

# Rangos y el Ciclo For

Lógica y Algoritmia

# Rangos

range([inicio], fin [,paso])

↑  
Valor opcional,  
es incluyente

↑  
Valor obligatorio,  
no incluido

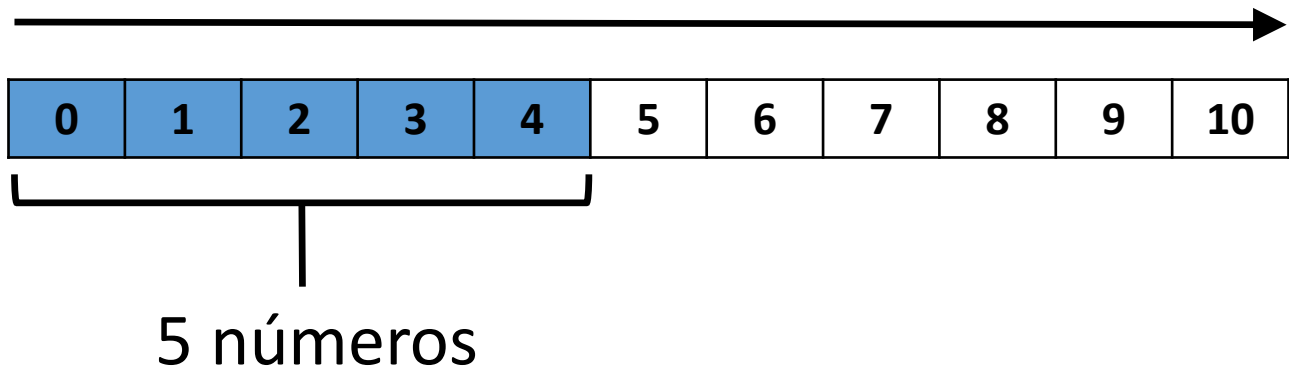
↑  
Valor opcional

Secuencia numérica delimitada por un valor de inicio hasta un valor final (que no es incluido), y cuyos elementos están **determinados** por el valor de paso.

# Caso 1: `range(fin)`

Se genera una secuencia de números que va desde 0 hasta alcanzar el valor final (sin incluirlo):

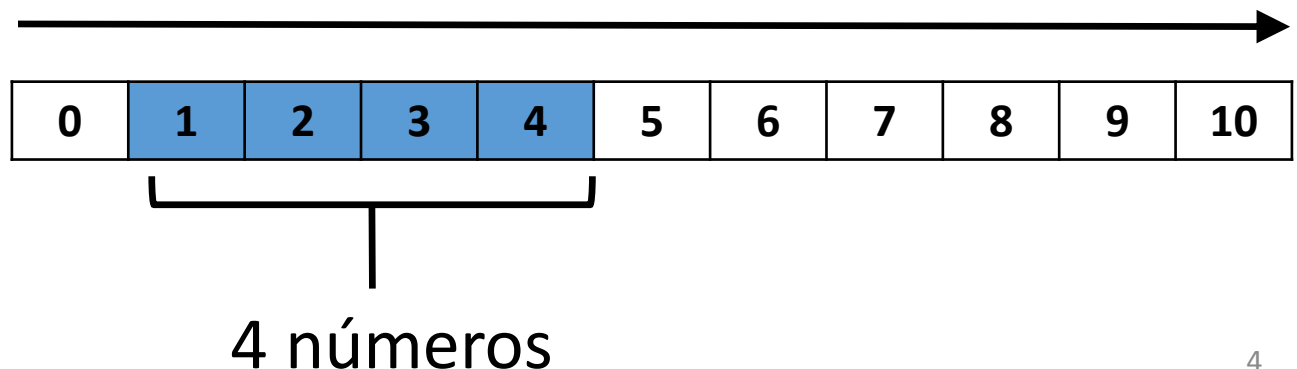
Ejemplo: ***range(5)***



## Caso 2: `range(inicio, fin)`

Se establece el primer número como inicial inclusivo y se incrementa la secuencia en una unidad hasta el número final sin incluirlo:

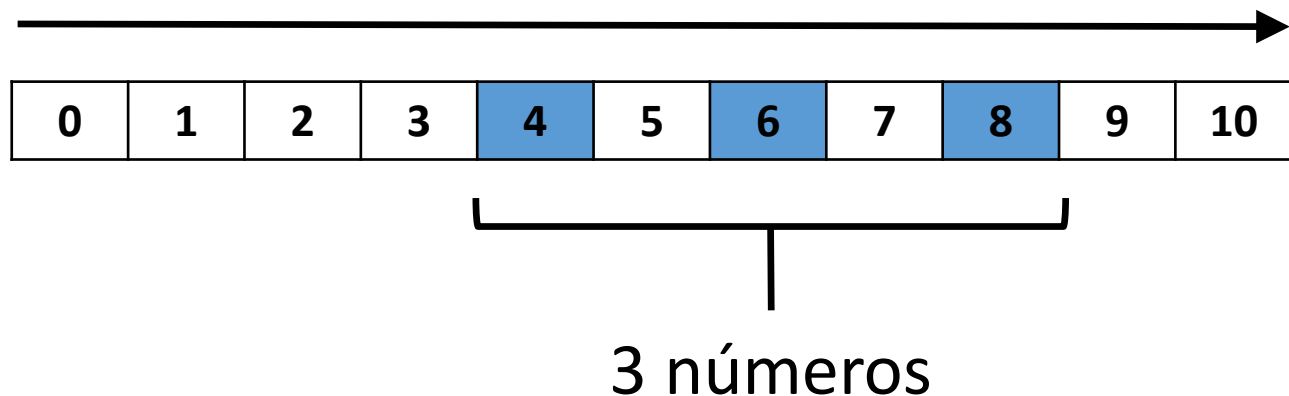
Ejemplo: ***`range(1,5)`***



## Caso 3: `range(inicio, fin, paso)`

Se establece el primer número como inicial inclusivo y se incrementa la secuencia en el número de pasos indicado, hasta llegar al valor final sin incluirlo:

