



Estud-IA

Introducción a la Lógica de Programación y Algoritmos con Python

Miguel Angel Carrillo - 2025

@miguelcmo

#ciudadanosdigitales

Apoya:


cymetria
El futuro es digital

Organiza:



Alcaldía de Medellín
Distrito de
Ciencia, Tecnología e Innovación

SESIÓN 3 - VIRTUAL (4 horas): Lists, dictionaries and other data structures

Objetivos:

- Estructuras de datos
- Lists
- Dictionaries
- Tuples
- Sets
- Retos de desarrollo



Data Structures

En programación, una **estructura de datos** es una forma organizada de almacenar y gestionar información para que pueda ser utilizada de manera eficiente. Cada estructura de datos tiene características, ventajas y usos específicos según el tipo de problema que se quiera resolver.

¿Qué son las estructuras de datos?

Son **modelos o formatos organizados** que permiten:

- Guardar datos
- Acceder a ellos
- Modificarlos
- Procesarlos eficientemente

Ejemplos comunes:

Listas, Tuplas, Diccionarios, Sets, Pilas (Stacks), Colas (Queues), Árboles, Grafos

En Python, los más usados inicialmente son **listas, tuplas, diccionarios y sets**.

Lists

En Python, una **lista** es una estructura de datos que permite almacenar múltiples valores en un solo lugar. Son uno de los tipos más usados en el lenguaje por su **flexibilidad** y **facilidad** de uso.

¿Qué es una lista?

Una lista es:

- **Ordenada** → los elementos mantienen un orden fijo.
- **Mutable** → puedes agregar, eliminar o modificar elementos.
- **Heterogénea** → puede contener distintos tipos de datos (números, cadenas, otras listas).
- **Indexada** → se accede a los elementos por índice (empezando en 0).

Sintaxis:

```
mi_lista = [10, "hola", 3.14, True]
```

o o o o
o o o o

Dictionaries

En Python, un **diccionario** es una estructura de datos que almacena información en forma de **pares clave-valor**. Es uno de los tipos más importantes del lenguaje porque permite organizar datos de manera clara, flexible y accesible.

✓ ¿Qué es un diccionario?

Un diccionario:

- **Es mutable** → puedes agregar, modificar o eliminar datos.
- **No está ordenado por índices**, sino por **claves**.
- **Las claves deben ser únicas**.
- **Cada clave apunta a un valor** (que puede ser cualquier tipo de dato).

Sintaxis:

```
persona = {  
    "nombre": "Miguel",  
    "edad": 30,  
    "profesion": "Ingeniero"  
}
```



Tuples

En Python, una **tupla** es una **colección ordenada e inmutable** de elementos. Se parecen a las listas, pero **no se pueden modificar** después de crearlas.

¿Qué es una tupla?

Es una estructura de datos que:

- **Mantiene el orden de los elementos**
- **Permite elementos duplicados**
- **No permite agregar, eliminar o cambiar elementos**
- Usa paréntesis: ()

Ejemplo:

```
mi_tupla = (10, 20, 30)
```

o o o o

o o o o



Sets

En Python, un **set** (conjunto) es una **colección desordenada, mutable y sin elementos duplicados**. Está inspirado en los conjuntos de la matemática.

¿Qué es un set?

- **No tiene orden** → no puedes acceder por índice.
- **No permite duplicados** → cualquier repetido se elimina automáticamente.
- **Es mutable** → puedes añadir o eliminar elementos.
- Se define con llaves: { } o usando set () .

