**﻿﻿Módulo 5.1**

**Manipulación de datos (Intro)**

**Archivos externos texto plano**

Hasta la fecha, hemos explorado cómo manipular datos utilizando objetos en Python. Sin embargo, Python es un lenguaje de programación de alto nivel que se utiliza para trabajar con grandes volúmenes de datos.

¿De dónde provienen estos datos?

Este es un aspecto fundamental a considerar en cualquier proyecto. En general, la fuente de datos depende del proyecto, del cliente o del programa en sí. Además, se deben tener en cuenta factores como la cantidad de datos, su tamaño, su disponibilidad y otros elementos relevantes. En Python, no existe una estructura de datos predefinida para el almacenamiento de datos. Tenemos la flexibilidad de guardar datos en diversos formatos, como texto plano, archivos de intercambio de información, bases de datos, DataFrames, hojas de cálculo y muchas otras opciones.

Esta versatilidad en la elección de la fuente y el formato de los datos es una de las fortalezas de Python y nos permite adaptarnos a las necesidades específicas de cada proyecto.

Daremos los primeros pasos con archivos de texto plano que podemos abrir directamente con el block de notas, vim o preferentemente cualquier IDE.

**Lectura de archivos de texto:**

|  |
| --- |
| **Características:**  **Legibilidad humana:** Los archivos de texto están diseñados para ser legibles por humanos, lo que significa que puedes abrirlos en un editor de texto y ver y entender el contenido sin necesidad de software especializado.  **Formato simple:** Los archivos de texto no tienen formato complejo, como colores, fuentes o estilos. La información se almacena en líneas de texto y, a menudo, se utiliza un formato de codificación de caracteres, como UTF-8, para admitir caracteres especiales y múltiples idiomas.  **Tamaño pequeño:** Los archivos de texto tienden a ser más pequeños en tamaño en comparación con otros formatos de archivo, como imágenes o archivos binarios, porque no almacenan información adicional de formato.  **Extensión de archivo común:** Los archivos de texto suelen tener extensiones de archivo comunes, como .txt en sistemas Windows o .txt en sistemas Unix/Linux.  **Edición manual:** Los archivos de texto se pueden editar fácilmente con programas de edición de texto, como Notepad en Windows, TextEdit en macOS o editores de texto en sistemas Unix/Linux.  **Portabilidad:** Los archivos de texto son altamente portátiles y pueden abrirse en una variedad de sistemas operativos y aplicaciones de software.  **Lenguaje de marcado:** Aunque los archivos de texto suelen contener texto sin formato, también se pueden utilizar para almacenar datos estructurados mediante lenguajes de marcado como XML o JSON.  **Compatibilidad universal:** Los archivos de texto son compatibles con una amplia gama de programas y lenguajes de programación. Son utilizados en programación para almacenar código fuente y datos legibles por máquina.  **No comprimidos:** Los archivos de texto no suelen estar comprimidos, lo que significa que no se utiliza compresión para reducir su tamaño. Sin embargo, se pueden comprimir si es necesario para ahorrar espacio.  **Interoperabilidad:** Los archivos de texto son ampliamente utilizados en la comunicación de datos entre sistemas y aplicaciones debido a su naturaleza simple y legible. |

Para poder manipular la información de archivos primero debemos abrir este. Hay varios medos dependiendo de que se desea hacer con el archivo

|  |  |  |
| --- | --- | --- |
| **mode** | | **descripción** |
| w | write | Escribe el contenido en un nuevo archivo lo crea |
| a | append | Agrega (al final) los datos deseado |
| r | read | Lectura de un archivo |
| x | create | Crea un archivo vacío. **(en desuso)** |

**Modo de escritura ('w'):**

Se utiliza para crear un archivo nuevo o sobrescribir un archivo existente si este ya existe ( su contenido se eliminará , no hay papelera de reciclaje ni crtl+z. Adiós información para siempre)

·

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | objeto\_python ="""  Rima XXI  ¿Qué es poesía?, dices, mientras clavas  en mi pupila tu pupila azul,  ¡Qué es poesía! ¿Y tú me lo preguntas?  Poesía... eres tú."""  print (f"{type(objeto\_python)=}")  with open("archivo.txt", **mode="w"**) as objeto\_io:  print (f"{type(objeto\_io)=}")  objeto\_io.write(objeto\_python)  print ("Archivo guardado") |   ·Salida esperada por consola   |  | | --- | | type(objeto\_python)=<class 'str'>  type(objeto\_io)=<class '\_io.TextIOWrapper'>  Archivo guardado |   · |

A ver….

