



Practica 08 - Tuplas y Conjuntos



Instrucciones

Entregar los ejercicios en uno o varios `archivos.py` o en un notebook de Jupyter (`.ipynb`)

Parte 1: Tuplas

Ejercicio 1

Crea una tupla con números e indica el número con mayor valor y el que menor tenga.

Ejercicio 2

Escribir un código donde se le pide al usuario almacenar en una tupla el día, mes y año (todos en formato entero) y luego que imprima dicha fecha en el formato: "03/07/2023".

Ejercicio 3

Definir una tupla con tres valores enteros. Convertir el contenido de la tupla a tipo lista. Modificar la lista y luego convertir la lista en tupla.

Ejercicio 4

Crear una tupla y luego imprimirla en orden inverso en otra variable (otra tupla).

Ejercicio 5

Dada la tupla:

```
tuple1 = ("Orange", [10, 20, 30], (5, 15, 25))
```

Acceder al valor '20' e imprimir en una nueva variable.

Ejercicio 6

Escriba un programa para descomprimir la siguiente tupla en cuatro variables e imprimir cada variable:

```
tuple1 = (10, 20, 30, 40)
```

Salida esperada:

```
tuple1 = (10, 20, 30, 40)
# Your code
print(a) # should print 10
print(b) # should print 20
print(c) # should print 30
print(d) # should print 40
```

Ejercicio 7

¿Cuántas veces aparece 777 en la siguiente tupla? Escribir código

```
tt = (55, 6, 777, 54, 6, 76, 7777, 1, 777, 2, 6)
```

Ejercicio 8

¿Cuáles son los valores mínimo y máximo de la siguiente tupla? Escribir código

```
tt = (42, 1092, 11, 88, 65, 2, 6)
```

Ejercicio 9

Considere la siguiente tupla y escriba el código para las siguientes declaraciones:

```
T1 = (12, 3, 45, 'Hockey', 'Anil', ('a', 'b'))
```

- a) Imprimir el primer elemento de T1
 - b) Imprimir el último elemento de T1
 - c) Imprimir T1 en orden inverso
 - d) Imprimir 'Anil' desde la tupla T1
 - e) Imprimir 'b' de la tupla T1
-

Ejercicio 10

Escriba un programa que le pida a un usuario una cadena (string), luego convertirlo a una tupla e imprimir.

Ejercicio 11

Dada la siguiente tupla:

```
('F', 'l', 'a', 'b', 'b', 'e', 'r', 'g', 'a', 's', 't', 'e',  
'd')
```

Escribe un programa en Python que realice las siguientes operaciones:

- Añadir un carácter  al final de la tupla
- Convertir la tupla a una cadena

- Extraer la sub-tupla `('b', 'b')` de la tupla
- Contar cuántas veces aparece la letra `'e'` en la tupla
- Comprobar si el carácter `'r'` existe en la tupla
- Convertir la tupla en una lista
- Eliminar los caracteres `'b'`, `'b'`, `'e'`, `'r'` de la tupla

Ejercicio 12

Relaciona los siguientes pares:

- | | |
|--|---|
| a. <code>tpl1 = ('A',)</code> | 1. tupla de longitud 6 |
| b. <code>tpl1 = ('A')</code> | 2. tupla de listas |
| c. <code>t = tpl[::-1]</code> | 3. Tupla |
| d. <code>('A', 'B', 'C', 'D')</code> | 4. lista de tuplas |
| e. <code>[(1, 2), (2, 3), (4, 5)]</code> | 5. Cadena de texto |
| f. <code>tpl = tuple(range(2, 5))</code> | 6. Ordena tupla |
| g. <code>[[1, 2], [3, 4], [5, 6]]</code> | 7. <code>(2, 3, 4)</code> |
| h. <code>t = tuple('Ajooba')</code> | 8. tupla de caracteres |
| i. <code>[*a, *b, *c]</code> | 9. Desempaquetado de tuplas en una lista |
| j. <code>(*a, *b, *c)</code> | 10. Desempaquetado de listas en una tupla |

Parte 2: Set()

Ejercicio 1

Escribe un programa para realizar las siguientes operaciones sobre el conjunto dado:

```
s = {10, 2, -3, 4, 5, 88}
```

- Obtener el número de elementos en el conjunto `s`.

- Obtener el valor máximo en el conjunto `s`.
 - Obtener el valor mínimo en el conjunto `s`.
 - Calcular la suma de todos los elementos en el conjunto `s`.
 - Obtener un nuevo conjunto ordenado a partir de `s`, dejando el conjunto original sin cambios.
 - Indicar si el número `100` es un elemento del conjunto `s`.
 - Indicar si el número `3` es un elemento del conjunto `s`.
-

Ejercicio 2

Crea un conjunto vacío.