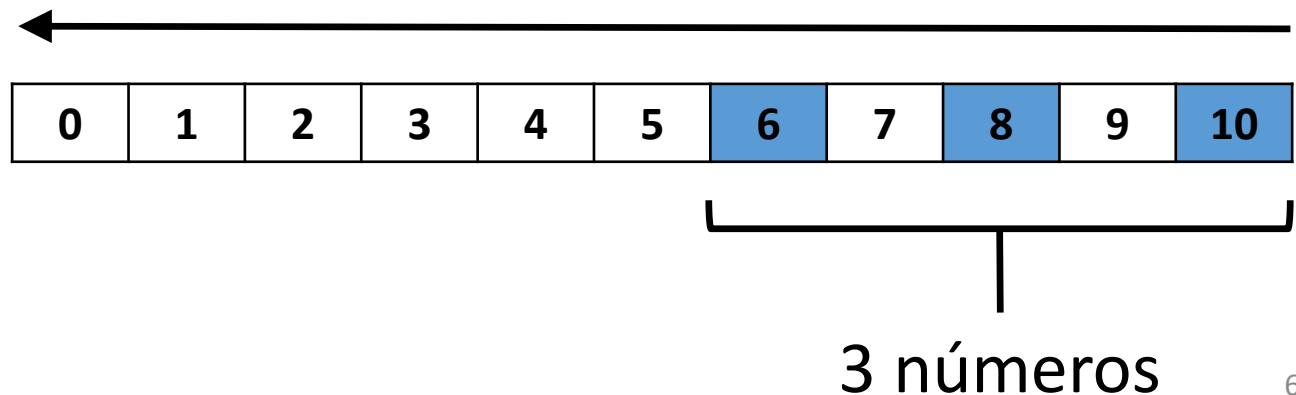
Ejemplo: **`range(4,10,2)`**



## Caso 3b: `range(inicio, fin, paso)`

La secuencia tiene un valor de inicio superior al de finalización, y el valor de paso es negativo; inicia con valor de 10 y decrementa en -2 para tomar el valor correspondiente y continúa así hasta encontrar el valor final sin incluirlo.

Ejemplo: ***range(10,4,-2)***



# Ciclo For



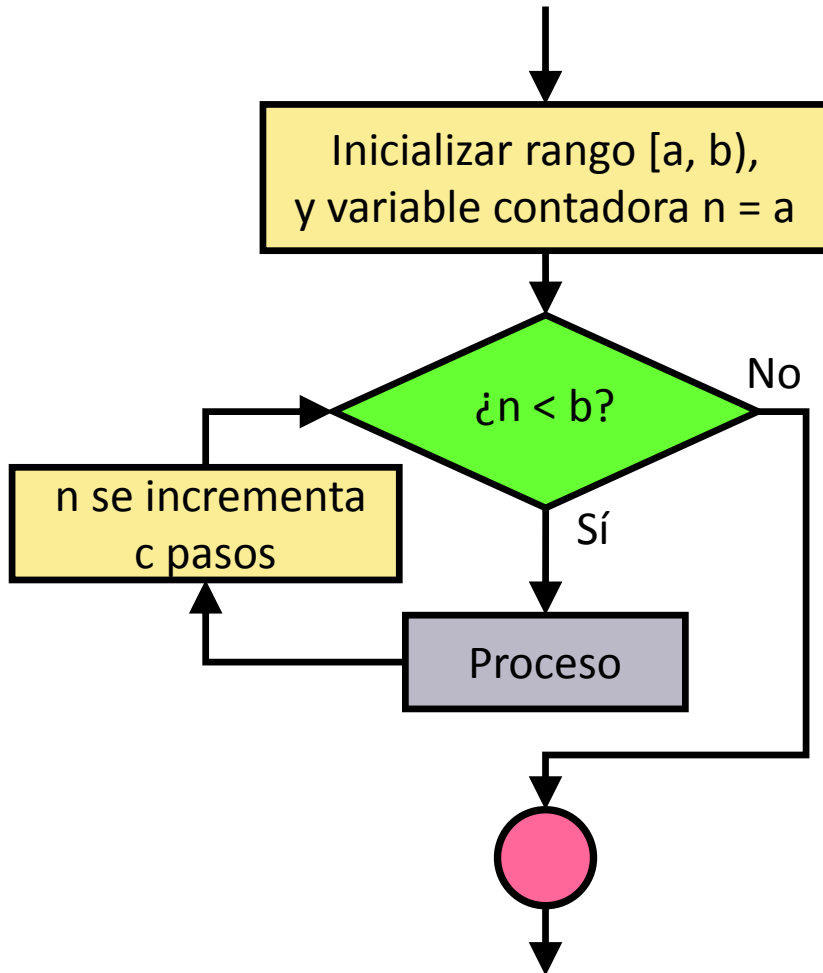
# Ciclo For

Estructura repetitiva que ejecuta un proceso una cantidad **determinada** de veces según el número de valores que haya dentro de un rango asociado.



