**¿Qué es el módulo en Python?**

El módulo en Python, representado por el operador %, te devuelve el **resto** de una división entera. Es decir, te indica qué número sobra después de dividir dos números enteros.

**¿Para qué sirve?**

* **Determinar si un número es par o impar:** Si el resto de dividir un número entre 2 es 0, el número es par.
* **Generar secuencias repetitivas:** Por ejemplo, para imprimir un mensaje cada 5 iteraciones de un bucle.
* **Convertir números a diferentes bases:** Como convertir un número decimal a binario.
* **Validar formatos de datos:** Por ejemplo, verificar si un número de tarjeta de crédito tiene una longitud válida.

**¿Cómo se utiliza?**

La sintaxis es muy sencilla:

Python

numero1 % numero2

Use code [with caution.](file:///C:/faq#coding)

Donde:

* numero1: El dividendo.
* numero2: El divisor.

**Ejemplos:**

1. **Determinar si un número es par o impar:**

Python

numero = 15if numero % 2 == 0: print("El número es par") else: print("El número es impar")

Use code [with caution.](file:///C:/faq#coding)

1. **Generar una secuencia repetitiva:**

Python

for i in range(1, 11): if i % 5 == 0: print("¡Hola!") else: print(i)

Use code [with caution.](file:///C:/faq#coding)

1. **Convertir un número decimal a binario (simplificado):**

Python

numero\_decimal = 13while numero\_decimal > 0: residuo = numero\_decimal % 2 print(residuo, end='') numero\_decimal //= 2 # División entera para obtener el cociente

Use code [with caution.](file:///C:/faq#coding)

**Otros usos comunes:**

* **Validar formatos de fecha:** Verificar si un año es bisiesto.
* **Crear patrones:** Generar secuencias de números o caracteres con patrones específicos.
* **Resolver problemas algorítmicos:** Como el problema del máximo común divisor.

**¡Practica!**

La mejor manera de entender el módulo es experimentando. Prueba diferentes números y observa los resultados. Puedes crear programas más complejos combinando el módulo con otras operaciones y estructuras de control.