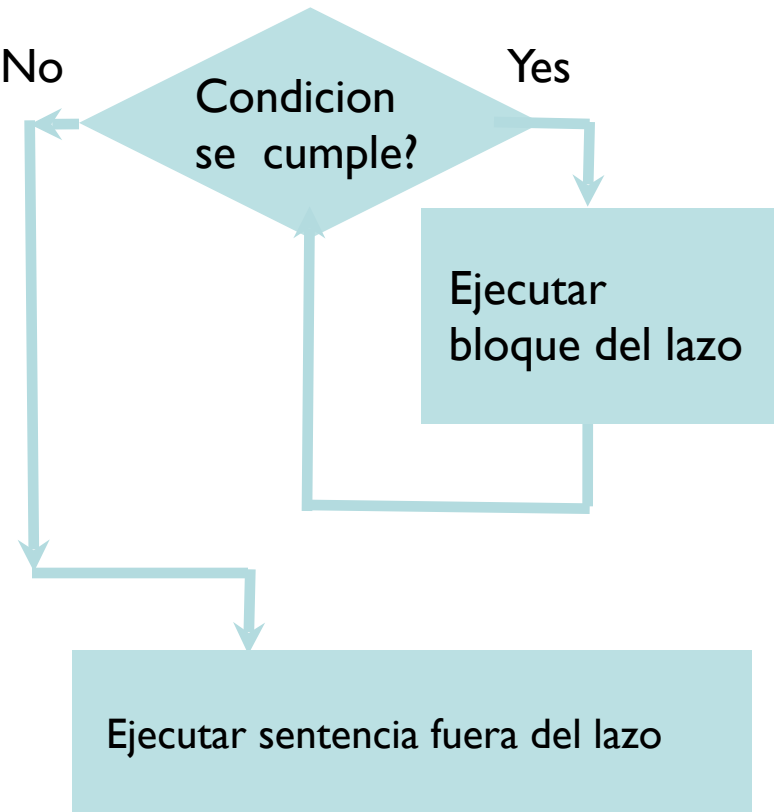
Fundamentos de programación

Estructuras de control Iterativas

Iteraciones: Lazos

- Iterar se refiere a la repetición de un conjunto de sentencias.
- Existen dos sentencias para implementar iteraciones: while y for.
- Sentencia while
 - Repite sentencias siempre y cuando una condición sea cierta
- Sentencia for
 - Repite instrucciones un número específico de veces

Lazo while



- El lazo se ejecuta MIENTRAS una condición se cumple
- En la condición se verifica el valor de una variable
 - Inicializar la variable fuera del lazo
 - Actualizar variable en el lazo (siguiente slide muestra un ejemplo)
- Si el valor de la variable verificada en la condición no se actualiza, se tendría un lazo infinito

Lazo while

count = 4

count > 0 ?

