PYTHON

Los bucles for y while – 100 días de Python #5

PROGRAMACIÓN FÁCIL
SEPTIEMBRE 5, 2022

Tabla de contenidos

¿Qué son los bucles?

Los bucles son estructuras de control de flujo, que permiten realizar repeticiones en el código. En Python tenemos varios tipos de bucles. Empecemos por el más común, el bucle for y luego pasemos al while.

El bucle for en Python

El siguiente ejemplo es un bucle for que se ejecuta 5 veces:

Python

```python
for i in range(5) :
    print(f"El valor del bucle es: {i}.")
```

Resultado en la consola

El valor del bucle es: 0.
El valor del bucle es: 1.
El valor del bucle es: 2.
El valor del bucle es: 3.
El valor del bucle es: 4.

El funcionamiento de esto es muy sencillo. «i» es una variable, se puede llamar como quieras, pero normalmente se llama «i» en los bucles de Python. También es común ver el nombre «x» de variable en el bucle.

Con «in range(5)» le indicamos que «i» se ejecute en un rango de 5 repeticiones. Esto le dará un valor inicial de 0 a la variable «i», que es el que se muestra en la primera ejecución.

Después de que se ejecute esas 5 veces, el bucle termina y el programa sigue con normalidad.

Esto es solo un print(), pero en un bucle podemos meter todo el código que queramos.

Podemos hacer rangos diferentes. Por ejemplo, empezar desde el valor 3 hasta el 7.

Python

```python
for i in range(3,7) :
    print(f"El valor del bucle es: {i}.")
```

Resultado en la consola

El valor del bucle es: 3.
El valor del bucle es: 4.
El valor del bucle es: 5.
El valor del bucle es: 6.

Cuando el valor de «i» llega a 7, este ya no se ejecuta y por lo tanto, no sale el print con el valor de 7.

El incremento del bucle va normalmente de uno en uno, pero se puede añadir otro parámetro más al range() para que los incrementos sean mayores.

Python

```python
for i in range(3, 12, 3) :
    print(f"El valor del bucle es: {i}.")
```

Resultado en la consola

El valor del bucle es: 3.
El valor del bucle es: 6.
El valor del bucle es: 9.

También se puede especificar un decremento, utilizando valores negativos:

Python

```python
for i in range(3, -12, -3) :
    print(f"El valor del bucle es: {i}.")
```

Resultado en la consola

El valor del bucle es: 3.
El valor del bucle es: 0.
El valor del bucle es: -3.
El valor del bucle es: -6.
El valor del bucle es: -9.

Va del 3 al -12 (-12 igual, no se muestra).

Bucle for para iterar listas

Con los bucles podemos hacer infinidad de cosas, iremos utilizándolos siempre a partir de este día. Una de estas cosas, es iterar listas o tuplas enteras, es decir, recorrer sus elementos de uno en uno.

```python
colores = ["rojo", "azul", "verde", "amarillo"]

print("---LISTADO DE COLORES---")

for color in colores:
    print(f"-{color}.")
```

Resultado en la consola

```
—LISTADO DE COLORES—
-rojo.
-azul.
-verde.
-amarillo.
```

Tal y como te he dicho, la «i» la podemos cambiar por lo que queramos. En este caso, «color» va a contener siempre un color, por eso, es un buen nombre.

Por cierto, el «in» lo utilizaremos para recorrer elementos iterables como una lista, tupla, un string...

¿Tiene posiciones, ergo es iterable (recorrible con un bucle)?

Saltar iteraciones en los bucles o romper su ejecución

Hay ocasiones, en las que queremos establecer posibles condiciones dentro de los bucles, para hacer que finalicen antes de tiempo o que salten ciertas iteraciones.

Continue

Con «continue», vamos a omitir ciertas iteraciones, no saldremos del bucle hasta que este finalice normalmente.

```python
colores = ["rojo", "azul", "verde", "amarillo"]

print("---LISTADO DE COLORES---")

for color in colores:
    if color == "azul" or color == "verde":
        continue
    print(f"-Color {color}.")
```

Resultado en la consola

```
—LISTADO DE COLORES—
-Color rojo.
-Color amarillo.
```

¿Ves?, en este caso, según una condición, estoy omitiendo que el bucle itere ciertas partes de la lista.

break

Con «break», cuando se cumpla la condición, se romperá la ejecución del bucle y saldremos de él. No se ejecutará el resto.

```python
colores = ["rojo", "azul", "verde", "amarillo"]

print("---LISTADO DE COLORES---")

for color in colores:
    if color == "azul":
        break
    print(f"-Color {color}.")
```

Resultado en la consola

```
Resultado en la consola

—LISTADO DE COLORES—
-Color rojo.
```

El bucle while

El bucle while nos permite hacer lo mismo que el for, pero con una sintaxis diferente.