Abre la carpeta donde esta guardado el script de Python, podrás encontrar ‘archivo.txt’. Ábrelo con el bloc de notas – note pad, vim o con cualquier programa que te ofrezca.

|  |
| --- |
| Para poder leer este archivo primero debe existir, por lo que corre el script anterior y verifica que exista ‘archivo.txt’ |

**Modo de escritura ('r'):**

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | with open("archivo.txt", **mode="r"**) as objeto\_io:  print (f"{type(objeto\_io)=}")  nuevo\_objeto\_python = objeto\_io.read()  print (f"{type(nuevo\_objeto\_python)=}")  print(f"el contenido del archivo es:{nuevo\_objeto\_python}") |   ·Salida esperada por consola   |  | | --- | | type(objeto\_io)=<class '\_io.TextIOWrapper'>  type(nuevo\_objeto\_python)=<class 'str'>  el contenido del archivo es:  Rima XXI  ¿Qué es poesía?, dices, mientras clavas  en mi pupila tu pupila azul,  ¡Qué es poesía! ¿Y tú me lo preguntas?  Poesía... eres tú. |   · |

·

Al usa un bloque n para garantizar que el archivo se cierre automáticamente después de su uso y no hace falta usar **close**.

**Modo de escritura ('a'):**

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | pie="Gustavo Adolfo Bécquer\n"  print("\*"\*50)  with open("archivo.txt", **mode= "a"**) as objeto\_io:  print (f"{type(objeto\_io)=}")  objeto\_io.write(pie)  print ("Archivo actualizado")  #··················································································  print("\*"\*50)  with open("archivo.txt", mode= "r") as objeto\_io:  print (f"{type(objeto\_io)=}")  nuevo\_objeto\_python = objeto\_io.read()  print (f"{type(nuevo\_objeto\_python)=}")  print(f"el contenido del archivo es:{nuevo\_objeto\_python}") |   ·Salida esperada por consola   |  | | --- | | type(objeto\_io)=<class '\_io.TextIOWrapper'>  type(nuevo\_objeto\_python)=<class 'str'>  el contenido del archivo es:  Rima XXI  ¿Qué es poesía?, dices, mientras clavas  en mi pupila tu pupila azul,  ¡Qué es poesía! ¿Y tú me lo preguntas?  Poesía... eres tú.  Gustavo Adolfo Bécquer |   · |

·

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | pie="Gustavo Adolfo Bécquer\n"  print("\*"\*50)  with open("archivo.txt", **mode= "a"**) as objeto\_io:  print (f"{type(objeto\_io)=}")  objeto\_io.write(pie)  print ("Archivo actualizado")  #··················································································  print("\*"\*50)  with open("archivo.txt", mode= "r") as objeto\_io:  print (f"{type(objeto\_io)=}")  nuevo\_objeto\_python = objeto\_io.read()  print (f"{type(nuevo\_objeto\_python)=}")  print(f"el contenido del archivo es:{nuevo\_objeto\_python}") |   ·Salida esperada por consola   |  | | --- | | type(objeto\_io)=<class '\_io.TextIOWrapper'>  type(nuevo\_objeto\_python)=<class 'str'>  el contenido del archivo es:  Rima XXI  ¿Qué es poesía?, dices, mientras clavas  en mi pupila tu pupila azul,  ¡Qué es poesía! ¿Y tú me lo preguntas?  Poesía... eres tú.  Gustavo Adolfo Bécquer |   · |