# Ciclo For

Diagrama de Flujo:



Pseudocódigo:

Para n en el rango (a, b, c):  
Ejecute Proceso

# Ciclo For: Ejemplo 1 (1)

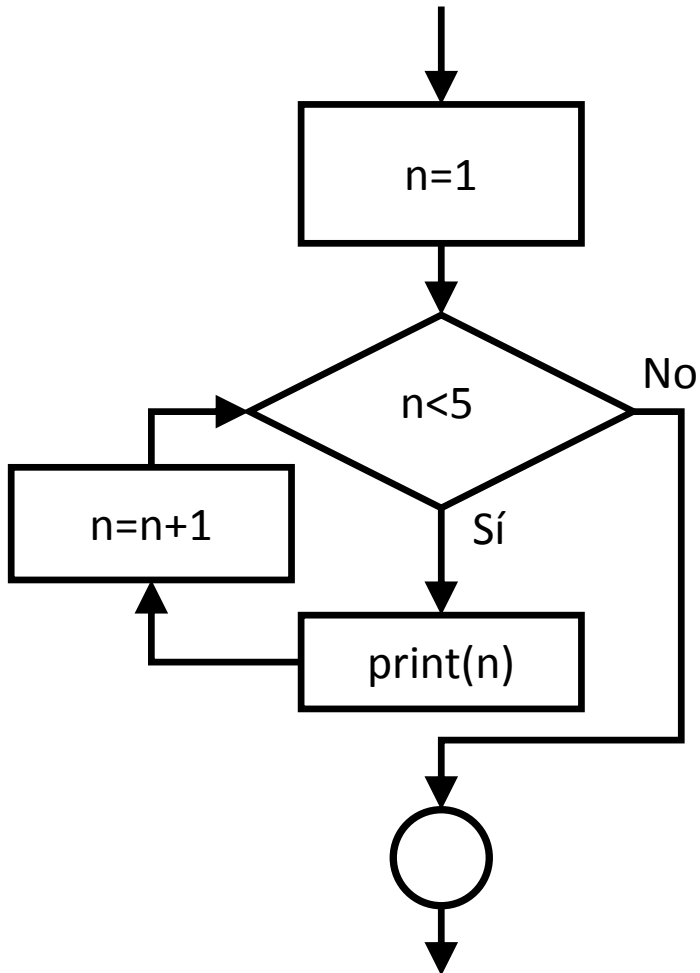
Código:

```
for n in range(1, 5, 1):  
    print(n)
```

Ejecución:

```
>>>  
1  
2  
3  
4  
>>> |
```

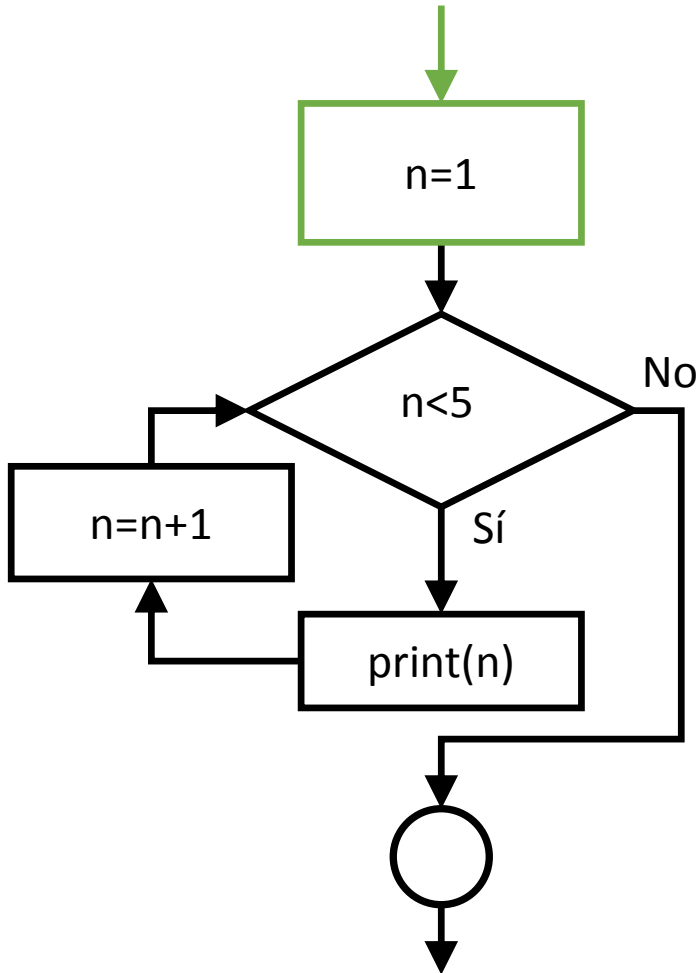
# Ciclo For: Ejemplo 1 (2)



