



# Python: Colecciones de Datos

Centro de Servicios y Gestión Empresarial  
SENA Regional Antioquia

# Conceptualización

# Colecciones de Datos

En Python, las colecciones de datos son estructuras que permiten almacenar múltiples valores en una sola variable. Python ofrece cuatro tipos principales de colecciones:

Listas (list)

↓  
Colección ordenada y modificable.

Tuplas (tuple)

↓  
Colección ordenada e inmutable.

Conjuntos (set)

↓  
Colección no ordenada y sin duplicados.

Diccionarios (dict)

↓  
Colección de pares clave: valor.

# Colecciones de Datos

En Python, el estilo recomendado para nombrar las colecciones es la convención **snake\_case** (minúsculas con guion bajo) según la **Guía de Estilo PEP 8**. Se recomienda plural cuando la colección contiene varios elementos.

## 💡 Reglas Generales

- ◊ **Listas, tuplas y conjuntos** → Plural (usuarios, productos, nombres).
- ◊ **Diccionarios** → Singular (porque representa un solo objeto con varios atributos) (usuario, producto).

# Tuplas

# Tuplas

Las tuplas son ordenadas y no pueden modificarse (inmutables).

## 💡 Sintaxis

```
# Definir una tupla  
  
colores = ("rojo", "verde", "azul")
```

# Tuplas

## 🔨 Métodos Principales

Método	Descripción	Ejemplo
<code>count(x)</code>	Cuenta las veces que x aparece en la tupla	<code>colores.count("rojo")</code>
<code>index(x)</code>	Devuelve la posición de x	<code>colores.index("verde")</code>

# Tuplas

## 🔨 Ejemplo

```
# Ejemplo de una Tupla de Colores
colores = ("rojo", "verde", "azul", "amarillo", "rojo")

# Posiciones de una Tupla
print(colores[1]) # 'verde'

# Contar elementos de una Tupla
print(colores.count("rojo")) # 1
```

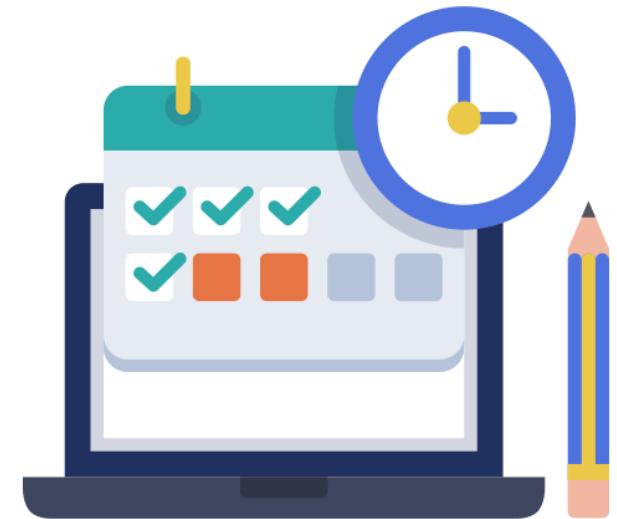
# Otros métodos

Método	Descripción	Ejemplo	Salida
<code>len(tupla)</code>	Devuelve la cantidad de elementos en la tupla.	<code>len((1, 2, 3, 4))</code>	4
<code>max(tupla)</code>	Devuelve el valor más grande de la tupla.	<code>max((4, 7, 2, 9))</code>	9
<code>min(tupla)</code>	Devuelve el valor más pequeño de la tupla.	<code>min((4, 7, 2, 9))</code>	2
<code>sum(tupla)</code>	Devuelve la suma de todos los elementos de la tupla.	<code>sum((1, 2, 3, 4))</code>	10
<code>sorted(tupla)</code>	Devuelve una lista ordenada con los elementos de la tupla.	<code>sorted((4, 1, 3, 2))</code>	[1, 2, 3, 4]
<code>tuple(lista)</code>	Convierte una lista en una tupla.	<code>tuple([10, 20, 30])</code>	(10, 20, 30)
<code>zip(tupla1, tupla2)</code>	Combina dos tuplas en pares (tuplas dentro de una lista).	<code>list(zip((1, 2, 3), ('a', 'b', 'c')))</code>	[(1, 'a'), (2, 'b'), (3, 'c')]

# Ejemplo

## 💡 Ejemplo:

- Un colegio tiene un horario semanal donde los días de clases nunca cambian.
- Los profesores necesitan un programa que verifique si un día ingresado es un día de clases o no.
- Dado que los días de la semana son fijos e inmutables, se usa una tupla.



# Ejemplo

```
# Tupla con los días de clases (inmutable)
dias_clase = ("Lunes", "Martes", "Miércoles", "Jueves",
"Viernes")

# Solicitar al usuario un día
dia = input("Ingrese un día de la semana: ").capitalize()

# Verificar si es un día de clase
if dia in dias_clase:
    print(f"{dia} es un día de clases.")
else:
    print(f"{dia} NO es un día de clases.")
```



# G R A C I A S

Presentó: Alvaro Pérez Niño

Instructor Técnico

Correo: aperezn@sena.edu.co

<http://centrodesserviciosygestionempresarial.blogspot.com/>

Línea de atención al ciudadano: 01 8000 910270

Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)