

Lectura / Escritura de Archivos (Ficheros)

Ejemplos de Archivos de Textos Planos (Ficheros)

Busca en el documento "Listado de Comandos", como puedes crear tu propio archivo de texto en una carpeta de Windows, ya que será necesario para poder trabajar con este tipo de archivos.

Crear un Archivo de texto, en la carpeta de tu elección y colócale el nombre "probando.txt". Escribe en su interior el contenido que puedes ver a la derecha (**no lo copies, escríbelo**) y anota la dirección donde lo guardaste, a continuación, cambia las direcciones que usan los programas **1a** y **1b** en el comando "**with open**" por la tuya.

Contenido del Archivo

Línea 01
Línea 02
Línea 03
Línea 04

Archivos_001_TXT_Plano_Lectura_001

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Lectura de un Archivo txt -
@author: Horacio
"""
print("-----")
print(" Lectura de un Archivo txt.")
print(" Cuidado, lee atentamente el funcionamiento de la sentencia ")
print("f.read( ) en el documentos Listado de Comandos")
print("-----")

with open('d:\Programacion\probando.txt', 'r') as f:
    contenido = f.read()
    print(contenido)
    f.close()

# Termina Programa
print("\n-----")
print("Programa Terminado.")

# Se recomienda ampliar información referente a los comandos
# usados, puedes consultar el el documento "Listado de Comandos".
```

Archivos_001_TXT_Plano_Lectura_002

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Lectura de un Archivo txt - línea por línea.
@author: Horacio
"""
# Uso del import time, para generar tiempo de espera entre línea
# y línea
import time

# Guardo Ruta y Nombre del Archivo en la Variable:
Nombre_Archivo = "d:\M_de_Falla\Programacion\ probando.txt"

print("-----")
print(" Lectura de un Archivo de texto (línea por línea)")
print(" Hay un tiempo de espera de 3 segundos entre línea y línea")
print(" para permitir que veas lo que se lee desde archivo")
print("-----")

print("Abro archivo físico ", Nombre_Archivo)
print("y asigno dirección al puntero Mi_Archivo_TXT")
Mi_Archivo_TXT = open( Nombre_Archivo, 'r') # Abro archivo
print("-----\n")

try:
    for linea in Mi_Archivo_TXT: # Leo una línea del archivo
        print(linea) # Muestro línea leída
        time.sleep(3) # Demora de tres segundos
finally:
    print("\n-----")
    Mi_Archivo_TXT.close() #Cierro archivo
    print("Termino Lectura y Cierro Archivo")

# Termina Programa
print("\n-----")
print("Programa Terminado.")
```

Lectura / Escritura de Archivos (Ficheros)

Ejemplos de Archivos de Textos Planos (Ficheros)

Archivos_001_TXT_Plano_Lectura_003

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Lectura de un Archivo txt - línea por línea mientras se carga en lista
@author: Horacio
"""

# Uso del "import" time, para generar tiempo de espera entre
# línea y línea
import time

# Define lista vacía
Comandos = []

# Armo dirección y nombre de Archivo para trabajar
Camino = "d:\M_de_Falla\Programacion\Prog_2022"
Nombre = "\probando.txt"
Archivo = Camino + Nombre

print("-----")
print("Lee Archivo de texto (línea a línea) y lo cargas en una lista")
print(" Con espera de 2 segundos antes de leer próxima línea")
print(" para permitir que visualices lo que se lee desde archivo")
print("-----")

print("Abro archivo físico: ", Archivo)
print("y asigno dirección al puntero f")
f = open( Archivo, 'r') # Abro archivo físico y asigno al puntero f
print("-----\n")

try:
    for linea in f: #Leo UNA Línea del Archivo
        Comandos.append(linea) #Guardo línea leída en Lista
        print(linea) #Muestro por pantalla línea leída
        time.sleep(2) #Tiempo de espera

finally:
    print("\n-----")
    f.close() # Cierro archivo
    print("Termino Lectura del Archivo y lo Cierro")
print("\n-----")