Código:

```
for n in range(1, 5, 1):  
    print(n)
```

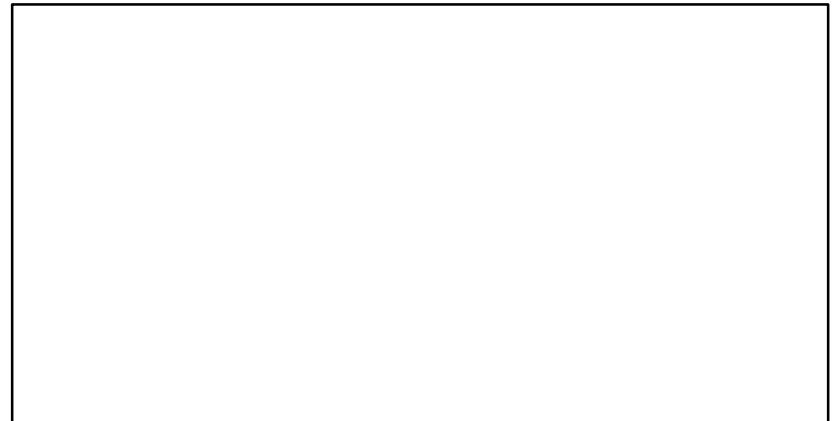
# Ciclo For: Ejemplo 1 (3)



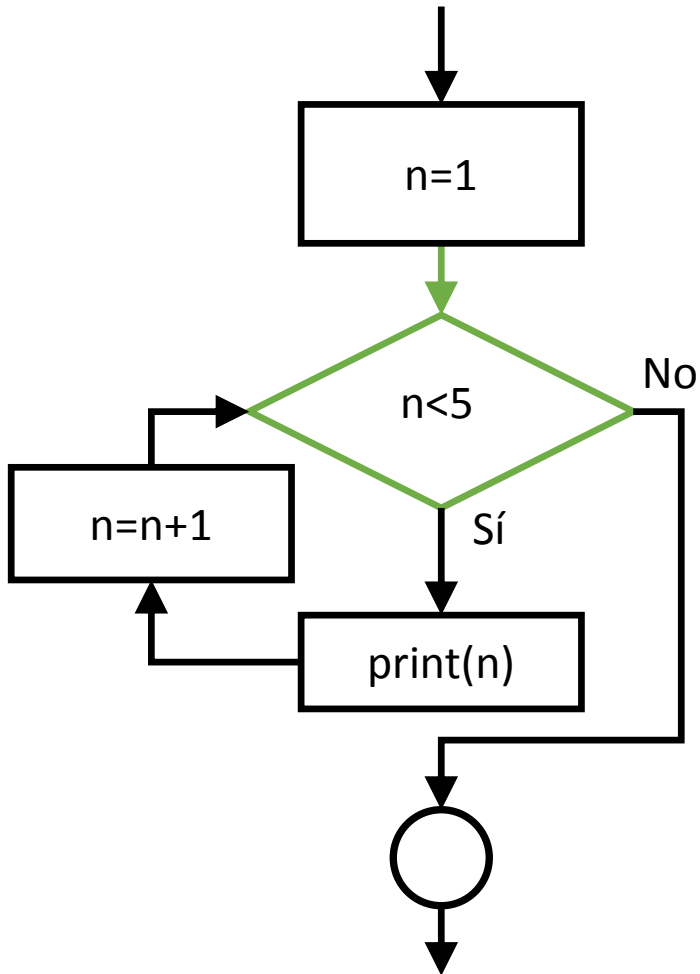
Variables:

Se crea la variable contadora “n” y se le asigna el valor inicial del rango, es decir 1

Ejecución:



# Ciclo For: Ejemplo 1 (4)



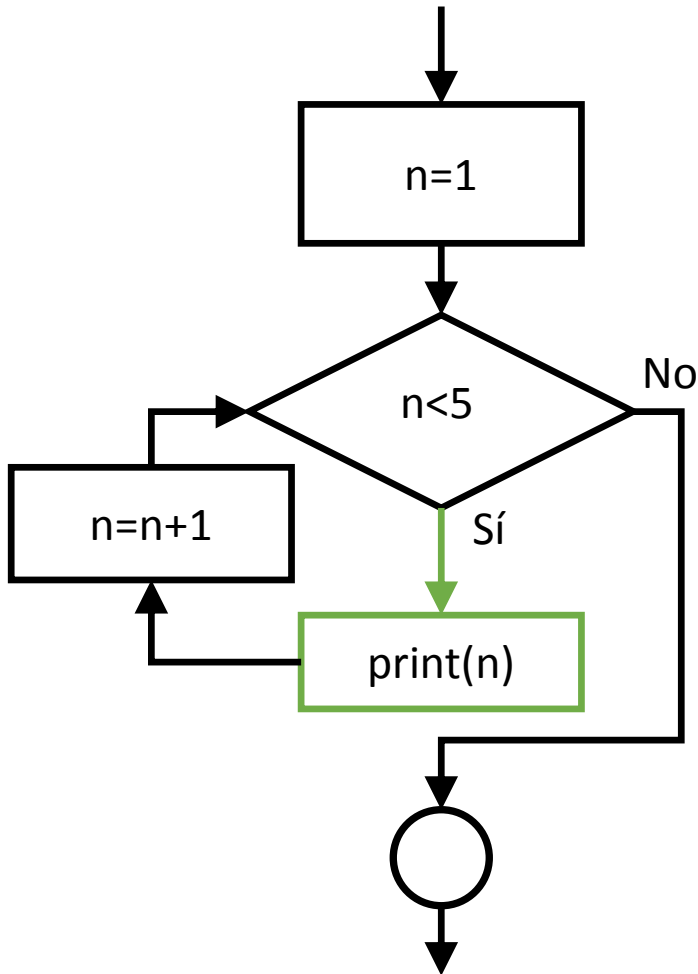
Variables:

Se comprueba si la variable  $n$  es inferior al valor final del rango, es decir 5.

