

*args

En Python, `*args` es un parámetro especial que se utiliza en la definición de una función para indicar que puede recibir un número variable de argumentos posicionales. La palabra clave `args` es solo una convención y puedes elegir cualquier otro nombre, pero el asterisco (*) es lo que indica que se trata de un parámetro especial.

Aquí tienes un ejemplo que muestra cómo utilizar `*args` en una función:

```
def sumar(*args):  
    """  
    Función que suma un número variable de argumentos.  
    :param args: Argumentos posicionales.  
    :return: Suma de los argumentos.  
    """  
  
    resultado = 0  
    for numero in args:  
        resultado += numero  
    return resultado  
  
# Ejemplo de uso de la función  
resultado = sumar(2, 4, 6, 8)  
print(resultado) # Imprime: 20  
  
resultado = sumar(1, 3, 5, 7, 9)  
print(resultado) # Imprime: 25
```

En este ejemplo, la función `sumar` utiliza `*args` como parámetro. Esto significa que la función puede recibir cualquier número de argumentos posicionales al llamarla. Dentro de la función, los argumentos se tratan como una tupla, lo que permite iterar sobre ellos y realizar alguna operación, como en el ejemplo donde se realiza la suma de los argumentos.

Al llamar a la función `sumar`, puedes pasar cualquier cantidad de argumentos separados por comas. Estos argumentos se agruparán en una tupla y se procesarán dentro de la función.

El uso de `*args` en una función es útil cuando no se sabe de antemano cuántos argumentos serán pasados o cuando se desea proporcionar flexibilidad al usuario de la función para pasar una cantidad variable de argumentos.

Es importante tener en cuenta que `*args` solo captura argumentos posicionales. Si necesitas capturar argumentos con nombre, puedes utilizar `**kwargs`, que es otro parámetro especial que se utiliza para representar un número variable de argumentos de palabras clave.