# Muestro todos los elementos de la lista
print("Muestro todos los elementos Cargados en la lista")
for i in range( len(Comandos) ):
    Cadena = "Elemento " + str(i+1) + ": "
    print(Cadena, Comandos[i])

# Termina Programa
print("\n-----")
print("Programa Terminado.")
```

Lectura / Escritura de Archivos (Ficheros)

Ejemplos de Archivos de Textos Planos (Ficheros)

Archivos_002_TXT_Plano_Escritura_001

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Saludar al Mundo al generar Primer Archivo con Python
@author: Horacio
"""
print("-----")
print(" Saludar al Mundo al generar Primer Archivo con Python.")
print("-----")

# Contruyo direccion y nombre de Archivo con el que trabajaré
Camino = "d:\M_de_Falla\Programacion\Prog_2022"
Nombre_txt = "probando.txt" # Nombre Físico archivo de Texto
Mi_Archivo_txt = Camino + Nombre_txt

# f_txt es el Nombre lógico, puntero al archivo u objeto con el que
# trabajamos. En esta sentencia Abriremos el archivo físico que
# esta en nuestro HD (en modo solo lectura) el que mientras se
# ejecute el programa y lo mantengamos abierto, será conocido
# por el programa como f_txt (es decir su nombre lógico) y que
# Python le asigna funciones (o métodos o comandos u ordenes)
# que nos servirán para poder trabajar con el

with open( Mi_Archivo_txt, 'w') as f_txt:
    f_txt.write('Hola mundo\n') # Termina la línea con el \n

f_txt.close()
print("¿ y que paso ?")

# Termina Programa, y acabas de perder el primer archivo de
# texto que escribiste para los ejercicios anteriores. :)
print("\n-----")
print("Programa Terminado.")
print("Ahora, Rápido, ve a fijarte como quedo el archivo.")

# Se recomienda ampliar información referente a los comandos
# usados, puedes consultar el documento "Listado de Comandos".
```

Archivos_002_TXT_Plano_Escritura_002#

```
 -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Grabar Archivo con datos de una lista - Opción I - Usaremos
sentencia print() - Método rápido para grabación de archivos
desde una lista
@author: Horacio
"""
print("-----")
print("Grabar Archivo con datos de una lista - Opcion I.")
print("      Usaremos sentencia print()")
print("-----")

# Definir lista con nombres que usaremos grabaremos en archivo
nombres = ["Antonio", "Andrea", "Juan", "Susana", "Pedro", "Melanie",
"HPC"]

# Abrimos en modo escritura
archivo = open("personas.txt", "w")

# Recorremos la lista y agregamos cada nombre al archivo
print("Método escritura A")
for nombre in nombres:
    print(nombre, file=archivo)

# Recuerda siempre cerrar el archivo
archivo.close()

# Termina Programa
print("\n-----")
print("Programa Terminado.")

# Se recomienda ampliar información referente a los comandos
# usados, puedes consultar el documento "Listado de Comandos".
```

Lectura / Escritura de Archivos (Ficheros)

Ejemplos de Archivos de Textos Planos (Ficheros)

Archivo_003_TXT_Plano_Lect_Escrit_001

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Lectura de un Archivo txt - línea por línea mientras se carga en lista
con posterior grabación de la lista en nuevo archivo. Método
completo de grabación
@author: Horacio
"""

# import time, para generar tiempo de espera entre línea y línea
import time

# Define lista vacía
Comandos = []

print("-----")
print("Leer Archivo texto (línea por línea) que se carga en una lista")
print(" Luego, grabar todos los elementos de la lista en otro archivo")
print(" - Opciones II y III - ")
print("-----\n")

