



INGENIERIA EN COMPUTACION



Materia: Python

Alumno: Ayon Dimas Andres

Matricula: 366845

Profesor: Pedro Nunes Yepiz

Tema: Dicionarios y Archivos de texto

INTRODUCCION

En este programa realizado en Python podremos ver lo que es un menú, en el cual se tiene distintas opciones, contando con un total de 10, en las cuales nos enfocaremos en lo que son los diccionarios, ya que en base a este generaremos cierta información la cual nos sirve para poder realizar ciertos archivos, donde dentro de las opciones, mediante inciso podremos seleccionar que tipo de archivo queremos generar y con la información que generamos con los diccionarios podremos imprimirla en el tipo de archivo de nuestra elección.

COMPETENCIA

Debemos aprender a utilizar los diccionarios con ciertas condiciones como ya vimos en programas anteriores, así mismo el uso de los archivos de texto, ya que nosotros podemos elegir qué tipo de texto queremos generar con cierta información que pueda ir dentro del programa y que mediante este mismo la información sea generada y almacenada en un archivo de texto.

FUNDAMENTOS

DICCIONARIOS

Un **diccionario en Python es una estructura de datos que permite almacenar cualquier tipo de información**, desde cadenas de texto o caracteres hasta números enteros, con decimales, listas e incluso otros diccionarios. Al igual que sucede con un diccionario de lengua, los datos se encuentran ordenados utilizando una clave única para cada uno de ellos, lo que permite localizar cada uno de los datos de una forma muy rápida.

Los **diccionarios Python son mutables**, esto quiere decir que no tienen un tamaño predefinido y que su contenido aumenta o disminuye según las necesidades de la aplicación. Todos los datos son también modificables, es decir, se puede añadir, modificar, eliminar y consultar todos los datos de una manera sencilla y rápida.

Se trata de una **estructura de datos similar a las listas, las tuplas o los conjuntos**. Se utilizan básicamente para almacenar una gran cantidad de datos de forma ordenada,

con el objetivo de utilizarlos para algún propósito en concreto, más adelante podrás ver algunos ejemplos de uso.

Ejemplos de diccionario de datos en Python

Los valores de un diccionario se guardan utilizando un **par de valores que siempre van enlazados**. Una es la denominada como **Key o Clave**, que es la que nos permite encontrar un dato dentro del diccionario. Cada clave está acompañada por el **dato o valor al que representa**.

ARCHIVOS DE TEXTO

Los archivos de texto en Python son una forma de almacenar información de texto en un archivo. Python proporciona una variedad de formas de trabajar con archivos de texto, lo que permite a los programadores leer, escribir y manipular datos en archivos de texto.

Además, Python proporciona métodos para trabajar con archivos binarios, como rb (lectura binaria), wb (escritura binaria), y ab (agregar binario), que son útiles al tratar con tipos de datos no textuales. En general, el manejo de archivos de texto en Python es una parte fundamental de la programación, ya que permite a los desarrolladores manipular datos de una manera que pueda ser fácilmente leída y modificada por humanos y por otros programas.

PROCEDIMIENTO

Diccionarios

```
productos = {  
    "producto1": {"nombre": "Laptop", "precio": 1000},  
    "producto2": {"nombre": "Teléfono", "precio": 500},  
    "producto3": {"nombre": "Tablet", "precio": 300}  
}  
  
# Acceder a valores anidados  
print(productos["producto1"]["nombre"]) # Salida: Laptop  
print(productos["producto2"]["precio"]) # Salida: 500
```

Archivos de texto

1.- Lectura de un archivo de texto

```
# Abre un archivo en modo lectura y lee su contenido
with open('archivo.txt', 'r') as archivo:
    contenido = archivo.read ()
    print(contenido)
```

2.- Escritura de un archivo de texto

```
# Abre un archivo en modo escritura y escribe en él
with open('archivo.txt', 'w') as archivo:
    archivo.write('Este es un ejemplo de escritura en un archivo de
texto en Python.')
```

3.- Agregar contenido a un archivo de texto

```
# Abre un archivo en modo agregar y agrega contenido al final
with open('archivo.txt', 'a') as archivo:
    archivo.write('\nEste es un nuevo texto agregado al archivo.')
```

4.- Lectura línea por línea

```
# Abre un archivo en modo lectura y lee línea por línea
with open('archivo.txt', 'r') as archivo:
    lineas = archivo.readlines()
    for linea in lineas:
        # Procesa cada línea como desees
        print(linea)
```

RESULTADOS Y CONCLUSIONES

Como pudimos ver en esta práctica ya vimos la manera en la que podemos implementar un diccionario en Python y también como que es que funcionan los archivos de texto en este mismo, por que cual también podemos ver cómo es que se puede almacenar la información que se genera mediante un programa en Python y esta se almacena en un archivo que se crea externo al programa con una terminación de nuestra preferencia, ya se texto, Excel, md, etc. Esto nos e muy útil ya que con esto no necesitamos pasar la información que generamos con el programa de forma manuela si no mediante un programa.