SESIÓN 10 · MÓDULO 1

Tablas, tuplas y conjuntos

Representar una tabla con una lista de listas y elegir entre lista, tupla y conjunto según lo que haga falta.

Antonio Chanes Espigares · CC BY-NC-ND 4.0

Hoy

0:00	Corrección de la tarea 9.
0:15	Situación: el plano de una sala de cine; una entrada dice fila y asiento.
0:25	**Predice**. Se compara `[[0] * 3 for _ in range(2)]` con `[[0] * 3] * 2`: el visualizador enseña por qué la…
0:45	Construcción: tablas, tuplas y desempaquetado, conjuntos y sus operaciones.
1:20	Ejercicios de los dos capítulos.
1:45	Preguntas tipo examen y tarea 10.

Para empezar

El plano de una sala de cine; una entrada dice fila y asiento.

Predice

Escribe en papel qué crees que pasará. Después, lo ejecutamos.

```
t = (1, 2) + (3,)
s = {3, 1, 3, 2}
print(t, len(s), 2 in s)
m = [[0] * 3 for _ in range(2)]
m[1][2] = 5
print(m)
```

Predice

Escribe en papel qué crees que pasará. Después, lo ejecutamos.

```
t = (1, 2) + (3,)
s = {3, 1, 3, 2}
print(t, len(s), 2 in s)
m = [[0] * 3 for _ in range(2)]
m[1][2] = 5
print(m)
```

LO QUE SALE

`(1, 2, 3) 3 True` · `[[0, 0, 0], [0, 0, 5]]`

`(3,)` es una tupla de un elemento; el conjunto no repite.

Tablas: listas de listas

Listas a fondo · 5

```
notas = [[7, 5, 9], [6, 8, 4]]
medias = []
for i in range(len(notas)):
    suma = 0
    for j in range(len(notas[i])):
        suma = suma + notas[i][j]
    medias.append(suma / len(notas[i]))
print(medias)
```

Tuplas: crearlas y leerlas

Tuplas y conjuntos · 1

```
vacia = ()
uno = (5,)                # la coma es la que hace la tupla
no_es_tupla = (5)         # esto es solo el número 5 entre paréntesis
fecha = (18, 11, 2026)
punto = 3, 7              # los paréntesis se pueden omitir
print(type(uno).__name__, type(no_es_tupla).__name__, len(fecha))
print(fecha[0], fecha[-1], fecha[1:])
print(fecha.count(18), fecha.index(2026), 2026 in fecha)
```

Inmutables

Tuplas y conjuntos · 2

```
alumno = ("Ana", [7, 9])
alumno[1].append(8)       # la lista de dentro sí cambia
print(alumno, alumno[1][0])
```

Empaquetar y desempaquetar

Tuplas y conjuntos · 3

```
a, b = 1, 2
a, b = b, a
print(a, b)
for nombre, nota in [("Ana", 7), ("Biel", 9)]:
    print(nombre, "ha sacado un", nota)
```

Conjuntos: sin repetidos y sin orden

Tuplas y conjuntos · 4

```
mates = {"Ana", "Biel", "Carla"}
fisica = {"Biel", "Dani"}
print(sorted(mates | fisica), sorted(mates & fisica), sorted(mates - fisica))
print(len(set([3, 1, 3, 2, 1])))
try:
    mates[0]
except TypeError as e:
    print("Error:", e)
```

Manos a la obra

Abre el cuaderno de la sesión en Colab, con los ejemplos de hoy y estos ejercicios:

1. Índices negativos
2. ¿Qué imprime?
3. Quitar todos. Escribe un programa que quite todos los 0 de datos =…
4. El total de la tabla
5. La diagonal. Para la tabla m = [[1, 2, 3], [4, 5, 6], [7, 8, 9]],…
6. ¿Qué tipo es? a) (3); b) (3,); c) 3, 4; d) {}; e) set().



Cuaderno en Colab

Tarea 10 · Asientos de cine

1. Crea una sala de 5 filas y 8 asientos con una lista de listas: `0` es libre y `1` ocupado.
2. Ocupa la primera fila entera y los dos primeros asientos de la segunda.
3. Escribe una función `primera_libre(sala)` que devuelva la tupla `(fila, asiento)` del primer asiento libre, recorriendo…
4. Escribe una función `libres(sala)` que cuente los asientos libres.
5. Reserva el asiento que te dé `primera_libre`, desempaquetando la tupla, y dibuja la sala con `X` y `·`.

COMPROBACIÓN Antes de reservar, el primer libre es `(1, 2)` y hay 30 libres.

Antes de irte

Las **preguntas tipo examen** del capítulo, en directo: se corrigen al momento.



Preguntas del capítulo

LA PRÓXIMA SESIÓN

Sesión 11 · Cadenas y formato

Programación en Python · Sesión 10 · Tablas, tuplas y conjuntos