No

Yes

print count

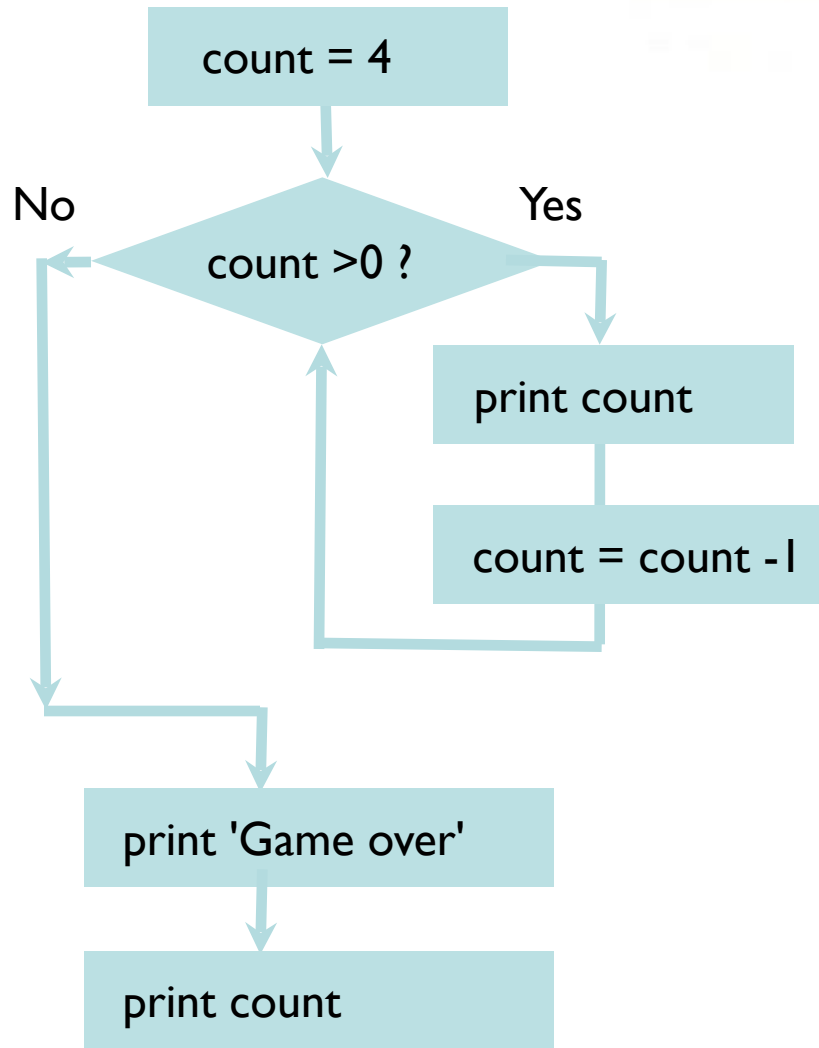
count = count - 1

print 'Game over'

print count

```
count = 4
while count > 0 :
    print count
    count = count - 1
print 'Game over'
print count
```

Lazo while



```
count = 4
while count > 0 :
    print count
    count = count - 1
print 'Game over'
print count
```

