Repaso de Listas

Una lista es un conjunto ordenado de valores, en el cual cada valor va identificado por un índice. Los valores que constituyen una lista son sus elementos.
Las listas son similares a las cadenas de texto (strings), que son conjuntos ordenados de caracteres, excepto en que los elementos de una lista pueden ser de cualquier tipo.
Las listas y las cadenas, y otras cosas que se comportan como conjuntos ordenados, se llaman secuencias.

Valores de una lista

Hay varias maneras de crear una nueva lista; la más sencilla es encerrar sus elementos entre corchetes: [10, 20, 30, 40]
["spam", "elástico", "golondrina"]
El primer ejemplo es una lista de cuatro enteros.
El segundo es una lista de tres cadenas de texto.
Los elementos de una lista no tienen por que ser del mismo tipo.
La siguiente lista contiene una cadena, un número con decimales y un entero, y, maravilla de las maravillas, otra lista:
["hola", 2.0, 5, [10, 20]]
Se dice que una lista dentro de otra lista está anidada.

Acceso a los elementos

La sintaxis para acceder a los elementos de una lista es la misma que para acceder a los caracteres de una cadena: el operador corchetes [].
La expresión dentro de los corchetes especifica el índice.
Recuerde que los índices siempre comienzan en cero:
print (numeros[0])
numeros[1] = 5

Las listas son mutables

A diferencia de las cadenas, las listas son mutables, lo que significa que podemos cambiar sus elementos.
Podemos modificar uno de sus elementos usando el operador corchetes en el lado izquierdo de una asignación:

```
>>> fruta = ["plátano", "manzana", "membrillo"]
>>> fruta[0] = "pera"
>>> fruta[-1] = "naranja"
>>> print fruta
['pera', 'manzana', 'naranja']
```

Borrado en una lista

El uso de porciones para borrar elementos de una lista puede ser extraño, y por ello propicio a los errores. Python nos da una alternativa que resulta más legible.

del elimina un elemento de una lista:
>>> a = [’uno’, ’dos’, ’tres’]
>>> del a[1]
>>> a [’uno’, ’tres’]

Tuplas

Las tuplas son colecciones ordenadas e inmutables.

Se utilizan para almacenar datos que no deben modificarse.

Solo hay dos diferencias entre el tipo de datos *de tupla* y el tipo de datos de lista. La primera diferencia es que escribes tuplas usando paréntesis en lugar de corchetes. Por ejemplo:

#se definen usando paréntesis () son inmutables

nombres =("iris",1.60,10)

print(type(nombres))

print(nombres[0])

La segunda y principal manera en que las tuplas son diferentes de las listas es que las tuplas, como las cadenas, son inmutables: no se puede modificar, anexar o eliminar sus valores.

#no puedo asignar verificar nombres[0]="gaston"
nombres[0]="gaston"
print(nombres)

Puede usar tuplas para transmitir a cualquier persona que lea su código que no tiene la intención de que cambie esa secuencia de valores. Si necesita una secuencia ordenada de valores que nunca cambia, utilice una tupla.

Diccionarios

Los diccionarios almacenan datos mediante pares clave–valor.

Son muy utilizados en sistemas reales porque permiten representar información estructurada.

Como una lista, un *diccionario* es una colección mutable de muchos valores. Pero a diferencia de los índices para las listas, los índices para diccionarios pueden usar muchos tipos de datos diferentes, no solo enteros. Estos índices de diccionario se llaman *claves*, y una clave con su valor asociado se llama un *par clave-valor*.

#diccionario usan clave: valor

alumno = {"nombre":"José","edad":29,"estatura":1.78}

#mostrar el diccionarios

print(alumno)

#mostrar por clave

print(alumno["edad"],alumno["estatura"])

Los diccionarios tienen claves, no índices.

Operaciones con diccionarios

Acceso a valores

```
print(alumno["nombre"])
```

Agregar datos

```
#agregamos otra clave
alumno["deporte"]="fútbol"
print(alumno)
```

```
{'nombre': 'Josue', 'edad': 29, 'estatura': 1.78, 'deporte': 'fútbol'}
```

Eliminar

usar del, por ejemplo eliminar la clave edad.

print(alumno)

```
{'nombre': 'Josue', 'estatura': 1.78, 'deporte': 'fútbol'}
```

Recorrer diccionarios

```
for clave, valor in persona.items():
    print(clave, valor)
```

Actividades Prácticas

Actividad 1:

Crear un diccionario vacío.

Agregar tres elementos (clave:valor)

Imprimir diccionario

Actividad 2

Definir un diccionario que almacene los nombres de paises como clave y como valor la cantidad de habitantes. Usar un bucle for para mostrar cada clave y valor.

paises={"argentina":40000000, "españa":46000000, "brasil":190000000, "uruguay": 3400000}

```
for clave in paises:
    print(clave, paises[clave])
```

Actividad 3

Crear un diccionario que permita almacenar 5 artículos, utilizar como clave el nombre de productos y como valor el precio del mismo.

Imprimir el diccionario en su totalidad.

Imprimir solo los artículos con precio superior a 100.

Actividad 4 – Agenda telefónica

Crear un diccionario con nombres y teléfonos.

Actividad 4 – Sistema de productos

Guardar productos y precios en un diccionario.

Actividad 5 – Inventario simple

Permitir:

Agregar productos.
Modificar stock.
Mostrar inventario.