·

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | with open('archivo.txt', 'r') as objeto\_io:  for line in objeto\_io:  print(line)  #··················································································  print("\*"\*50)  with open('archivo.txt', 'r') as objeto\_io:  sector = objeto\_io.read(25)  print(f"{sector=}") |   ·Salida esperada por consola   |  | | --- | | Rima XXI¿Qué es poesía?, dices, mientras clavas  en mi pupila tu pupila azul,  ¡Qué es poesía! ¿Y tú me lo preguntas?  Poesía... eres tú.  Gustavo Adolfo Bécquer  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  sector='\nRima XXI\n¿Qué es poesía?'  <class '\_io.TextIOWrapper'> |   · |

·A veces, es más eficiente leer un archivo línea por línea, especialmente si es grande. Puedes hacerlo usando un bucle for:

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | with open('archivo.txt', 'r') as objeto\_io:  for linea\_objeto\_py in objeto\_io:  print(f"{linea\_objeto\_py}", end="")  print ("-"\*40)  #··················································································  print("\*"\*50)  with open("archivo.txt", 'r') as objeto\_io:  for linea\_objeto\_py in objeto\_io.readlines():  print(f"{linea\_objeto\_py}", end="")  print ("-"\*40) |   ·Salida esperada por consola   |  | | --- | | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ----------------------------------------  Rima XXI  ----------------------------------------  ¿Qué es poesía?, dices, mientras clavas  ----------------------------------------  en mi pupila tu pupila azul,  ----------------------------------------  ¡Qué es poesía! ¿Y tú me lo preguntas?  ----------------------------------------  Poesía... eres tú.  ----------------------------------------  Gustavo Adolfo Bécquer  ----------------------------------------  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ----------------------------------------  Rima XXI  ----------------------------------------  ¿Qué es poesía?, dices, mientras clavas  ----------------------------------------  en mi pupila tu pupila azul,  ----------------------------------------  ¡Qué es poesía! ¿Y tú me lo preguntas?  ----------------------------------------  Poesía... eres tú.  ----------------------------------------  Gustavo Adolfo Bécquer  ---------------------------------------- |   · |

·

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | with open('archivo.txt', 'r') as objeto\_io:  sector = objeto\_io.read(25)  print(f"{sector=}") |   ·Salida esperada por consola   |  | | --- | | sector='\nRima XXI\n¿Qué es poesía?'  **# ^ ^**  **# 0 carácter a 25** |   · |

·

Puedes realizar diversas operaciones en los datos leídos de un archivo de texto, como dividir líneas en palabras, realizar cálculos y crear nuevos archivos con los resultados. Por ejemplo, puedes contar la cantidad de caracter “e” hay en un archivo de texto:

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | with open("archivo.txt", 'r') as objeto\_io:  for linea\_objeto\_py in objeto\_io.readlines():  lista.append(linea\_objeto\_py)  print ("\*"\*40)  print (f"{lista}")  #··················································································  with open("archivo.txt", 'r') as objeto\_io:  lista = objeto\_io.read().split()  print ("\*"\*40)  print (f"{lista}") |   ·Salida esperada por consola   |  | | --- | | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ['\n', 'Rima XXI\n', '¿Qué es poesía?, dices, mientras clavas\n', 'en mi pupila tu pupila azul,\n', '¡Qué es poesía! ¿Y tú me lo preguntas?\n', 'Poesía... eres tú.\n', 'Gustavo Adolfo Bécquer\n']  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ['Rima', 'XXI', '¿Qué', 'es', 'poesía?,', 'dices,', 'mientras', 'clavas', 'en', 'mi', 'pupila', 'tu', 'pupila', 'azul,', '¡Qué', 'es', 'poesía!', '¿Y', 'tú', 'me', 'lo', 'preguntas?', 'Poesía...', 'eres', 'tú.', 'Gustavo', 'Adolfo', 'Bécquer'] |   · |

·

split(): Divide el contenido en palabras separadas por espacios en blanco y las almacena en una lista.

|  |  |  |
| --- | --- | --- |
| **mode** | | **descripción** |
| w+ | write | Permite abrir un archivo en modo de escritura y lectura |
| a+ | append | Permite abrir un archivo en modo de append y lectura |
| r+ | read | Permite abrir un archivo en modo de lectura y escritura |