4

3

2

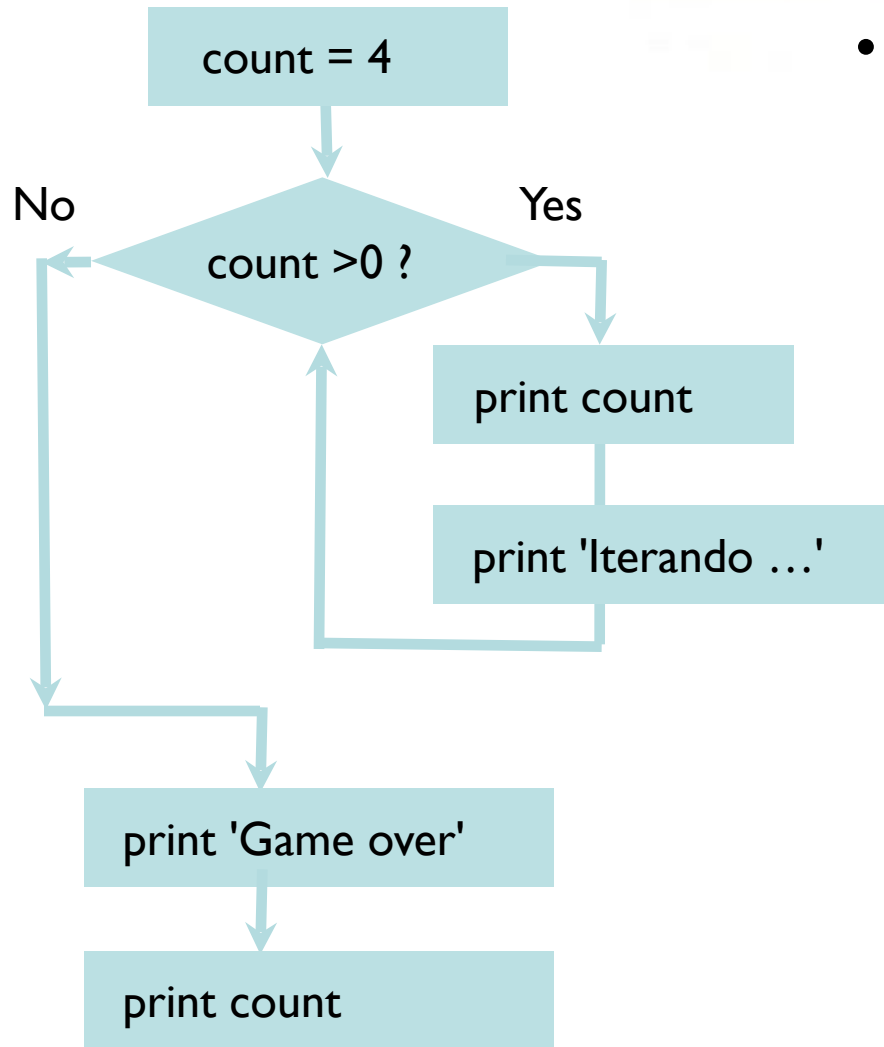
1

Game over

0

Lazo while

- Cuál es el problema con este lazo?



```
count = 4
while count > 0 :
    print count
    print 'Iterando'
print 'Game over'
print count
```

Lazo For

Fundamentos de Programación

- El lazo for en Python itera sobre una secuencia de elementos. Por ejemplo, una lista de números.
- Para cada elemento en la secuencia, se ejecutarán todas las sentencias dentro del bloque del lazo.

```
for variableName in groupOfValues:  
    sentencias
```

- Cuando se procesa el último elemento del grupo de valores, automáticamente, el lazo termina.

Lazo For

Fundamentos de Programación

- Una lista de números se puede definir usando:
`elementos = [1, 2, 3, 4, 5]`
- La sentencia `range(inicio, fin)` devuelve una lista de números entre el valor `inicio` y el valor `fin - 1`
 - `>> range(0, 4)`
 - `[0, 1, 2, 3]`

Lazo for

Fundamentos de Programación

```
for i in range(1, 4) :  
    print i  
print "Fin del programa"
```

Lazo for

Fundamentos de Programación

- Es posible producir secuencias de números no consecutivos:

```
print("Contando de dos en dos:")  
for i in range(0, 11, 2):  
    print i
```

Lazo for

- Es posible iterar en orden descendente:

```
print("Inicia el conteo regresivo: ")  
for i in range(5, 0, -1):  
    print i  
print "Fin"
```

Inicia el conteo regresivo:

5

4

3

2

1

Fin

Terminar un lazo: break

Fundamentos de Programación

```
x = 120
while (x > 1):
    x /= 2
    if (x % 2 == 1):
        break
print x
```

```
while (True):
    /* ... Obtener un valor del usuario ... */
    if (condition):
        break;
    /* ... Procesar el valor ... */
```

Omitir parte del lazo: continue

```
for letter in 'ESPOL':  
    if letter == 'O':  
        continue  
    print 'Letra actual :', letter  
  
var = 9  
while var > 0:  
    if var <= 5 and var >2:  
        var = var -1  
        continue  
    print 'Valor de la variable :', var  
    var = var -1  
  
print "Bye bye!"
```

Omitir parte del lazo: continue

```
for letter in 'ESPOL':
    if letter == 'O':
        continue
    print 'Letra actual :', letter

var = 9
while var > 0:
    if var > 2 and var <= 5:
        var = var - 1
        continue
    print 'Valor de la variable :', var
    var = var - 1

print "Bye bye!"
```

```
Letra actual : E
Letra actual : S
Letra actual : P
Letra actual : L
Valor de la variable : 9
Valor de la variable : 8
Valor de la variable : 7
Valor de la variable : 6
Valor de la variable : 2
Valor de la variable : 1
Bye bye!
```

