

EXERCISES QUE TRABAJAREMOS EN EL CUE

- EXERCISE 1: ACCEDIENDO A DATOS DE UNA LISTA.
- EXERCISE 2: ACCEDIENDO A DATOS DE UNA TUPLA.
- EXERCISE 3: ACCEDIENDO A ELEMENTOS DE UN SET.
- EXERCISE 4: ACCEDIENDO A LOS DATOS DE UN DICCIONARIO.
- EXERCISE 5: DATOS EN DISTINTAS ESTRUCTURAS DE DATOS.
- EXERCISE 6: CONOCIENDO LA CANTIDAD DE DATOS DE UN DICCIONARIO.

EXERCISE 1: ACCEDIENDO A DATOS DE UNA LISTA

En este ejercicio estudiaremos cómo acceder a un dato en una lista.

Dada una lista con los números 1, 2, 3, 4 y 5, imprimiremos el valor del dato con índice 3.

Primero creemos la lista según las especificaciones:

```
1|mi_lista = [1, 2, 3, 4, 5]
```

Ahora imprimamos el valor del índice 3, accediendo a él usando el operador corchetes.

```
1|print(mi_lista[3])
```

Veamos el código completo, y el resultado:

```
1|mi_lista = [1, 2, 3, 4, 5]
2|print(mi_lista[3])
3|
4|#Resultado:
5|4
```

Recordemos que el mismo procedimiento aplica para las tuplas.

EXERCISE 2: ACCEDIENDO A DATOS DE UNA TUPLA

En este ejercicio estudiaremos cómo acceder a un dato, y a un grupo de datos en una tupla.

Dada una tupla que llamaremos **tup1** con los siguientes datos (1, 2, 3, 4, 5, 6, 7), imprimir los valores desde el índice 1 hasta el 5 de **tup1**.

Primero creamos la tupla según el enunciado:

```
1| tup1 = (1, 2, 3, 4, 5, 6, 7)
```

Luego, haciendo uso del operador corchetes, accederemos a los datos y los imprimimos. Debemos tomar en cuenta que para el caso de **tup2** indicaremos un rango de claves.

```
1| print ("tup1[1:5]: ", tup1[1:5])
```

Veamos el código completo, y el resultado:

```
1| tup1 = (1, 2, 3, 4, 5, 6, 7)
2| print ("tup1[1:5]: ", tup1[1:5])
3|
4| #Resultado:
5| tup1[0]: (2, 3, 4, 5)
```

Notemos que el resultado, cuando indicamos un rango, es otra tupla solo con los datos del rango indicado.

También es importante recordar que el mismo procedimiento aplica para las listas.

EXERCISE 3: ACCEDIENDO A ELEMENTOS DE UN SET

Este ejercicio tiene por finalidad mostrar cómo acceder a los elementos de un set.

Imprimiremos todos los valores del siguiente set:

```
1| set1 = set(["Esto", "es", "un", "Ejercicio"])
```

Luego, verificaremos si el valor "prueba" se encuentra en el set.

Primero declaremos el set según las indicaciones:

```
1| set1 = set(["Esto", "es", "un", "Ejercicio"])
```

Ahora haciendo uso de la sentencia **for**, y de la palabra **in**, accederemos a todos los valores del set para imprimirlos. Usemos la indicación **end=""** en el **print()** para que no salte de línea en cada iteración.

```
1| for i in set1:
```

```
2 | print(i, end=" ")
```

Ahora verifiquemos si el valor “prueba” se encuentra dentro del set. Para ello haremos uso de la palabra **in**.

```
1 | print() #Este print es para que tengamos un salto de línea
2 | print("prueba" in set1)
```

La sentencia anterior dará como resultado *True* si se encuentra, o *False* en caso contrario.

Veamos el código completo, y el resultado:

```
1 | set1 = {"Esto", "es", "un", "Ejercicio"}
2 |
3 | for i in set1:
4 |     print(i, end=" ")
5 |
6 | print()
7 | print("Ejercicio" in set1)
8 | #Resultado:
9 | Ejercicio un Esto es
10 | True
```

Como podemos notar, dado que el set no tiene orden, el resultado que obtenemos al imprimir todos los valores no necesariamente está en el mismo orden en que se escribieron en la declaración.

