TIC

Módulo 1 - Unidad 2 - Lección 1

Actividad 5

Ciclos en Python

TALENTO TECH

UTP Universidad Tecnológica de Pereira | faceIT

Ciclos en Python

Introducción a las estructuras de control de flujo, destacando los bucles for y while. Ejemplos prácticos de cómo utilizar ciclos para procesar datos.

El bucle for.

En Python se utiliza para iterar sobre una secuencia (que puede ser una lista, una tupla, un diccionario, un conjunto o cualquier objeto iterable), permite ejecutar un bloque de códigos repetidamente para cada elemento en la secuencia.

Estructura Básica del Ciclo For:

```
for variable in secuencia:
    # Cuerpo del bucle
    # Se ejecutará para cada elemento en la secuencia
    # Puedes usar la variable para acceder al elemento
    # ...
```

Ciclos en Python

Variable:

Es la variable que toma el valor de cada elemento en la secuencia en cada iteración del bucle.

Secuencia: es la colección de elementos sobre la cual se iterará.

Ejemplo Práctico:

```
frutas = ["manzana", "plátano", "uva"]

for fruta in frutas:
    print(fruta)
```

En este ejemplo, el bucle for itera sobre la lista frutas, y en cada iteración, la variable fruta toma el valor de uno de los elementos de la lista. El bloque de código dentro del bucle (print(fruta)) se ejecuta para cada elemento, imprimiendo cada fruta en una línea.

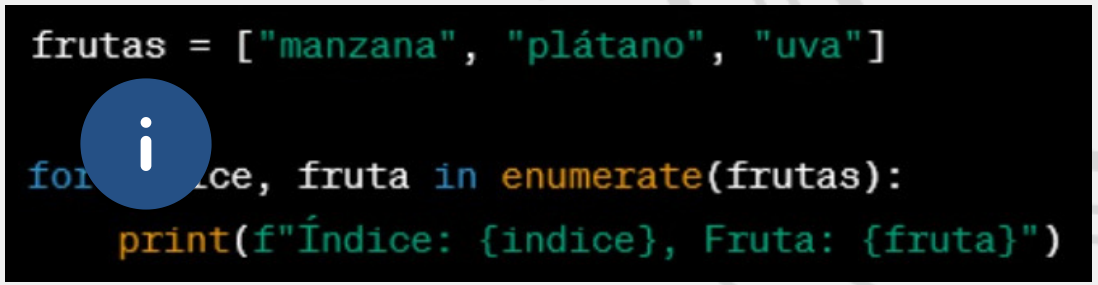
Ciclos en Python

Funciones Relacionadas con el Ciclo For:

range(): La función range() se utiliza comúnmente con el bucle for para generar una secuencia de números. Pueden proporcionar uno, dos o tres argumentos a range().

range(5): genera una secuencia de 0 a 4.
range(2, 8): genera una secuencia de 2 a 7.
range(1, 10, 2): genera una secuencia de 1 a 9 con un paso de 2.

enumerate(): la función enumerate() se utiliza para obtener tanto el índice como el valor de los elementos en una secuencia.



Notas Adicionales:

Pueden usar cualquier nombre para la variable en el bucle (for elemento in secuencia).
Los dos puntos (:) indican el inicio de un bloque de código que se ejecutará en cada iteración.
La indentación es crucial en Python y de ne el alcance del bloque de código dentro del bucle.

El bucle for es una herramienta poderosa para procesar elementos en una secuencia y es ampliamente utilizado en programación Python.

Ciclos en Python

El bucle while En Python se utiliza para repetir un bloque de códigos mientras una condición sea verdadera. A diferencia del bucle for, que itera sobre una secuencia específica, el bucle while se ejecutará mientras la expresión condicional sea evaluada como True.

Estructura Básica del Ciclo While:

```
while condicion:
    # Cuerpo del bucle
    # Se ejecutará mientras la condición sea verdadera
    # ...
```

Ciclos en Python

Condición: Es una expresión booleana que determina si el bucle debe continuar ejecutándose.

Ejemplo Práctico:

```
contador = 0

while contador < 5:
    print(f"Contador: {contador}")
    contador += 1
```

En este ejemplo, el bucle while se ejecutará mientras la variable contador sea menor que 5. En cada iteración, se imprime el valor actual de contador y se incrementa en 1, el bucle se detendrá cuando la condición (contador < 5) sea falsa.

Ciclos en Python

Funciones Relacionadas con el Ciclo

While: La palabra clave break se utiliza para salir del bucle while antes de que la condición sea

Break:
falsa. Puede usarse con una condición específica dentro del bucle.

En este ejemplo, el bucle while se ejecuta indefinidamente hasta que la condición contador >= 5 se cumple, momento en el que se utiliza break para salir del bucle.

```
contador = 0

while True:
    print(f"Contador: {contador}")
    contador += 1
    if contador >= 5:
        break
```

Ciclos en Python

Continue: La palabra clave continue se utiliza para pasar a la siguiente iteración del bucle sin ejecutar el resto del código dentro del bucle en esa iteración.

```
contador = 0

while contador < 5:
    contador += 1
    if contador == 3:
        continue
    print(f"Contador: {contador}")
```

En este ejemplo, cuando el contador es igual a 3, continue salta a la siguiente iteración sin imprimir el mensaje.

Notas Adicionales:

La condición se evalúa antes de cada iteración, y el bucle se ejecutará mientras la condición sea verdadera. Asegúrate de que la condición eventualmente se vuelva falsa para evitar bucles in nitos.

El bucle while es útil cuando no sabes cuántas veces se repetirá el bloque de código y dependes de una condición para decidir cuándo detener la ejecución.