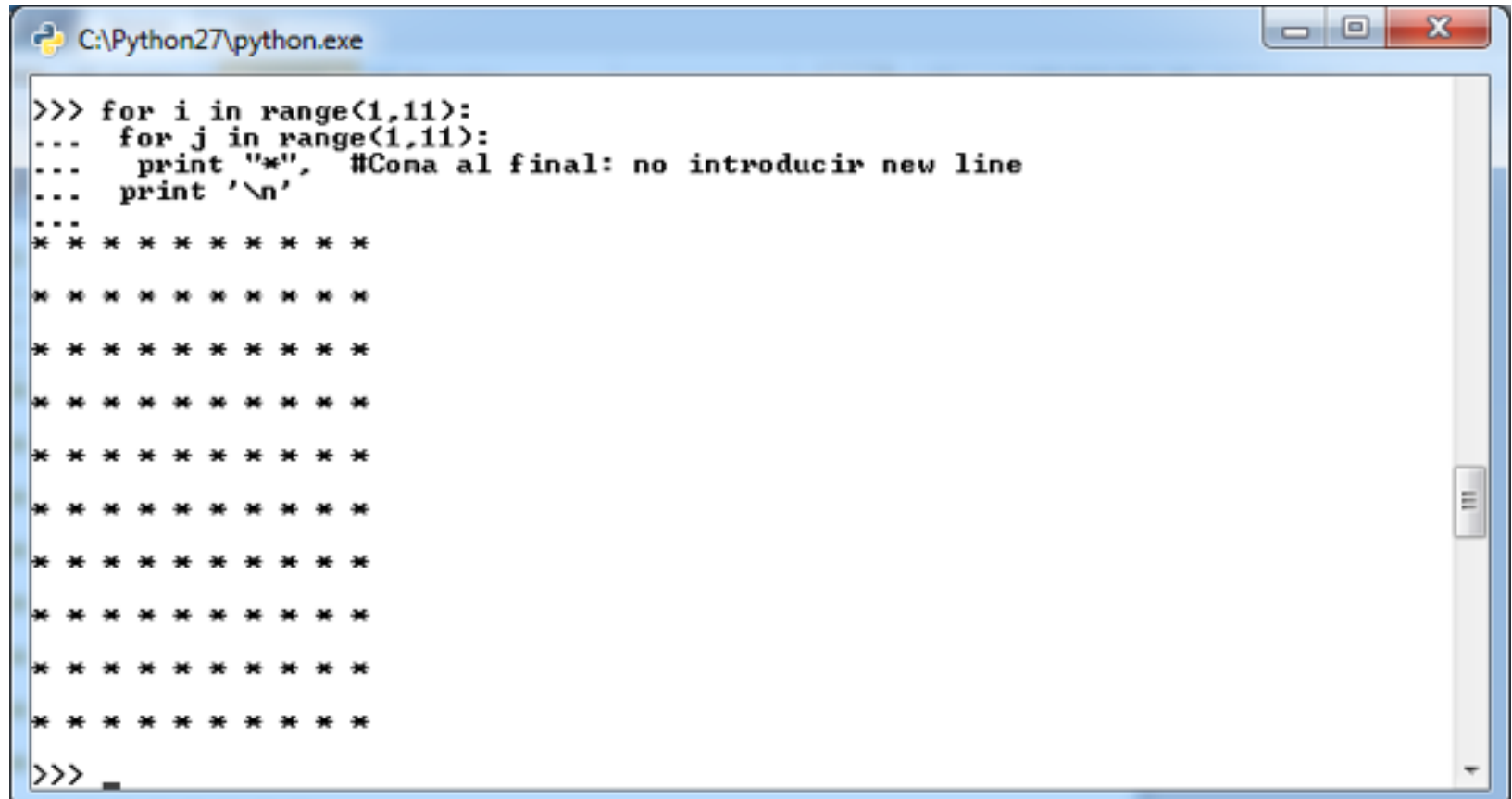
Lazos anidados

- Outer loop(n iteraciones): El lazo externo se ejecutará el número de veces definidas (n veces)
- Inner loop(m iteraciones): El lazo interno se ejecutará por completo cada vez que se ejecute el lazo externo ($n*m$)

```
for i in range(1,11):  
    for j in range(1,11):  
        print "*",      #Coma al final: no introducir new line  
    print '\n'
```

Lazos anidados

Fundamentos de Programación



A screenshot of a Python interpreter window titled "C:\Python27\python.exe". The window displays a nested loop structure in Python code that prints a 10x10 grid of asterisks. The code is as follows:

```
>>> for i in range(1,11):  
...   for j in range(1,11):  
...     print "*", #Coma al final: no introducir new line  
...     print '\n'  
...  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

The output shows a 10x10 grid of asterisks, with each row containing 10 asterisks. The prompt ">>>" is visible at the bottom left of the window.