

L U M B R E



Listas y tuplas

Grado: 1

Materia: Fundamentos de Programación · Programación

11/10/2026

Antes de empezar

Casi todo dato real llega en fila: una columna de medidas, una lista de clientes. Las listas y las tuplas son los contenedores ordenados con los que se guarda, recorta y recorre esa información, y la diferencia entre ambas decide cuándo una secuencia puede cambiarse.

QUÉ APRENDERÁS

- Manipular listas y tuplas en Python, distinguiendo su mutabilidad, y aplicar indexación, porción e iteración para procesar secuencias de datos.

Listas y tuplas

Ordenados, pero no iguales

Una lista es una pizarra donde puedes borrar y reescribir; una tupla es una placa grabada. Ambas guardan elementos en orden, pero solo una admite cambios. Elegir la correcta comunica intención y evita errores.

Las secuencias almacenan varios valores en una sola variable, accesibles por su posición. Python ofrece dos protagonistas: la lista, que es mutable, y la tupla, que es inmutable. La indexación y el recorrido funcionan casi igual en las dos.

VIDEO

Listas y tuplas en Python para principiantes

Cómo crear, indexar y recorrer secuencias, y en qué se diferencia una lista de una tupla.



Escanea para verlo en Lumbre

<https://www.youtube.com/watch?v=KWKWswDfAb0>

Indexar: acceder por posición

- La primera posición es la cero, no la uno.
- Los índices negativos cuentan desde el final: menos uno es el último.

- Salirse del rango produce un error, no un valor vacío.

Porciones para recortar

La notación de porción toma un inicio (incluido) y un final (excluido): lista desde uno hasta tres devuelve los elementos en las posiciones uno y dos. Omitir el inicio o el final equivale a tomar hasta los extremos, y un tercer valor fija el paso.

Listas mutables

- Se pueden cambiar, añadir o quitar elementos en sitio.
- `append` agrega al final; `insert` lo coloca en una posición.
- `sort` ordena modificando la propia lista.

Tuplas inmutables

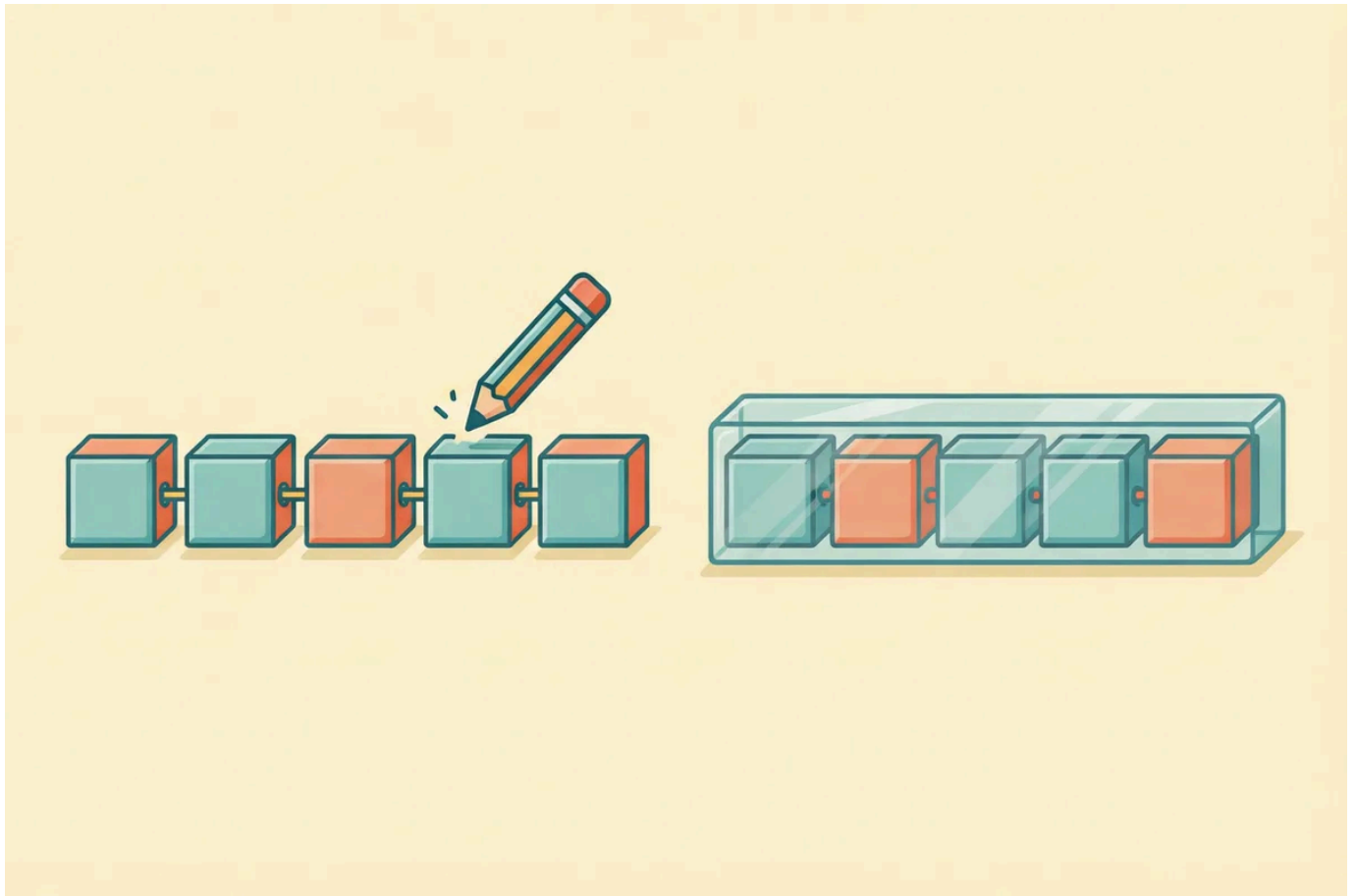
Una vez creada, una tupla no cambia. Eso la hace ligera y segura para datos que no deben alterarse, y permite usarla como clave de diccionario, cosa que una lista no puede. Un solo elemento se escribe con la coma final para distinguirlo de un paréntesis de agrupación.

