



INGENIERIA EN COMPUTACION



Materia: Programación Python

Alumno: Ayon Dimas Andres

Matricula: 366845

Profesor: Pedro Nunes Yepiz

Actividad: #9

Tema: Listas y Diccionesarios

Ensenada B.C 17 de octubre del 2023

INTRODUCCIÓN

En esta práctica podremos ver lo que es el uso de listas y diccionarios en Python las cuales nos permiten almacenar una colección de elementos, mientras que los diccionarios nos ayudan a manejar datos en pares clave-valor.

Aquí podremos ver ejemplos de cómo estas herramientas pueden ser aplicadas en situaciones cotidianas. Así mismo podemos ver sobre como utilizar las listas y también los diccionarios en python de manera eficiente en Python, para que puedas comprender mejor su utilidad y funcionalidad en la programación.

COMPETENCIA

Aprender a aplicar estas sentencias mencionadas para su implementación dentro del programa en Python cuando es adecuado utilizarlas. Es de suma importancia dominar estas sentencias ya que son base fundamental de Python y nos serán útiles a la hora de querer guardar información, ya que esto nos permite hacerlo de manera eficiente y generando datos aleatorios proporcionados por el usuario.

FUNDAMENTOS

LISTAS

Una lista en Python es un tipo de datos nativos construido dentro del lenguaje de programación Python. Estas listas son similares a matrices (o arrays) que se encuentran en otros lenguajes. Sin embargo, en Python se manejan como variables con muchos elementos. Aun así, las listas se consideran un tipo de datos para guardar colecciones de información.

Una lista en Python es:

1. Ordenada: esto quiere decir que los elementos dentro de ella están indexados y se accede a ellos a través de una locación indexada.
2. Editable: los elementos dentro de una lista pueden editarse, añadir nuevos o eliminar los que ya tiene.

3. Dinámica: las listas pueden contener diferentes tipos de datos y hasta de objetos. Esto significa que pueden soportar paquetes multidimensionales de datos, como un array o muchos objetos.

4. No única: esencialmente, esto quiere decir que la lista puede contener elementos duplicados sin que arroje un error.

Diccionarios en python

Los diccionarios en Python nos permiten almacenar una serie de *mapeos* entre dos conjuntos de elementos, llamados *keys* and *values* (*Claves y Valores*).

- Todos los elementos en el diccionario se encuentran encerrados en un par de corchetes.
- Cada elemento en un diccionario contiene una clave y un valor - es decir un *par de clave-valor*.
- Cada par de clave-valor es denominado como *elemento (item)*.
- La ventaja de esto es que puedes acceder a todos los valores almacenados usando simplemente las claves.

Algunas propiedades de los diccionario en Python son las siguientes:

- Son dinámicos, pueden crecer o decrecer, se pueden añadir o eliminar elementos.
- Son indexados, los elementos del diccionario son accesibles a través del `key`.

Y son anidados, un diccionario puede contener a otro diccionario en su campo `value`.

PROCEDIMIENTO

Creación de un diccionario

Acceder y modificar elementos

Se puede acceder a sus elementos con `[]` o también con la función `get()`

```
print(d1['Nombre']) #Sara
```

```
print(d1.get('Nombre')) #Sara
```

Para modificar un elemento basta con usar `[]` con el nombre del key y asignar el valor que queremos.

Python

```
d1['Nombre'] = "Laura"

print(d1)

#{'Nombre': Laura, 'Edad': 27, 'Documento': 1003882}
```

Si el key al que accedemos no existe, se añade automáticamente.

Python

```
d1['Direccion'] = "Calle 123"

print(d1)

#{'Nombre': 'Laura', 'Edad': 27, 'Documento': 1003882, 'Direccion': 'Calle123'}
```

Iterar diccionario

Los diccionarios se pueden iterar de manera muy similar a las listas u otras estructuras de datos. Para imprimir los key.

Python

```
# Imprime los key del diccionario

for x in d1:

    print(x)

#Nombre

#Edad

#Documento

#Direccion
```

Se puede imprimir también solo el value.

Python

```
# Impre los value del diccionario

for x in d1:

    print(d1[x])

#Laura

#27

#1003882

#Calle 123
```

O si queremos imprimir el key y el value a la vez.

Python

```
# Imprime los key y value del diccionario

for x, y in d1.items():

    print(x, y)
```

```
#Nombre Laura
```

```
#Edad 27
```

```
#Documento 1003882
```

```
#Direccion Calle 123
```

RESULTADOS Y CONCLUSIONES

Aquí pudimos aprender que las listas pueden contener varios tipos de objetos, mientras que los diccionarios son estructuras que permiten almacenar algún tipo de mapeo entre claves(**Keys**) y valores, siendo dinámicos y anidados. Así mismo pudimos aprender a crear, acceder y modificar elementos en un diccionario, así como a iterar sobre ellos para imprimir sus claves, valores o ambas cosas.

Por lo tanto podemos decir que es de suma importancia que comprendamos a fondo el uso de estas sentencias ya que resulta fundamental para el desarrollo de programas eficientes en Python y para manejar eficazmente la información.

El hecho de hacer este tipo de programas en Python nos proporciona un mejor manejo de la programación en Python y facilita el manejo de datos de manera eficiente y a su vez eficaz.