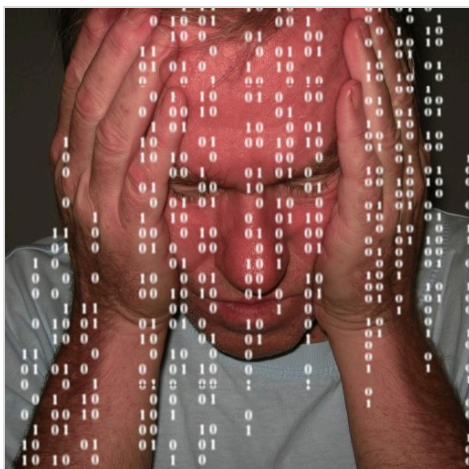


L U M B R E



Comprensiones y funciones lambda

Grado: 2

Materia: Programación con Python para Datos · Programación

11/10/2026

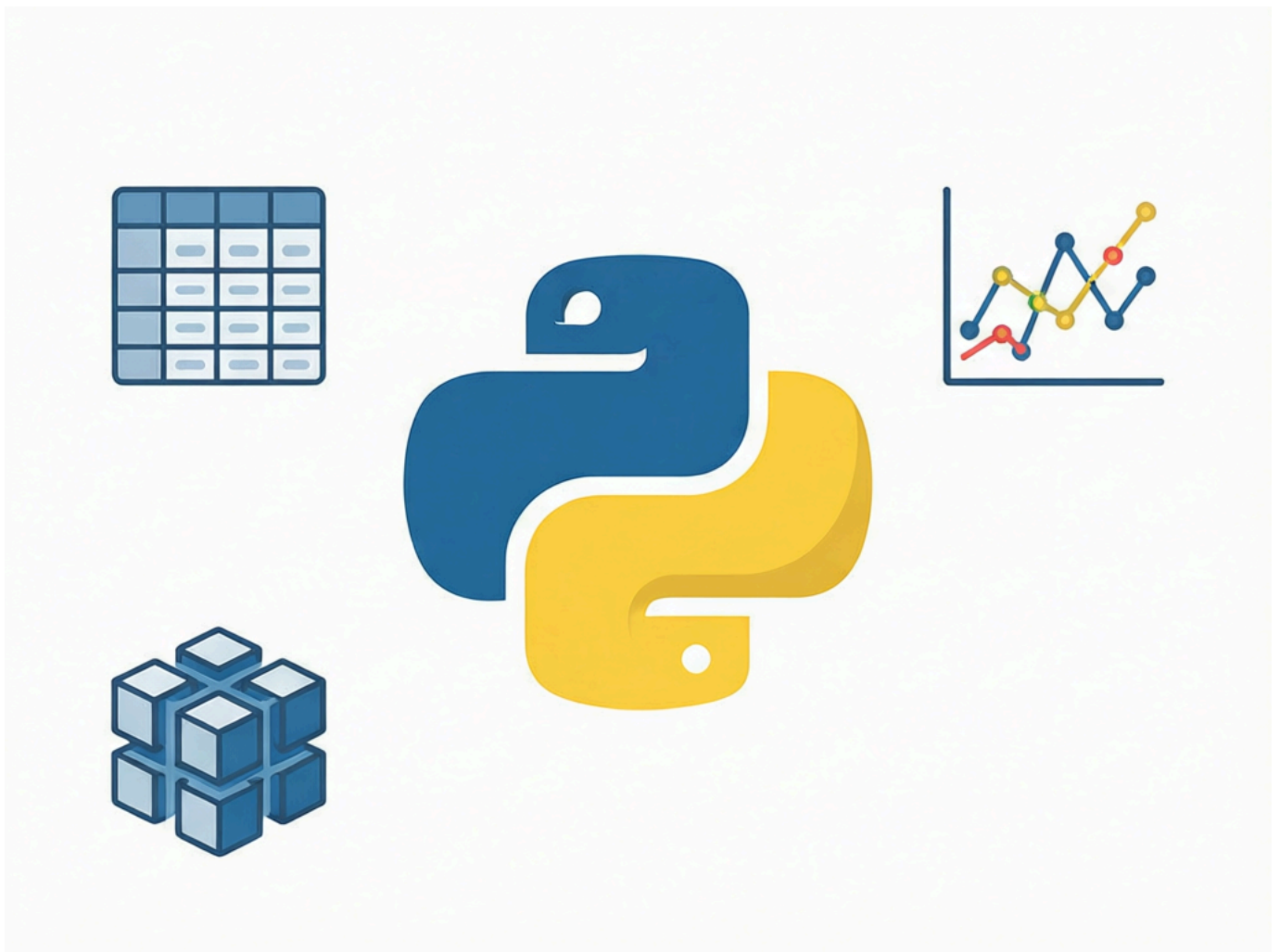
Antes de empezar

Las comprensiones construyen listas, diccionarios y conjuntos declarativamente; las lambda son funciones expresas. Bien usadas limpian el código; en exceso, lo vuelven indescifrable.