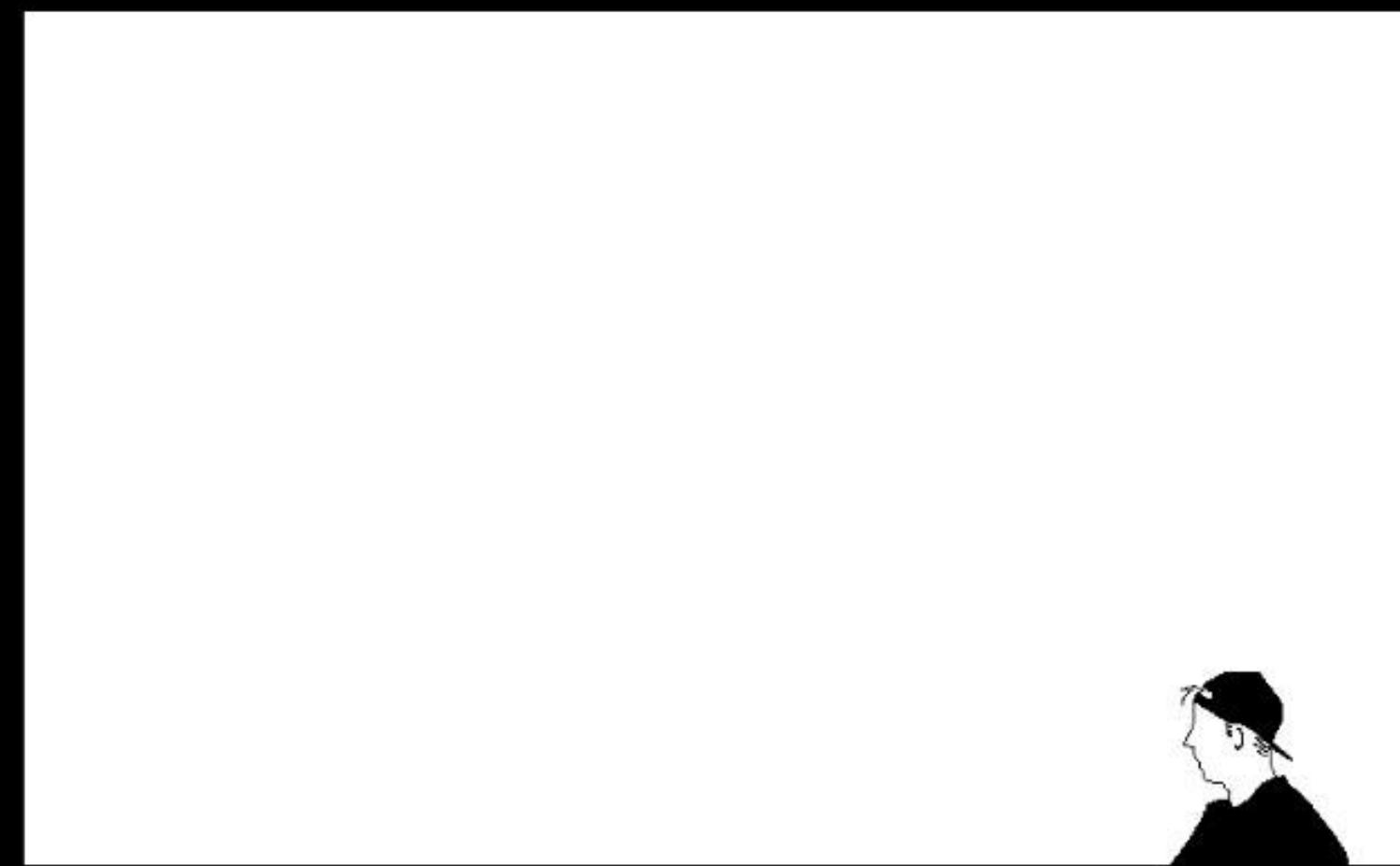
Ejemplo:

```
frutas = {"manzana", "pera", "uva"}
```

o o o o

o o o o

Break!



#ciudadanosdigitales

o o o o

👉 Reto de desarrollo 1 - Gestor de Contactos (listas, diccionarios, funciones)

Descripción:

Crear un programa que permita registrar contactos con nombre y número de teléfono.

El usuario debe poder:

- Agregar un contacto
- Mostrar todos los contactos
- Buscar un contacto por nombre
- Salir

Debe usarse:

- Una **lista** de diccionarios para almacenar los contactos
- Ciclos `while`
- Condicionales `if`
- Funciones para cada operación

Reto de desarrollo 2 - Análisis de Notas de Estudiantes

Descripción:

Pedir al usuario ingresar varias notas entre 0 y 5.

Guardar las notas en una **lista**.

Luego:

- Mostrar el promedio
- Mostrar la nota más alta y la más baja
- Devolver los resultados como una **tupla**: (promedio, mayor, menor)

El programa debe validar que cada nota ingresada sea correcta.



Reto de desarrollo 3 - Sistema de Inventario con Conjuntos (sets)

Descripción:

Crear un pequeño sistema para administrar un inventario de productos usando un **set**.

El programa debe permitir:

- Agregar un producto
- Eliminar un producto
- Verificar si un producto existe
- Mostrar el inventario completo

Debe usar:

- Set
- If
- Ciclo while
- Funciones para las operaciones

💪 Reto de desarrollo 4

Contador de Vocales en una Cadena (funciones, ciclos, sets)

Descripción:

Pedir al usuario una cadena y contar cuántas vocales tiene.

Debe:

- Convertir la cadena a minúsculas
- Recorrerla con un `for`
- Usar un **set** para las vocales (`{ 'a', 'e', 'i', 'o', 'u' }`)
- Devolver el total de vocales como un número

Ideal para practicar sets + loops + funciones.

⚠ Reto de desarrollo 5 - Tienda Virtual (listas, diccionarios, if, for, while)

Descripción:

Simular un pequeño carrito de compras.

Usar una lista de diccionarios para representar los productos disponibles:

```
productos = [{"nombre": "Pan", "precio": 1500}, {"nombre": "Leche", "precio": 3200}, {"nombre": "Huevos", "precio": 5000}]
```

Permitir al usuario:

- **Ver productos**
- **Agregar un producto al carrito**
- **Ver el carrito**
- **Calcular el total a pagar**

Se debe usar:

Listas, diccionarios, funciones, bucle while true para el menú



#ciudadanosdigitales

Estud-IA
GRACIAS

Apoya:


cymetria
El futuro es digital

Organiza:



Alcaldía de Medellín
Distrito de
Ciencia, Tecnología e Innovación