

Python



Primeros Pasos

Parte IV

Algoritmos y Programación I - FIUBA
Lic. Gustavo Bianchi

1

Estructuras de Control Iterativas

- Con qué instrucciones podremos repetir un grupo de acciones, y cuáles son sus diferencias
- Cómo y cuándo usar While
- La función range()
- Cómo y cuándo usar For

Algoritmos y Programación I
Lic. Gustavo Bianchi

2

Estructura de Control Iterativas

Qué uso para iterar?

Para lograr repetir la ejecución de un conjunto de acciones, un determinado número de veces, ó mientras una condición sea verdadera, disponemos de dos instrucciones de tipo iterativas, **for** y **while**.

for

Para recorrer una secuencia de elementos, y por cada elemento de la secuencia; ejecutar una ó más acciones.

Sabemos cuántos ciclos debemos realizar

while

Cuando queramos repetir una ó más acciones, **mientras una condición sea verdadera**, y no sabemos en qué momento se hará falsa.

Desconocemos cuándo sucederá el evento que hará falsa la condición

Algoritmos y Programación I
Lic. Gustavo Bianchi

3

Estructura de Control Iterativas - while

Las acciones dentro del **while**, se ejecutaran repetidamente, **mientras la condición sea verdadera**. Como desconocemos cuando la condición será falsa, **tenemos que asegurarnos, que dentro de las acciones, al menos una, en algún momento, hará que la condición sea falsa**.

Ejemplo: Solicitar el ingreso de un número, repitiendo esta acción, siempre y cuando el número ingresado sea distinto a cero. Al final, mostrar el total acumulado resultante de sumar, los números ingresados.

| while | |
|--|---|
| Formato | Ejemplo |
| <pre> while condición : acción 1 acción 2 acción N Instrucción siguiente al while </pre> | <pre> total = 0 valor = int(input("Valor: ")) while valor != 0 : total = total + valor valor = int(input("Valor:")) print("Total Acumulado:", total) </pre> |

Algoritmos y Programación I
Lic. Gustavo Bianchi

4

Otro ejemplo con while

Ejemplo: Solicitar el ingreso de un número, asegurando que el número ingresado sea mayor ó igual a cero, y menor a 1000; de lo contrario enviar mensaje de error al usuario y solicitar que reingrese el número. Repetir el ingreso hasta que el número sea válido. Muestre al final, el valor aceptado.

```
numero = int(input("Número: "))

while (numero < 0) or (numero >= 1000) :

    print("ERROR: Valor ingresado inválido, reingrese valor")

    numero = int(input("Número: "))

print("Número válido ingreado:", numero)
```

Algoritmos y Programación I
Lic. Gustavo Bianchi

5

Función range()

Nos permitirá generar una **secuencia ordenada de números enteros**

Con 1 parámetro, indicando hasta que valor (sin incluirlo), partiendo siempre de cero

range(5) → 0, 1, 2, 3, 4

Con 2 parámetros, el 1ro. para decir desde que valor (incluido), y el 2do. hasta que valor (sin incluirlo)

range(1, 11) → 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Con 3 parámetros, los 2 primeros, igual al caso anterior; el 3ro. Indica el incremento a aplicar a cada valor para obtener el siguiente

range(2, 9, 3) → 2, 5, 8

Con 3 parámetros, para generar una secuencia decreciente, el 3ro. debe ser negativo, el 1ro. mayor al 2do.

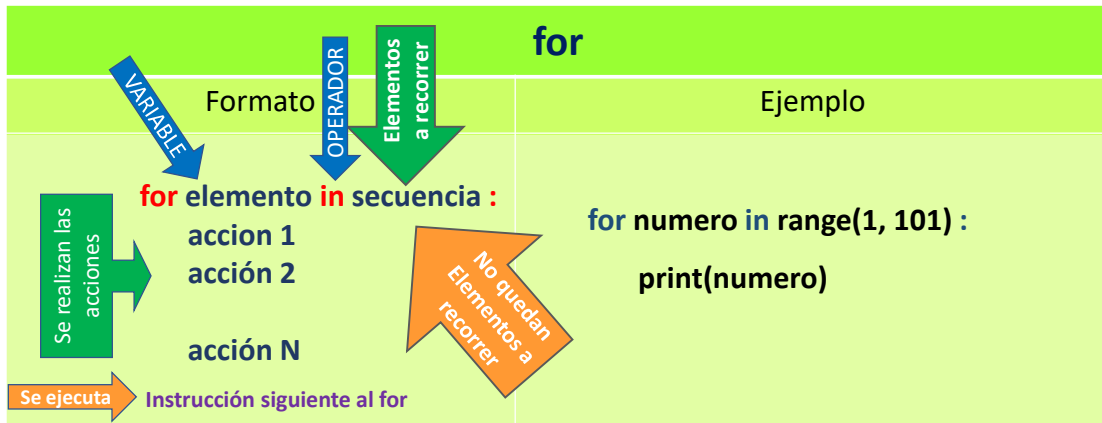
range(5, 0, -1) → 5, 4, 3, 2, 1

Algoritmos y Programación I
Lic. Gustavo Bianchi

6

El uso del ciclo **for** en Python, es más amplio al que podemos encontrar en otros lenguajes convencionales. Comenzaremos viendo un uso de aplicación muy simple y similar al utilizado en otros lenguajes. Entonces, en esta primera instancia, utilizaremos un ciclo **for** para recorrer una secuencia de elementos, de a uno, y por cada elemento, ejecutar una ó más acciones.

Ejemplo: Imprimir por pantalla los números del 1 al 100 inclusive.



Algoritmos y Programación I
Lic. Gustavo Bianchi

7

Otro ejemplo con for

Ejemplo: Solicitar el ingreso de un número, a continuación calcular la suma de todos los números pares entre 2, y el número ingresado (inclusive); e informar el valor obtenido.

```
total_pares = 0
numero = int(input("Número: "))
for par in range(2, numero+1, 2) :
    total_pares = total_pares + par
print("Suma total de números pares entre 2 y", numero, "=", total_pares)
```

An annotation box with a green arrow points to the line `total_pares = total_pares + par`, containing the text **total_pares += par**.

Algoritmos y Programación I
Lic. Gustavo Bianchi

8