

Funciones anónimas

Ejercicio: Calcular el producto de todos los elementos de una lista mediante una función *lambda* (uso de *reduce*)

- *reduce()* es una función del módulo *functools* que se usa para aplicar una función a los elementos de un iterable de manera acumulativa. La función toma dos argumentos y combina los resultados de las aplicaciones sucesivas de esta función.*

Sintaxis:

- *reduce(funcion, iterable, initializer=None)*

En este ejemplo, se utiliza la función **reduce** junto con una función anónima (**lambda**) para calcular el producto acumulado de los elementos de una lista.

1. Importar **reduce**:

- **reduce** se importa desde el módulo **functools**:

```
from functools import reduce
```

2. Definición de la Lista de Números:

- Se define una lista de números enteros

3. Definición de la Función Lambda para el Producto:

- La función **lambda** se define para calcular el producto de dos números: Aquí, **lambda** define una función anónima que toma dos argumentos **x** e **y** y retorna su producto (**x * y**).

4. Aplicación de **reduce**:

- Se usa **reduce** para aplicar la función **producto** a la lista **numeros**, acumulando el resultado: **reduce** toma la lista **numeros** y aplica la función **producto** de manera acumulativa. En otras palabras, **reduce** multiplica los elementos de la lista en el siguiente orden: (((1 * 2) * 3) * 4).

5. Impresión del Resultado:

- Finalmente, se imprime el resultado: Esto muestra el producto acumulado de los números en la lista.

```
from functools import reduce

numeros = [1,2,3,4]

producto = lambda x, y : x * y

resultado = reduce(producto, numeros)

print(resultado)
```

24