QUÉ APRENDERÁS

- Escribir código Python expresivo con comprensiones y funciones lambda, y saber cuándo favorecen o dañan la legibilidad.

Comprensiones y funciones lambda



Python para ciencia de datos: NumPy, pandas y manipulación de datos.

Mapa conceptual

- **Python para Datos**
 - Entorno
 - pip

- venv
- Reproducibilidad
- Funcional
 - Comprensiones
 - Lambda
 - map filter
- Calidad
 - try except
 - pytest
 - Asserts
- NumPy
 - Arrays
 - Broadcasting
 - Indexacion
- Pandas
 - DataFrame
 - Groupby
 - Merge Join

El mapa mental interactivo es interactivo en la versión online.



Del bucle al «qué»

En lugar de un bucle con append, una comprensión dice qué quieres en una sola línea. Es más cercano al «qué» y menos al «cómo». Solo recuerda: legibilidad primero.

Comprensión de lista

Una comprensión construye una lista aplicando una expresión a cada elemento de un iterable, con opción de filtrar. Es la forma idiomática de mapear y filtrar sin escribir el bucle y el append a mano.

Dict y set

- Comprensión de diccionario: pares clave-valor generados en línea.

- Comprensión de conjunto: igual que la de lista, pero produce un set sin duplicados.

Funciones lambda

Una lambda define una función anónima de una sola expresión, usable donde hace falta un objeto invocable sin declararla con `def`: típico al ordenar con clave o al pasar una transformación breve.

Construcciones declarativas que reemplazan bucles manuales.

Estructura	Ejemplo (idea)	Devuelve
List comprehension	<code>[x*x para x en nums]</code>	lista de cuadrados
Con filtro	<code>[x para x en nums si x > 0]</code>	solo los positivos
Dict comprehension	<code>k: len(k) para k en keys</code>	diccionario de longitudes
Lambda	<code>lambda p: p[1] para ordenar</code>	función clave exprés

1. ¿Cuál es la ventaja principal de una comprensión frente a un bucle con `append`?

- ☐ Es siempre más rápida
- ☐ Expresa el resultado deseado de forma concisa e idiomática
- ☐ Puede contener lógica arbitrariamente compleja
- ☐ No necesita iterable

Cuándo duele

Una comprensión anidada con dos filtros y una expresión larga es difícil de leer: si necesitas más de una o dos líneas lógicas, mejor un bucle normal. La legibilidad manda: la concisión no es un trofeo.

Lambda o `def`

- Lambda: una expresión corta, sin ramas ni sentencias, reutilizada en el mismo lugar.
- `def`: cuando hay varias líneas, condiciones o el comportamiento merece nombre propio.

map y filter

`map` aplica una función a cada elemento; `filter` conserva los que cumplen un predicado. En Python, muchas veces una comprensión es más clara que combinar `map` y `filter`. Úsalos cuando encajen con una función ya existente.

2. Una lambda puede contener varias sentencias y devoluciones con return.

- ☐ Verdadero
- ☐ Falso

Errores frecuentes

- Comprender con efectos secundarios (imprimir, mutar): la comprensión es para producir datos.
- Anidar comprensiones hasta volverlas indescifrables.
- Nombrar variables crípticas que ocultan qué genera la expresión.

Expresivo pero legible

Python premia leer como se escribe. Una comprensión bien nombrada es casi prosa: cuadrados de los pares mayores que dos. Si se puede leer así en voz alta, vas por buen camino.

3. Decide qué construcción corresponde:

Elementos:

- Construir lista filtrada en una línea
- Clave de ordenación de una línea
- Función con varias sentencias y condiciones
- Generar diccionario de longitudes en línea

Comprehension	. . .
Lambda	. . .
def normal	. . .

Python Docs <https://docs.python.org/es/3/tutorial/datastructures.html> Sección sobre comprensiones de lista y funciones.

VIDEO

Video complementario

Recurso audiovisual para reforzar los conceptos.



Escanea para verlo en Lumbre

<https://www.youtube.com/watch?v=rfscVS0vtbw>

4. Toma un bucle con append que filtra y transforma. Reescríbelo como comprensión y evalúa: ¿gana en claridad? Si la lógica creció demasiado, ¿qué hubieras preferido?

Documentación pandas <https://pandas.pydata.org/docs/Guia de referencia para manipulacion de datos en Python>.

Referencia curricular

Grado:	2
Materia:	Programación con Python para Datos · Programación

Versión interactiva completa (con actividades, videos y autoevaluación):



<https://univ.lumbre.lat/object/cdia-pyth-002-comprensiones-y-funciones-lambda>

Material educativo generado desde Lumbre · lumbre.lat