·

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | with open('archivo.txt', 'a+') as archivo:  archivo.write('\nfin.')  archivo.seek(0) # Mover el cursor al principio del archivo  objeto\_python = archivo.read()  print (f"{type(objeto\_python)=}") |   ·Salida esperada por consola   |  | | --- | | Rima XXI  ¿Qué es poesía?, dices, mientras clavas  en mi pupila tu pupila azul,  ¡Qué es poesía! ¿Y tú me lo preguntas?  Poesía... eres tú.  Gustavo Adolfo Bécquer  **fin.** |   · |

·

**Binarios**

Los archivos binarios son aquellos que contienen datos en forma de bytes. Los archivos binarios pueden contener datos codificados, como imágenes, sonidos, videos y datos estructurados.

No se puede ver el contenido con note pad / bloc de notas

|  |  |  |
| --- | --- | --- |
| **mode** | | **descripción** |
| wb | write | Escribe el contenido en binario en un nuevo archivo lo crea |
| ab | append | Agrega en binario (al final) los datos deseado |
| rb | read | Lectura de un archivo escrito en binario |

·

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | objeto\_python\_bin ="""  Rima XXX  Asomaba a sus ojos una lágrima  y a mi labio una frase de perdón;  habló el orgullo y se enjugo su llanto  y la frase en mis labios expiró.  Yo voy por un camino: ella, por otro;  pero al pensar en nuestro mutuo amor,  yo digo aún, ¿por qué callé aquel día?  Y ella dirá, ,¿por qué no lloré yo?  """  objeto\_python\_bin=objeto\_python.encode('utf-8')  print (f"{type(objeto\_python\_bin)=}")  with open("archivo\_bin.txt", mode="wb") as objeto\_io:  print (f"{type(objeto\_io)=}")  objeto\_io.write(objeto\_python\_bin)  print ("Archivo guardado")  #··················································································  pausa()  limpiar()  print("\*"\*50)  with open("archivo\_bin.txt", mode= "rb") as objeto\_io:  print (f"{type(objeto\_io)=}")  nuevo\_objeto\_python\_bin = objeto\_io.read().decode('utf-8')  print (f"{type(nuevo\_objeto\_python\_bin)=}")  print(f"el contenido del archivo es:{nuevo\_objeto\_python\_bin}")  #··················································································  pausa()  limpiar()  pie="Gustavo Adolfo Bécquer\n".encode('utf-8')  print("\*"\*50)  with open("archivo\_bin.txt", mode= "ab") as objeto\_io:  print (f"{type(objeto\_io)=}")  objeto\_io.write(pie)  print ("Archivo actualizado")  #··················································································  pausa()  limpiar()  print("\*"\*50)  with open("archivo\_bin.txt", mode= "rb") as objeto\_io:  print (f"{type(objeto\_io)=}")  nuevo\_objeto\_python\_bin = objeto\_io.read().decode('utf-8')  print (f"{type(nuevo\_objeto\_python\_bin)=}")  print(f"el contenido del archivo es:{nuevo\_objeto\_python\_bin}")  #·················································································· |   ·Salida esperada por consola   |  | | --- | | type(objeto\_io)=<class '\_io.BufferedReader'>  type(nuevo\_objeto\_python\_bin)=<class 'str'>  el contenido del archivo es:  Rima XXI  ¿Qué es poesía?, dices, mientras clavas  en mi pupila tu pupila azul,  ¡Qué es poesía! ¿Y tú me lo preguntas?  Poesía... eres tú.  Gustavo Adolfo Bécquer |   · |

·

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | print('Creación de un archivo binario:')  nombre\_archivo = 'numeros.bin'  archivoBinario = open(nombre\_archivo, 'wb')  numeros = [2, 3, 5, 7, 11]  archivoBinario.write(bytearray(numeros))  archivoBinario.close()  print ("Archivo escrito-grabado y cerrado ")  #··················································································  archivoBinario = open(nombre\_archivo, 'rb')  objeto\_numeros= archivoBinario.read()  print (b"objeto\_numeros:",objeto\_numeros)  print ("Archivo Abierto, leido - y cerrado ")  print ("esto esta en el archivo numeros.bin y lo ingresamos a py:",objeto\_numeros) |   ·Salida esperada por consola   |  | | --- | | Creación de un archivo binario:  Archivo escrito-grabado y cerrado  b'objeto\_numeros:' b'\x02\x03\x05\x07\x0b'  Archivo Abierto, leido - y cerrado  esto esta en el archivo numeros.bin y lo ingresamos a py: b'\x02\x03\x05\x07\x0b' |   · |