En este caso, la variable de control va fuera del bucle. Le damos el valor que queramos (no tiene porqué ser 1) y se pone la condición de salida «i < 5».

Mientras que la condición de salida sea True, se va a ejecutar.

El incremento o decremento del bucle, se hace dentro de él. Con saltos del valor que queramos. Aquí se utilizan los operadores de asignación de incremento «+=» o asignación de decremento «-=».

Python

```python
# Variable para el bucle
i = 1

# Bucle While
while i < 5:
  print(f"El valor del bucle es: {i}.")
  i += 1
```

```
Resultado en la consola

El valor del bucle es: 1.
El valor del bucle es: 2.
El valor del bucle es: 3.
El valor del bucle es: 4.
```

¿Y el operador de incremento «++» y decremento «—«?

Para quienes ya programan en otros lenguajes de programación, puede que queráis en algún momento incrementar el valor de «i» en el bucle con «i++» o decrementar con «i—«.

En Python, no existe el operador «++» o «—«. Al usar esto, estaremos utilizando dos operadores de «suma», «resta» o concatenación.

Si no programas todavía, no le des mucha importancia a esto.

El bucle do while en Python

Python no tiene bucle do while. Esto puede ser un fastidio a veces, ya que este bucle, en otros lenguajes de programación, previene el siguiente error:

Un supuesto programa que depende de que se ejecute al menos una vez el bucle. El bucle puede ser True a veces y otras False. Pero si no se ejecuta al menos una vez, el programa no puede continuar porque depende de algo del bucle.

De hecho, el programa del proyecto, va a requerir esta mecánica.

La solución, no hacer el programa dependiente de ningún bucle (siempre que sea posible), si se ejecuta, bien, si no, le damos otra salida alternativa.

Los bucles infinitos

O también se puede emular para que su comportamiento sea el mismo:

Python

```python
while True:
    salida = input("Introduce 'salir' para finalizar.\n").lower()
    if salida == 'salir':
        break
```

La condición del while es True. Lo que hace que el bucle sea infinito. Siempre se cumple su condición y no depende de nada.

Si ponemos lo que sea en la consola, se va a seguir ejecutando. En cambio, si ponemos «salir», se va a romper la ejecución del bucle.

Como puedes ver, de cualquier forma, se va a ejecutar mínimo una vez, que es lo que esperamos de un bucle do while.

Por cierto, el método lower() aplicado al input(), hace que lo el string que introduce el usuario se transforme en minúsculas, por lo que aunque ponga, por ejemplo, «SALIR» o «Salir», lo que se genera siempre, es «salir».

Aquí tienes los ejercicios y proyecto correspondientes a este quinto día del curso.

+4

EN CURSO PYTHON, DO WHILE PYTHON, LOS BUCLES PYTHON, PYTHON, PYTHON EN ESPAÑOL, WHILE TRUE PYTHON

DEJA UNA RESPUESTA

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario *

Nombre * Correo electrónico * Web

Guarda mi nombre, correo electrónico y web en este navegador para la próxima vez que comente.

Publicar el comentario

Entrada relacionada

CUSTOMTKINTER PYTHON TKINTER

Cómo crear un menú superior con Tkinter: guía paso a paso

PROGRAMACIÓN FÁCIL
MARZO 24, 2023

CUSTOMTKINTER MYSQL PYTHON TKINTER

Cómo crear una interfaz gráfica en Python para visualizar bases de datos de MySQL

PROGRAMACIÓN FÁCIL
MARZO 15, 2023

CUSTOMTKINTER MYSQL PYTHON TKINTER

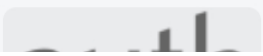
Cómo crear una interfaz gráfica para consultas SQL en Python: Tutorial paso a paso

PROGRAMACIÓN FÁCIL
MARZO 8, 2023

CUSTOMTKINTER PYTHON TKINTER

Clase para funciones del programa con TK y CTK

PROGRAMACIÓN FÁCIL
MARZO 4, 2023



CUSTOMTKINTER PYTHON TKINTER

Bucle autogenerador de BOTONES y DESTRUIR ventanas con Tkinter y CustomTkinter

PROGRAMACIÓN FÁCIL
MARZO 2, 2023

CUSTOMTKINTER PYTHON TKINTER

Ventanas secundarias TopLevel con Tkinter y CustomTkinter

PROGRAMACIÓN FÁCIL
FEBRERO 27, 2023

Subir arriba ↑