

# Bucles for en Python

for los bucles repiten una porción de código para un conjunto de valores.

Como se explica en la [documentación de Python](#), los bucles for funcionan de manera ligeramente diferente a como lo hacen en lenguajes como JavaScript o C.

Un bucle for establece la variable iteradora en cada valor de una lista, arreglo o cadena proporcionada y repite el código en el cuerpo del bucle for para cada valor de la variable iteradora.

En el ejemplo a continuación, usamos un bucle for para imprimir cada número de nuestro arreglo.

```
# Ejemplo bucle for
for i in [1, 2, 3, 4]:
    print(i, end=", ") # prints: 1, 2, 3, 4,
```

También podemos incluir una lógica más compleja en el cuerpo de un bucle for. En este ejemplo imprimimos el resultado de un pequeño cálculo basado en el valor de nuestra variable iteradora.

```
# Ejemplo más complejo
for i in [1, 3, 5, 7, 9]:
    x = i**2 - (i-1)*(i+1)
    print(x, end=", ") # prints 1, 1, 1, 1, 1,
```

Cuando los valores de un arreglo para nuestro bucle for son secuenciales, podemos usar la función range() de Python en lugar de escribir el contenido de nuestro arreglo.

## La función range() en Python

La función range() proporciona una secuencia de enteros basada en los argumentos de la función. Se puede encontrar información adicional en la [documentación de Python](#) para la función range().

```
range(parar)
range(incio, parar[, paso])
```

El argumento inicio es el primer valor del rango. Si se llama a range() con un solo argumento, Python asume inicio = 0.

El argumento parar es el límite superior del rango. Es importante tener en cuenta que este valor superior no está incluido en el rango.

En el ejemplo a continuación, tenemos un rango que comienza en el valor predeterminado de 0 e incluye enteros menores que 5.

```
# Ejemplo con un argumento
for i in range(5):
    print(i, end=", ") # prints: 0, 1, 2, 3, 4,
```

En nuestro siguiente ejemplo, establecemos inicio = -1 y de nuevo incluimos enteros menores que 5.

```
# Ejemplo con dos argumentos
for i in range(-1, 5):
    print(i, end=", ") # prints: -1, 0, 1, 2, 3, 4,
```

El valor de paso opcional controla el incremento entre los valores del rango. De forma predeterminada, paso = 1.

En nuestro ejemplo final, usamos el rango de enteros de -1 a 5 y establecemos paso = 2.

```
# Ejemplo con tres argumentos
for i in range(-1, 5, 2):
    print(i, end=", ") # prints: -1, 1, 3,
```

## Resumen

En este artículo, vimos bucles for en Python y la función range().

Los bucles for repiten un bloque de código para todos los valores de una lista, arreglo, cadena o range().

Podemos usar una función range() para simplificar la escritura de un bucle for. Se debe especificar el valor de parar del range(), pero también podemos modificar el valor inicio y el paso entre enteros en el range().