

LENGUAJES DE PROGRAMACIÓN

PYTHON





Sintaxis de Estructuras de Control en Python

La programación en Python ofrece diversas herramientas y estructuras de control que nos permiten controlar el flujo de ejecución de un programa. Estas estructuras de control nos permiten tomar decisiones, repetir acciones y crear lógica condicional en nuestros programas. En esta lección, exploraremos la sintaxis de las estructuras de control en Python y cómo utilizarlas para construir programas más dinámicos y flexibles.





Estructura de Control IF

La estructura de control if se utiliza para tomar decisiones basadas en condiciones. Nos permite ejecutar un bloque de código si una condición es verdadera. La sintaxis básica de un if en Python es la siguiente:

if condicion:

bloque de código a ejecutar si la condición es verdadera

Por ejemplo, supongamos que queremos verificar si un número es positivo. Podríamos escribir el siguiente código:

```
numero = 10

if numero > 0:
    print("El número es positivo")
```

En este ejemplo, la condición `numero > 0` se evalúa como verdadera, por lo tanto, el bloque de código dentro del if se ejecuta y se imprime el mensaje "El número es positivo".





Además del bloque if, también podemos utilizar la estructura de control else para especificar un bloque de código alternativo que se ejecutará si la condición es falsa. Por ejemplo:

```
numero = -5

if numero > 0:
    print("El número es positivo")
else:
    print("El número es negativo o cero")
```

En este caso, la condición `numero > 0` se evalúa como falsa, por lo tanto, el bloque de código dentro del else se ejecuta y se imprime el mensaje "El número es negativo o cero".





Estructura de Control IF-ELSE IF-ELSE

En ocasiones, es posible que tengamos múltiples condiciones y queramos evaluar diferentes bloques de código según estas condiciones. En esos casos, podemos utilizar la estructura de control if-elif-else. La sintaxis básica es la siguiente:

if condicion1:

bloque de código a ejecutar si la condicion1 es verdadera

elif condicion2:

bloque de código a ejecutar si la condicion1 es falsa y la condicion2 es verdadera

else:

bloque de código a ejecutar si ninguna de las condiciones anteriores es verdadera

Por ejemplo, supongamos que queremos clasificar un número en diferentes categorías: positivo, negativo o cero. Podríamos escribir el siguiente código:

```
if numero > 0:  
    print("El número es positivo")  
elif numero < 0:  
    print("El número es negativo")  
else:  
    print("El número es cero")
```



En este ejemplo, se evalúa la primera condición $\text{numero} > 0$ y, si es verdadera, se ejecuta el bloque de código correspondiente. Si la condición es falsa, se pasa a evaluar la siguiente condición $\text{numero} < 0$ y, si es verdadera, se ejecuta el bloque de código correspondiente. Si ninguna de las condiciones anteriores es verdadera, se ejecuta el bloque de código en el `else`.

Es importante tener en cuenta que, una vez que se encuentra una condición verdadera y se ejecuta su bloque de código correspondiente, el flujo de ejecución sale de la estructura `if-elif-else` y continúa con el resto del programa.





Estructura de Control WHILE

La estructura de control while se utiliza para repetir un bloque de código mientras una condición sea verdadera. La sintaxis básica de un while en Python es la siguiente:

while condicion:

bloque de código a repetir mientras la condición sea verdadera

Por ejemplo, supongamos que queremos imprimir los primeros 5 números naturales. Podríamos escribir el siguiente código:

```
contador = 1

while contador <= 5:
    print(contador)
    contador += 1
```

En este ejemplo, la condición `contador <= 5` se evalúa como verdadera, por lo tanto, se ejecuta el bloque de código dentro del while, que imprime el valor actual del contador y luego incrementa su valor en 1. Esto se repite hasta que la condición sea falsa, es decir, cuando el contador alcanza el valor 6.

Es importante tener cuidado al utilizar estructuras de control while, ya que si la condición nunca se vuelve falsa, el programa quedará atrapado en un bucle infinito y no se ejecutará más allá de esa parte del código.



Estructura de Control FOR

La estructura de control for se utiliza para iterar sobre una secuencia de elementos, como una lista o una cadena de caracteres. La sintaxis básica de un for en Python es la siguiente:

for variable in secuencia:

bloque de código a ejecutar para cada elemento de la secuencia

Por ejemplo, supongamos que queremos imprimir los elementos de una lista de frutas. Podríamos escribir el siguiente código:

```
frutas = ["manzana", "banana", "naranja"]  
  
for fruta in frutas:  
    print(fruta)
```

En este ejemplo, el for itera sobre cada elemento de la lista "frutas" y ejecuta el bloque de código dentro del for para cada elemento. En cada iteración, el valor del elemento actual se asigna a la variable "fruta", que luego se utiliza para imprimir el nombre de la fruta..



La estructura de control for también se puede utilizar con la función range() para generar una secuencia de números en un rango específico. Por ejemplo:

```
for i in range(1, 6):  
    print(i)
```

En este caso, el for itera sobre los números del 1 al 5 y los imprime en cada iteración.

En esta lección, hemos explorado la sintaxis de las estructuras de control en Python. Hemos aprendido a utilizar las estructuras if, if-else, if-elif-else, while y for para controlar el flujo de ejecución de un programa. Estas estructuras nos permiten tomar decisiones, repetir acciones y crear lógica condicional en nuestros programas.

Recuerda practicar con ejemplos adicionales y experimentar con diferentes combinaciones de estructuras de control para fortalecer tu comprensión de la sintaxis de Python. En las siguientes lecciones, exploraremos conceptos más avanzados, como funciones y estructuras de datos, que nos permitirán construir programas más complejos y completos.