Escribe un programa que:

- Añada cinco nombres nuevos al conjunto,
 - Modifique uno de los nombres existentes,
 - Elimine dos nombres existentes del conjunto.
-

Ejercicio 3: Intersección de conjuntos de estudiantes

Tienes dos listas:

- Una con nombres de estudiantes que asisten a clases de matemáticas
- Otra con nombres de estudiantes que asisten a clases de física

Escribe un programa que use conjuntos para encontrar e imprimir los nombres de los estudiantes que asisten a **ambas clases**.

Ejercicio 4: Detección de duplicados en una lista

Dada una lista de correos electrónicos con posibles duplicados, escribe un programa que:

1. Elimine los correos duplicados usando `set()`
2. Imprima la cantidad total de correos originales y únicos
3. Muestre la lista final sin duplicados

Ejercicio 5: Conjuntos por comprensión

Usa un **set comprehension** para construir un conjunto que contenga todos los múltiplos de 4 entre 1 y 100 que **no sean múltiplos de 6**.

Ejercicio 6: Unpacking de conjuntos

Tienes un conjunto de números enteros sin un orden específico. Escribe un programa que:

1. Use **unpacking** para extraer los valores del conjunto en variables individuales (tantas como sea posible)
 2. Imprima los valores desempaquetados
 3. Imprima el conjunto original y la cantidad de elementos
-

Ejercicio 7: Gestión de Clientes en una Empresa de e-Commerce

Una empresa de e-commerce gestiona tres listas:

- Clientes que realizaron compras en el **Black Friday**
- Clientes que realizaron compras en **Navidad**
- Una lista de **correos bloqueados** por comportamiento fraudulento

Desarrolla un programa en Python que:

1. Convierta las tres listas a conjuntos
2. Determine e imprima:
 - Clientes que compraron en **ambas campañas**
 - Clientes **exclusivos** de Black Friday
 - Todos los clientes que **compraron en al menos una campaña**
3. Elimine del conjunto final de clientes todos los **correos bloqueados**
4. Muestra:
 - El total de clientes únicos válidos

- Los 3 primeros correos desempaquetados
 - El conjunto final ordenado alfabéticamente (sin modificar el original)
5. ¿Cuántos clientes **compraron solo en Navidad?**
 6. ¿Qué porcentaje de los clientes totales fueron **bloqueados?**
 7. Desempaquetá los **5 primeros correos alfabéticamente** en variables individuales.
 8. Elimina del conjunto final todos los correos que **comiencen con "m"** (sin importar el dominio).
 9. Verifica si `"lucia.lopez@yahoo.com"` y `"raul.garcia@gmail.com"` aún están en el conjunto final.
 10. Muestra el conjunto final convertido a `frozenset` y comprueba que es inmutable intentando modificarlo.

i Listas

```
black_friday = [  
    "ana23@gmail.com", "luis89@hotmail.com", "carla.z@outloo  
k.com", "marcos_rp@gmail.com",  
    "lucia.lopez@yahoo.com", "raul.garcia@gmail.com", "ines32  
@protonmail.com", "daniel77@gmail.com",  
    "recio.rivera@gmail.com", "pedro_42@outlook.com", "mateo@  
hotmail.com", "sofia.gm@gmail.com",  
    "eva88@yahoo.com", "juancruz@gmail.com", "tomas23@outloo  
k.com", "clara.huerta@gmail.com",  
    "jose_lg@gmail.com", "camila.rey@gmail.com", "alejandro99  
@live.com", "maria.sol@hotmail.com"  
]  
  
navidad = [  
    "carla.z@outlook.com", "daniel77@gmail.com", "recio.river  
a@gmail.com", "sofia.gm@gmail.com",  
    "jose_lg@gmail.com", "marina.rr@yahoo.com", "leo.lopez@gm
```

```
ail.com", "valentina@outlook.com",
    "ricardo88@gmail.com", "martina.gc@gmail.com", "carlos.ro
jas@hotmail.com", "ana23@gmail.com",
    "camila.rey@gmail.com", "martin.ss@gmail.com", "natalia.v
g@live.com", "ignacio.hp@gmail.com",
    "lucas_mp@gmail.com", "victoria@yahoo.com", "maria.sol@ho
tmail.com", "lucia.lopez@yahoo.com"
]
```

```
bloqueados = [
    "pedro_42@outlook.com", "mateo@hotmail.com", "alejandro99
@live.com",
    "raul.garcia@gmail.com", "leo.lopez@gmail.com", "camila.r
ey@gmail.com"
]
```