Ejecución:



# Ciclo For: Ejemplo 1 (5)



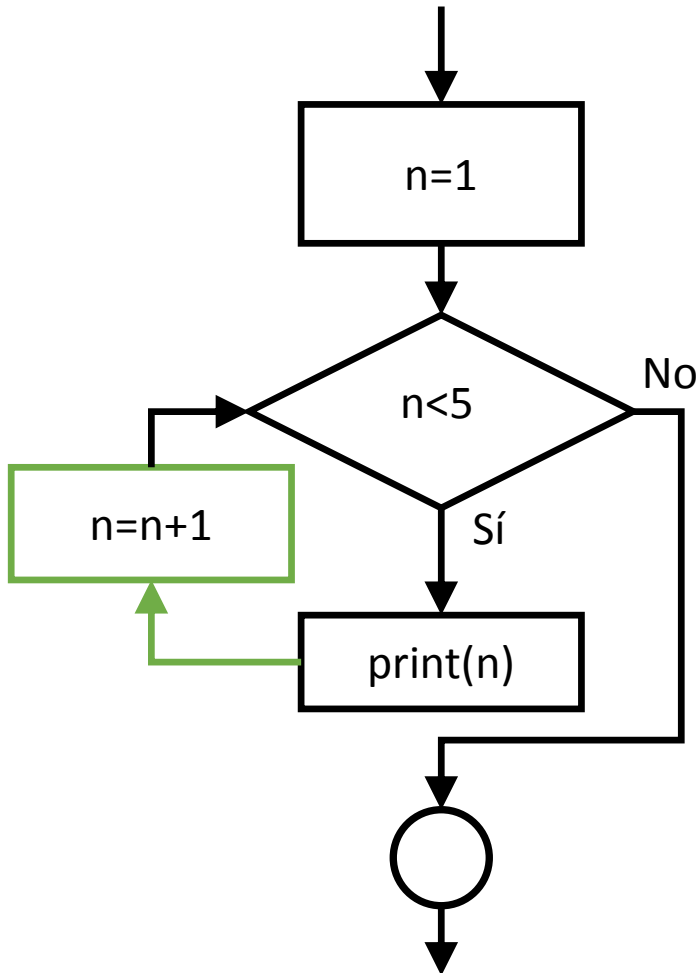
Variables:

Se cumple la condición y se ejecuta el código dentro del ciclo, imprimiendo el valor de  $n$  que es 1.

Ejecución:

1

# Ciclo For: Ejemplo 1 (6)



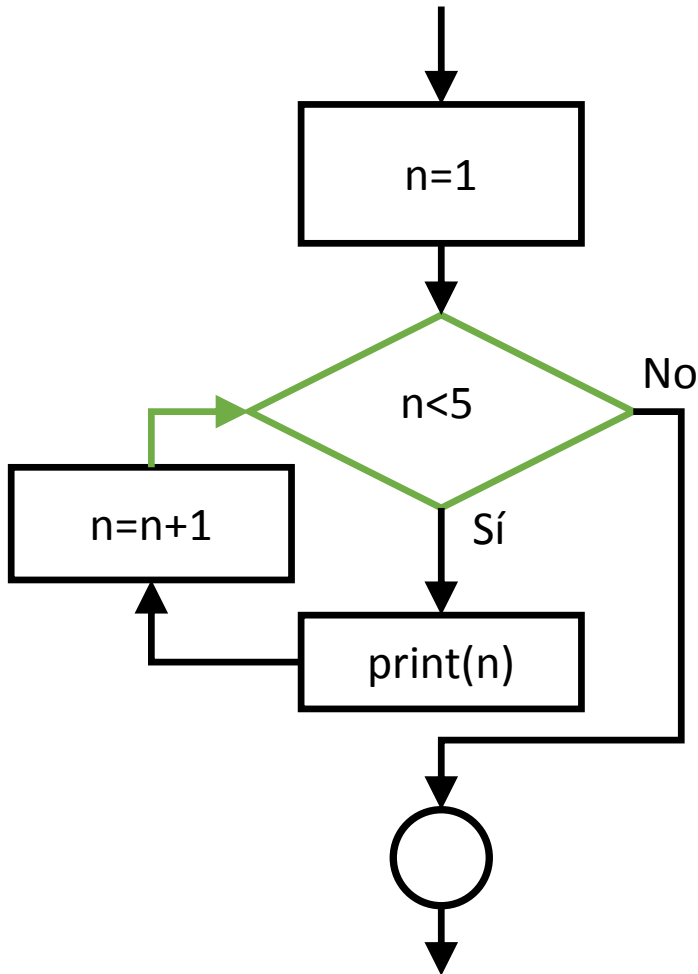
Variables:

La variable contadora  $n$  incrementa o disminuye  $c$  unidades, en este caso incrementa una unidad, entonces su nuevo valor es  $n=2$ .