# Construyo dirección y nombre de los Archivos para trabajar
Camino = "d:\M_de_Falla\Programacion\Prog_2022"

Nomb01 = "\Personas.txt"
Archi01 = Camino + Nomb01
Nomb02 = "\Datos_02.txt"
Archi02 = Camino + Nomb02
Nomb03 = "\Datos_03.txt"
Archi03 = Camino + Nomb03

print("Abro (modo lectura) archivo físico ", Archi01)
f = open( Archi01, 'r') # Abro archivo físico y asigno al puntero f
try:
    for linea in f:
        # Leo Línea del Archivo
        Comandos.append(linea) # Guardo línea leída en Lista
        print(linea) # Muestro por pantalla línea leída
finally:
    f.close() # Cierro archivo
    print("Termino lectura y Cierro archivo físico ", Archi01)

time.sleep(1) # Tiempo de espera
print("\n-----")
print("Método escritura B")
print("Abro (modo escritura) archivo físico ", Archi02)
with open( Archi02, 'w') as f_txt_02:
    print("Comienzo Grabación Elementos de la Lista")
    for i in range( len(Comandos) ):
        f_txt_02.write(Comandos[i])

f_txt_02.close() # Cierro archivo
print("Termino Grabación y Cierro archivo físico ", Archi02)

time.sleep(1) # Tiempo de espera
print("\n-----")
print("Método escritura C")
print("Abro (modo Escritura) archivo físico ", Archi03)
f_txt_03 = open( Archi03, 'w') # Abro archivo físico y asigno a f
try:
    print("Comienzo Grabación Elementos de la Lista")
    for i in range( len(Comandos) ):
        f_txt_03.write(Comandos[i])
finally:
    f_txt_03.close() # Cierro archivo
    print("Termino escritura y Cierro archivo físico ", Archi03)

# Termina Programa, y acabas de perder el primer archivo de
# texto que escribiste para los ejercicios anteriores. :)
print("\n-----")
print("Programa Terminado.")
print("Ahora, Rápido, ve a fijarte como quedaron los archivos.")
```

Archivos_004_TXT_Plano_Agrega_Registro_001

```
# -*- coding: utf-8 -*-
"""
Creado 24/04/2022
Agregar Registros en una Archivo de Texto
@author: Horacio
"""

# Uso del import time, para generar tiempo de espera
import time

print("-----")
print(" Crear Archivo de texto con varios registros. CERRARLO")
print(" y luego AGREGAR más Registros al Archivo")
print("-----\n")

# Definir lista con nombres que usaremos grabaremos en archivo
Nombres = ["Athos", "Porthos", "Aramís", "D'Artagnan",
           "Milady de Winter", "Ana de Austria", "H.P.C"]

Mas_Nombres = ["Fernando", "Alejandra", "Antonio",
               "Susana", "José", "Claudia"]

Todos = []

# Construyo dirección y nombre de los Archivos para trabajar
Camino = "d:\M_de_Falla\Programacion\Prog_2022"
Nomb01 = "\Datos_01.txt"
Archi01 = Camino + Nomb01

print("\n-----")
print("Creo Archivo: ", Archi01)
with open( Archi01, 'w') as f_txt_01:
    print("Comienzo Grabación Elementos de la Lista: Nombres"])
    for i in range( len(Nombres) ):
        f_txt_01.write(Nombres[i] + '\n')

f_txt_01.close() # Cierro archivo
print("Termino Primera Grabación y Cierro Archivo: ", Archi01)
time.sleep(1) # Tiempo de espera

print("\n-----")
print("Agrego Registros en Archivo: ", Archi01)
with open( Archi01, 'a') as f_txt_01:
    print("Agregando Registros de Lista: Mas_Nombre"])
    for i in range( len(Mas_Nombres) ):
        f_txt_01.write(Mas_Nombres[i] + '\n')

f_txt_01.close() # Cierro archivo
print("Termino Segunda Grabación y Cierro archivo: ", Archi01)
time.sleep(1) # Tiempo de espera

print("\n-----")
print("Abro (modo lectura) Archivo: ", Archi01)
print("Comienzo Carga de registros en Lista: Todos"])
f_txt_01 = open( Archi01, 'r') # Abro Archivo y asigno al puntero
try:
    for registro in f_txt_01:
        # Leo Línea del Archivo
        Todos.append(registro) # Guardo línea leída en Lista
finally:
    f_txt_01.close() # Cierro archivo
    print("Termino lectura y carga en lista - Cierro Archivo:", Archi01)

time.sleep(1) # Tiempo de espera

# Termina Programa. :)
print("\n-----")
print("Programa Terminado.")
print("Ahora, Rápido, ve a fijarte como quedo el archivo: ", Archi01)
```

Lectura / Escritura de Archivos (Ficheros)

Ejemplos de Archivos de Textos Con Estructura (Ficheros)

Archi_005_TXT_Estructura_Lect_Escrit_001