Desempaquetar secuencias

Asignar varios nombres a la vez reparte los elementos de una secuencia: `x, y` igual a una tupla de dos valores. Con un asterisco se captura el resto, lo que hace el desempaquetado muy cómodo para procesar registros.

Recorrer con `for`

- Iterar directamente sobre los elementos es más claro que hacerlo por índice.
- `enumerate` aporta posición y valor a la vez.
- `zip` recorre varias listas en paralelo, emparejando por posición.



La lista se puede borrar y reescribir; la tupla queda grabada e inmutable una vez creada.

Copiar sin enredar

Asignar una lista a otro nombre no la copia: ambas apuntan al mismo objeto. Para una copia real hay que crearla expresamente; confunde a casi todo el mundo al principio y provoca cambios fantasma.

Cuándo usar cada una

- Lista para colecciones que crecen o cambian, como acumular resultados.
- Tupla para registros fijos de campos conocidos, como las coordenadas de un punto.
- La inmutabilidad de la tupla la hace apta para claves y para proteger datos.

Medir y comprobar

`len` devuelve cuántos elementos hay, e `in` pregunta si un valor pertenece a la secuencia. Son las dos consultas que más se usan al validar antes de operar.

Errores frecuentes

- Suponer que el índice inicial es uno y desalinearse todo el recorrido.

- Intentar modificar una tupla y recibir un error de objeto inmutable.
- Comparar por identidad dos listas iguales esperando que coincidan por valor.

Ejemplo resuelto: recortar y desempaquetar medidas

Tienes una lista con las temperaturas de una semana y quieres quedarte con los tres primeros días y separar el resto. Se razona con porciones y desempaquetado.

1. Indexar desde cero: el primer día ocupa la posición cero, no la uno.
2. Tomar una porción con inicio incluido y final excluido para aislar los tres primeros valores.
3. Usar un asterisco al desempaquetar para capturar en una variable los elementos restantes.
4. Recordar que una porción devuelve una lista nueva, sin alterar la original.
5. Convertir a tupla la muestra si no debe volver a modificarse.

¿Para qué sirve en la realidad?

Casi toda columna de datos llega como secuencia: saber recortar, desempaquetar y recorrer con for o zip es lo que permite limpiar y transformar registros en cualquier proyecto de análisis.

1. La principal diferencia entre una lista y una tupla es que:

- ☐ la lista guarda números y la tupla texto
- ☐ la lista es mutable y la tupla inmutable
- ☐ la tupla está ordenada y la lista no
- ☐ solo la lista se puede recorrer

2. En Python el primer elemento de una secuencia está en la posición uno.

- ☐ Verdadero
- ☐ Falso

ACTIVIDAD INTERACTIVA DISPONIBLE EN LUMBRE

3. Secuencias



4. Clasifica cada operación según qué tipo de secuencia la admite naturalmente.

Elementos:

- añadir un elemento al final
- usar la secuencia como clave de un diccionario
- recorrer sus elementos con for
- cambiar un elemento en una posición
- proteger un registro de campos fijos
- consultar cuántos elementos tiene

Lista	. . .
Tupla	. . .
Ambas	. . .

Objeto inmutable https://en.wikipedia.org/wiki/Immutable_object Objeto cuyo estado no cambia tras su creación.

5. Une cada necesidad con la estructura adecuada:

5.	guardar valores que irán cambiando	-----
5.	proteger un registro de campos fijos	-----
5.	usar una secuencia como clave de diccionario	-----
5.	añadir elementos al final	-----

Opciones: lista · tupla · tupla · lista

Listas y tuplas

Secuencias

Lista

mutable

append e insert

crece y cambia

Tupla

inmutable

sirve como clave

registro fijo

Común

índice desde cero

porciones
recorrer con for

El mapa mental interactivo es interactivo en la versión online.



Estructura de datos lista https://en.wikipedia.org/wiki/List_data_structure Colección ordenada de elementos accesibles por índice.

6. Vas a almacenar los tres lados de un triángulo y después calcular su perímetro. ¿Elegirías lista o tupla? Justifica tu elección apelando a la mutabilidad y a la intención del código.

Referencia curricular

Grado:	1
Materia:	Fundamentos de Programación · Programación

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



<https://univ.lumbre.lat/object/cdia-prog1-006-listas-y-tuplas>

Material educativo generado desde Lumbre · lumbre.lat