

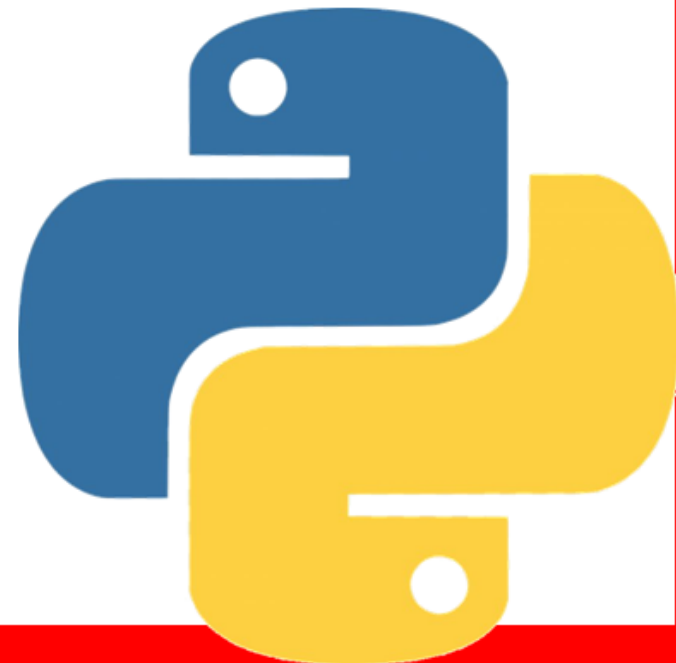


Ciclo for

09

Curso Básico

python



Recorre una lista, tupla, string, range,...

Ciclo for

Conceptos generales

- El bucle for, en Python, es aquel que nos permitirá **iterar sobre una variable compleja (compuesta)**, del tipo **lista, tupla, string, range,...**

```
mi_lista = ['Juan', 'Antonio', 'Pedro', 'Herminio']  
for nombre in mi_lista:  
    print(nombre)
```

Por cada nombre en mi_lista, imprimir nombre



Recorre una lista, **tupla**, string, range,...

Ciclo for

Conceptos generales

- El bucle for, en Python, es aquel que nos permitirá **iterar sobre una variable compleja (compuesta)**, del tipo **lista, tupla, string, range,...**

```
mi_tupla = ('rosa', 'verde', 'celeste', 'amarillo')  
for color in mi_tupla:  
    print(color)
```

Por cada color en mi_tupla, imprimir color



Recorre una lista, tupla, string, range,...

Ciclo for

Conceptos generales

- El bucle for, en Python, es aquel que nos permitirá **iterar sobre una variable compleja (compuesta)**, del tipo **lista, tupla, string, range,...**

```
acum=""  
for letra in "Hola Mundo":  
    acum+=letra  
    print(letra, " > ", acum)
```

```
H > H  
o > Ho  
l > Hol  
a > Hola  
 > Hola  
M > Hola M  
u > Hola Mu  
n > Hola Mun  
d > Hola Mund  
o > Hola Mundo
```



Recorre una lista, tupla, string, range,...

Ciclo for

Conceptos generales

- El bucle for, en Python, es aquel que nos permitirá **iterar sobre una variable compleja (compuesta)**, del tipo **lista, tupla, string, range,...**

```
for anio in range(2001, 2013):  
    print("Informes del Año", str(anio))
```

Por cada año en el rango 2001 a 2013, imprimir la frase "Informes del Año *año*"



La sentencia `for` en Python difiere un poco de lo que uno puede estar acostumbrado en lenguajes como C o Pascal. En lugar de siempre iterar sobre una progresión aritmética de números (como en Pascal) o darle al usuario la posibilidad de definir tanto el paso de la iteración como la condición de fin (como en C), la sentencia `for` de Python itera sobre los ítems de cualquier secuencia (una lista o una cadena de texto), en el orden que aparecen en la secuencia. Por ejemplo:

```
>>> # Midiendo cadenas de texto
... palabras = ['gato', 'ventana', 'defenestrado']
>>> for p in palabras:
...     print(p, len(p))
...
gato 4
ventana 7
defenestrado 12
```

For

range

- Si se necesita iterar sobre una secuencia de números, es apropiado utilizar la función integrada **range()**, la cual genera progresiones aritméticas:

```
>>> for i in range(5):  
...     print(i)  
...  
0  
1  
2  
3  
4
```

```
range(5, 10)  
5 through 9
```

```
range(0, 10, 3)  
0, 3, 6, 9
```

```
range(-10, -100, -30)  
-10, -40, -70
```



For

range() y len()

- Para iterar sobre los índices de una secuencia, puedes combinar **range()** y **len()** así:

```
>>> a = ['Mary', 'tenia', 'un', 'corderito']
>>> for i in range(len(a)):
...     print(i, a[i])
...
0 Mary
1 tenia
2 un
3 corderito
```



For

break y else

- La sentencia **break** termina el **ciclo** for o while más anidado.
- Los ciclos for y while pueden tener una cláusula **else** que es **ejecutada cuando el ciclo termina** (luego de agotar la lista con for) o cuando la condición se hace falsa (con while), **pero no cuando el ciclo es terminado con la sentencia break**.

```
>>> for n in range(2, 10):
...     for x in range(2, n):
...         if n % x == 0:
...             print(n, 'es igual a', x, '*', n/x)
...             break
...         else:
...             # sigue el bucle sin encontrar un factor
...             print(n, 'es un numero primo')
...
2 es un numero primo
3 es un numero primo
4 es igual a 2 * 2
5 es un numero primo
6 es igual a 2 * 3
7 es un numero primo
8 es igual a 2 * 4
9 es igual a 3 * 3
```

Fijate bien: el else pertenece al ciclo for, no al if.



For continue

- La declaración **continue**, continua con la siguiente iteración del ciclo:

```
>>> for num in range(2, 10):  
...     if num % 2 == 0:  
...         print("Encontré un número par", num)  
...         continue  
...     print("Encontré un número", num)  
Encontré un número par 2  
Encontré un número 3  
Encontré un número par 4  
Encontré un número 5  
Encontré un número par 6  
Encontré un número 7  
Encontré un número par 8  
Encontré un número 9
```





Ciclo for

09

Curso Básico

python