EXERCISE 4: ACCEDIENDO A LOS DATOS DE UN DICCIONARIO

Este ejercicio tiene como finalidad acceder a los datos de un diccionario.

Dado el siguiente diccionario con los datos de una persona:

```
1 | dict = {'Nombre': 'Luis', 'Apellido': 'Gonzalez', 'Edad': 21}
```

Imprimiremos en pantalla el nombre y la edad de la persona. Para eso, primero declaremos el diccionario según las especificaciones.

```
1 | dict = {'Nombre': 'Luis', 'Apellido': 'Gonzalez', 'Edad': 21}
```

Ahora imprimamos el nombre y la edad. Para ello utilizaremos el operador corchete para acceder al valor de cada dato según su clave, que en este caso es: “Nombre” y “Edad”.

```
1 | print("Nombre: ", dict['Nombre'])
2 | print("Edad: ", dict['Edad'])
```

Veamos el código completo, y el resultado.

```
1 dict = {'Nombre': 'Luis', 'Apellido': 'Gonzalez', 'Edad': 21}
2 print("Nombre: ", dict['Nombre'])
3 print("Edad: ", dict['Edad'])
4
5
6 #Resultado:
7 Nombre: Luis
8 Edad: 21
```

EXERCISE 5: DATOS EN DISTINTAS ESTRUCTURAS DE DATOS

El siguiente ejercicio tiene como finalidad conocer la cantidad de datos que contienen diversas estructuras de datos.

Debemos crear una lista, una tupla, y un set con los siguientes datos respectivamente:

- 10, 20, 14
- 1, 2, 4, 6
- 1000, 2000, 3000, 4000, 5000

Luego debemos imprimir en pantalla cuántos datos tiene cada una de las estructuras.

Primero creamos las estructuras según los requerimientos:

```
1 mi_lista = [10, 20, 14]
2 mi_tupla = (1, 2, 4, 6)
3 mi_set = {1000, 2000, 3000, 4000, 5000}
```

Ahora, usando la función **len()** imprimamos en pantalla la cantidad de datos de cada una:

```
1 print("Lista:", len(mi_lista))
2 print("Tupla:", len(mi_tupla))
3 print("Set:", len(mi_set))
```

Veamos el código completo, y el resultado:

```
1 mi_lista = [10, 20, 14]
2 mi_tupla = (1, 2, 4, 6)
3 mi_set = {1000, 2000, 3000, 4000, 5000}
4 print("Lista:", len(mi_lista))
5 print("Tupla:", len(mi_tupla))
6 print("Set:", len(mi_set))
```

```
7
8
9 #Resultado:
10 Lista: 3
11 Tupla: 4
12 Set: 5
```

EXERCISE 6: CONOCIENDO LA CANTIDAD DE DATOS DE UN DICCIONARIO

El siguiente ejercicio tiene como finalidad conocer la cantidad de datos que contiene un diccionario.

Debemos crear un diccionario que almacena el inventario de frutas de una alacena. En la alacena hay 100 manzanas, 120 peras y 50 naranjas. Luego, debemos imprimir en pantalla cuántos datos contiene el diccionario.

Para resolver el problema primero debemos obtener una lista con las claves de cada par usando la función `.keys()`, y luego contaremos la cantidad de elementos de esta lista con la función `len()`. Creemos el diccionario según las especificaciones:

```
nro_frutas = {"Manzanas" : 100, "Peras" : 120, "Naranjas" : 50}
```

Ahora obtengamos las claves de cada par usando la función `keys()`.

```
1 x = nro_frutas.keys()
```

Finalmente, haciendo uso de `len()`, imprimamos la cantidad de elementos de la lista de claves.

```
1 print("Cantidad de elementos: ", len(x))
```

Veamos el código completo, y el resultado:

```
1 nro_frutas = {"Manzanas" : 100, "Peras" : 120, "Naranjas" : 50}
2 x = nro_frutas.keys()
3 print("Cantidad de elementos:", len(x))
4
5
6 #Resultado:
7 Cantidad de elementos: 3
8
9 #Resultado:
10 Japon - Tokio
11 Inglaterra - Londres
```

```
12|Francia - Paris  
13|Alemania - Berlin
```