Ejecución:

1

# Ciclo For: Ejemplo 1 (7)



Variables:

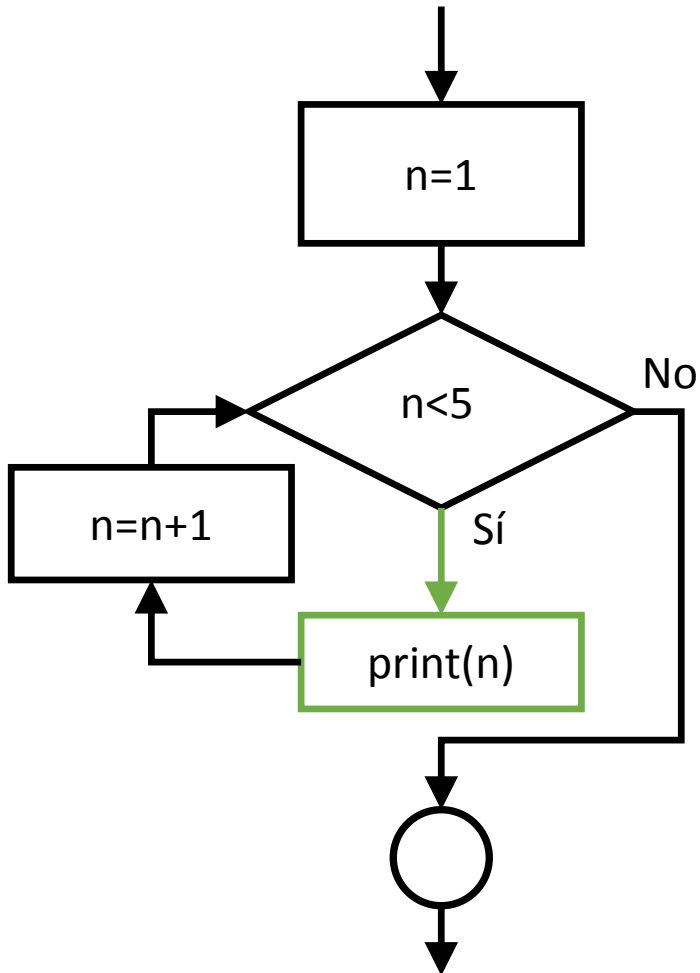
Nuevamente se comprueba que el valor de  $n$  esté dentro del rango establecido, en este caso  $n$  vale 2, luego es menor que 5, se cumple la condición.

Ejecución:

1



# Ciclo For: Ejemplo 1 (8)



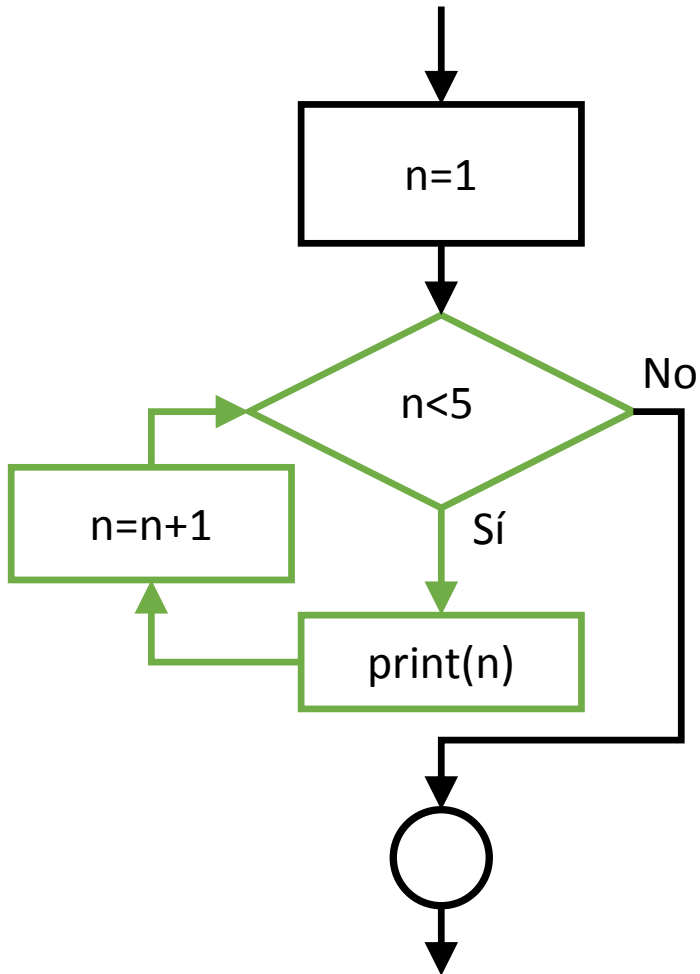
Variables:

Se cumple la condición y se ejecuta el proceso dentro del ciclo, imprimiendo el valor de  $n$  que es 2 y continúa el ciclo.