·

|  |  |
| --- | --- |
| **close()**  Es imperativo que una vez que se hayan realizado todas las operaciones de entrada y de salida de archivos, este sea cerrado de manera adecuada. En caso de no hacerlo, es altamente probable que el archivo se encuentre en un estado inestable y corra riesgo de que la información contenida se corrompa o destruya. | |
| closed()  writable()  readable()  seekable()  tell()  seek( offset[, whence]):  read(size)  readline()  readlines()  write(string)  writelines(list) | En desuso desde la utilización de with  Devolverá \_True\_ si el archivo está cerrado  Devolverá \_True\_ si el archivo está en modo de escritura.  Devolverá \_True\_ si el archivo está en modo de lectura.  Devolverá \_True\_ si es posible desplazarse dentro del archivo.  Devolverá la posición actual del archivo.  Cambia la posición actual del archivo a offset.  El parámetro opcional whence puede ser  0 (inicio del archivo),  1 (posición actual) o  2 (final del archivo).  Lee y devuelve hasta size bytes del archivo.  Si no se especifica ningún tamaño, se lee todo el archivo.  Lee y devuelve una línea del archivo.  Lee y devuelve todas las líneas del archivo como una lista.  Escribe la cadena string en el archivo.  Escribe todas las cadenas en la lista list en el archivo. |

Ejemplo:

|  |  |  |
| --- | --- | --- |
| ·Código Python   |  | | --- | | def operaciones\_sobre\_txt(archivo,modo,cant=""):  try:  objeto\_python="si es solo lectura 'objeto\_python ' no tiene info\n si es una segunda operacio debes ubicar seek al inicio"  texto="Si se corta la luz\nya no perdemos nada"  with open (archivo, mode=modo) as io\_obj:  print (f"Antes: me encuentro en la posicion {io\_obj.tell()}")  if modo == "x":  print ("Se creo vacio")  elif modo.startswith("w"):  print ("Se creo y se ingreso una linea al inicio")  io\_obj.write(texto)  elif modo.startswith("a"):  print ("Se agrego una linea al final del archivo si este no estaba creado se genero")  io\_obj.write("\nFin....")  if modo == "r" or "+" in modo:  io\_obj.seek(0)  if cant=="":  objeto\_python = io\_obj.read()  elif isinstance (cant,int):  objeto\_python = io\_obj.read(cant)  elif cant == "xlinea":  modo="readline"  objeto\_python = ""  for cada\_linea in io\_obj.readlines():  objeto\_python+=cada\_linea  print("\n"\*2)  print (f"El contenido es :\n{objeto\_python}")  print("\n"\*2)  print (f"Despues: me encuentro en la posicion {io\_obj.tell()}")  # tabular dentro y fuera del with <---->  print (f"mode = {modo}".center(20))  print (f"estado de cierre {io\_obj.closed=}")  print (f"estado de escritura {io\_obj.writable()=}")  print (f"estado de lectura {io\_obj.readable()=}")  print (f"estado de desplazamiento {io\_obj.seekable()=}")  except Exception as Error:  print (f"Error tipo {Error}")  finally:  print ("-"\*50)  operaciones\_sobre\_txt("Mineria.txt","x")  operaciones\_sobre\_txt("Mineria.txt","w")  operaciones\_sobre\_txt("Mineria.txt","w+")  operaciones\_sobre\_txt("Mineria.txt","a+")  operaciones\_sobre\_txt("Mineria.txt","r")  operaciones\_sobre\_txt("Mineria.txt","r",18)  operaciones\_sobre\_txt("Mineria.txt","r","xlinea") |   ·Salida esperada por consola   |  | | --- | | Verificar resultados con tabla.  Ejecutar 2 veces y chequear mode “x” |   · |