Ejecución:

1  
2

# Ciclo For: Ejemplo 1 (9)



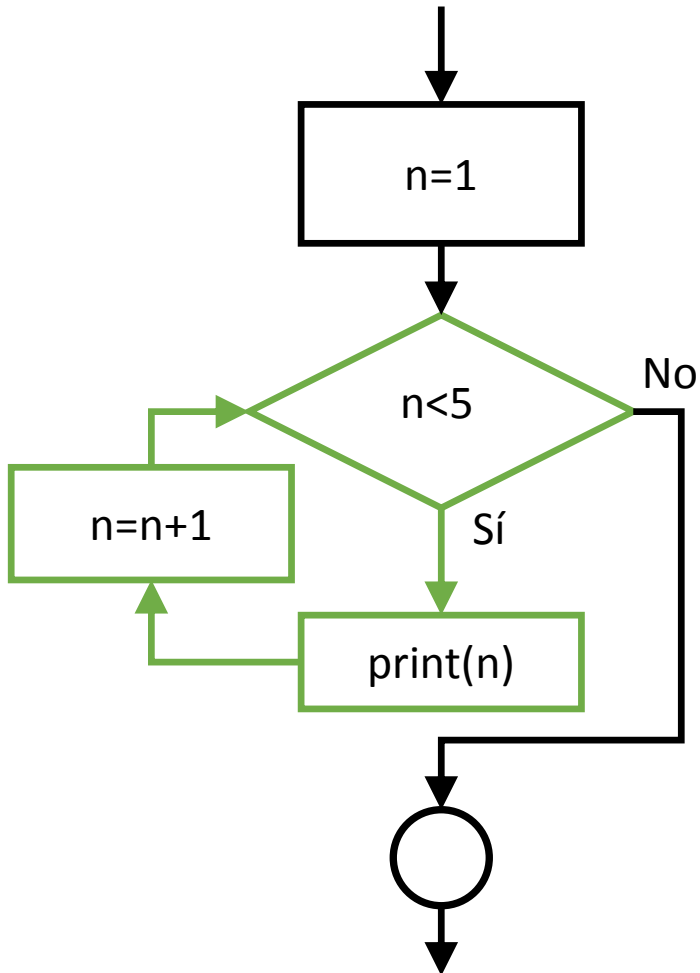
Variables:

n=2 (es menor que 5)  
a=1 (valor inicial de n)  
b=5 (límite del rango de n)  
c=1 (cambio de n por ciclo)  
n ya cumplió con la condición, se imprimió 2, n se incrementa en 1

Ejecución:

1  
2

# Ciclo For: Ejemplo 1 (10)



Variables:

$n=3$  (es menor que 5)

$a=1$  (valor inicial de  $n$ )

$b=5$  (límite del rango de  $n$ )

$c=1$  (cambio de  $n$  por ciclo)

$n$  cumple con la condición, se imprime  $n$  y se incrementa en 1

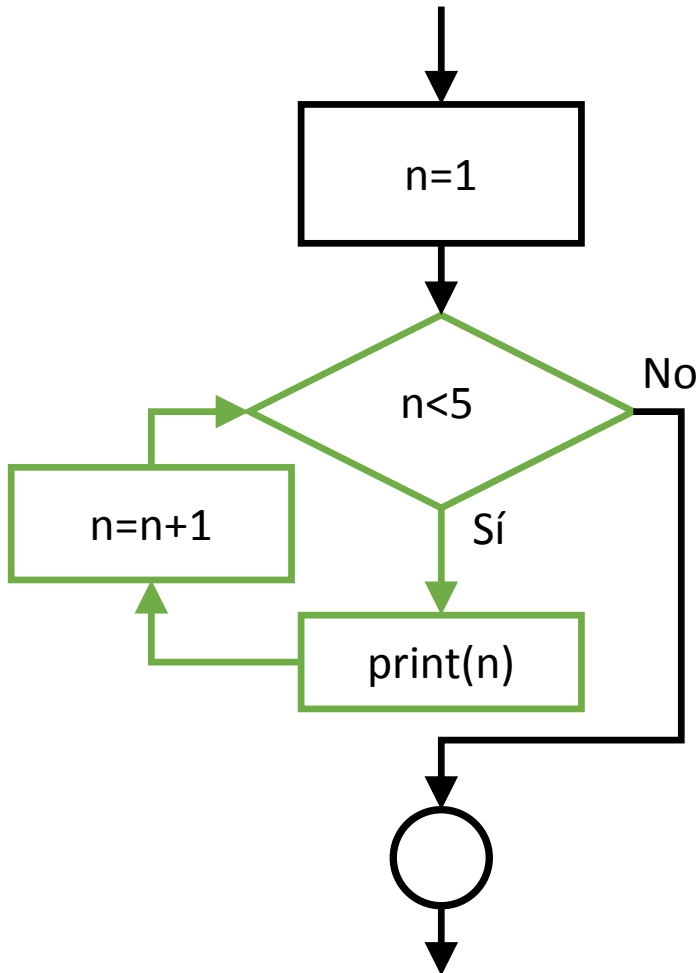
Ejecución:

1

2

3

# Ciclo For: Ejemplo 1 (11)



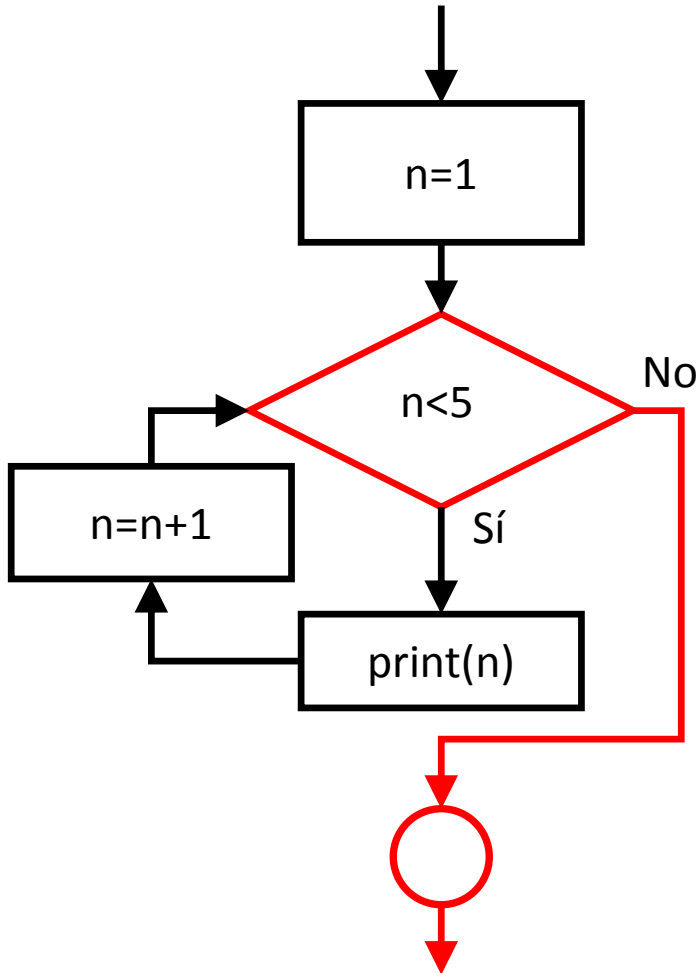
Variables:

n=4 (es menor que 5)  
a=1 (valor inicial de n)  
b=5 (límite del rango de n)  
c=1 (cambio de n por ciclo)  
n cumple con la condición, se imprime n y se incrementa en 1

Ejecución:

1  
2  
3  
4

# Ciclo For: Ejemplo 1 (12)



Variables:

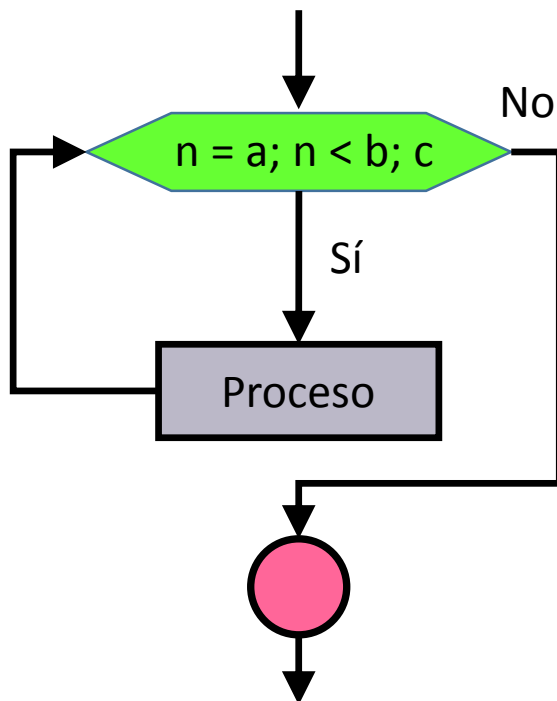
n=5 (es igual que 5)  
a=1 (valor inicial de n)  
b=5 (límite del rango de n)  
c=1 (cambio de n por ciclo)  
No cumple con la condición, el ciclo termina sin ejecutar la impresión

Ejecución:

1  
2  
3  
4

# Ciclo For

Diagrama de Flujo:



Existen varias formas de representar el ciclo for.

La figura muestra una forma más compacta: la inicialización del contador, evaluación del valor final, y el incremento del contador, están todos incluidos en el símbolo hexagonal.

Pseudocódigo:

Para n en el rango (a, b, c):  
Ejecute Proceso

# Ciclo For: Ejercicio 1

Desarrolle un programa en lenguaje Python que imprima los 10 primeros números impares.

# Ciclo For: Ejercicio 2

Desarrolle un programa en lenguaje Python que cuente cuántos múltiplos de 7 hay entre 3 y 49 inclusive.



# Ciclo For Anidado: Ejemplo 2

Código:

```
for filas in range (0,6,1):  
    for columnas in range (0,filas,1):  
        print ("*", end="")  
    print ()
```

Ejecución:

```
>>>
```

```
*
```

```
**
```

```
***
```

```
****
```

```
*****
```

```
>>> |
```

# Ciclo For:

## Ejercicio 3

Para cada una de las siguientes imágenes, desarrolle un programa en lenguaje Python que logre mostrarlas en consola.

```
*****  
*****  
*****  
*****  
*****
```

```
*****  
****  
***  
**  
*
```

```
*****  
****  
***  
**  
*
```

# Referencias Bibliográficas

- [1] Zelle, John M. Python Programming an Introduction to Computer Science. -- 2nd ed. -- Washington : Franklin, Beedle & Associates Inc, 2010.
- [2] Dawson, Michael. Python Programming for the Absolute Beginner. -- 3th ed. -- Australia : Cengage Learning, 2010.
- [3] Python Central. Python's range() Function Explained. [En línea] <http://goo.gl/6TecKf>. Consultado en Feb 2016.
- [4] Diana Teresa Gómez Forero. Presentación de clase: Ciclo For. Universidad Pontificia Bolivariana Seccional Bucaramanga. 2015.