```
##*- coding: utf-8 -*-
```

```
"""
```

```
Creado 24/04/2022
```

Escritura y Lectura de un Archivo txt con estructura de datos - Cada registro del archivo debe tener los campos separados por comas y el siguiente formato:

[(nombre1,puntaje1,tiempo1)].

@author: Horacio

```
"""
```

```
print("-----")
print(" Crear archivo con registros de 3 campos, luego recuperarlo")
print(" cargando en lista cada registro, finalmente mostrar lista.")
print("-----")
```

```
import time
```

```
def guardar_puntajes(nombre_archivo, puntajes):
```

```
    """ Guarda datos contenidos en lista (Matriz) en archivo de texto.
```

```
    La matriz o arreglo puede contener Nx3 elementos (Nombre, Puntos y hora de finalización) y es recibida en la función como parámetro, junto con la ruta de acceso y nombre del archivo.
```

```
    MUY IMPORTANTE: los valores que se guardan en el archivo, se encuentran separados por comas. La lista devuelta tiene el siguiente formato:
```

```
    [(nombre1,puntaje1,tiempo1),(nombre2,puntaje2,tiempo2)].
```

```
    """
```

```
    archivo = open(nombre_archivo, "w+")
```

```
    for nombre, puntaje, tiempo in puntajes:
```

```
        archivo.write(nombre+","+str(puntaje)+","+tiempo+"\n")
```

```
    archivo.close()
```

```
def recuperar_puntajes(nombre_archivo):
```

```
    """ Lee Archivo de texto. Recupera los datos y guarda en lista (matriz o arreglo). Retorna lista con los valores restaurados.
```

```
    MUY IMPORTANTE: el archivo contiene los datos con el mismo formato con el que se grabo, es decir, los datos están separados por comas. La lista devuelta tiene siguiente formato:
```

```
    [(nombre1,puntaje1,tiempo1),(nombre2,puntaje2,tiempo2)].
```

```
    """
```

```
    puntajes = []
```

```
    archivo = open(nombre_archivo, "r+")
```

```
    for línea in archivo:
```

```
        nombre, puntaje, tiempo = línea.rstrip("\n").split(",")
```

```
        puntajes.append((nombre,int(puntaje),tiempo))
```

```
    archivo.close()
```

```
    return puntajes
```

```
# definición de lista (arreglo) con sus elementos, esta es la que será grabada en el archivo
```

```
valores = [("Athos", 1080, "4:16"),
           ("Porthos", 2315, "8:42"),
           ("Dartagnan", 4235, "22:15")]
```

```
# Nombre del Archivo que usaremos
```

```
Archivo = "puntajes.txt"
```

```
# Llamada a la función de guarda datos en el archivo
```

```
guardar_puntajes(Archivo, valores)
```

```
# Llamada a la función de RECUPERA datos del archivo
```

```
Matriz = recuperar_puntajes(Archivo)
```

```
# Muestro elementos de lamatriz, uno por uno
```

```
print( Matriz[0][0], Matriz[0][1], Matriz[0][2],)
```

```
print( Matriz[1][0], Matriz[1][1], Matriz[1][2],)
```

```
print( Matriz[2][0], Matriz[2][1], Matriz[2][2],)
```

```
print("\n-----\n")
```

```
# Muestro elementos de la matriz (lista) mientras recorro filas y
```

```
# columnas con dos for anidados
```

```
for i in range(len(Matriz)):
```

```
    for j in range(len(Matriz[i])):
```

```
        print(Matriz[i][j], end = '')
```

```
        time.sleep(1) # Espera dos segundos
```

```
    print("\n") # Bajaré dos renglones
```

Lectura / Escritura de Archivos (Ficheros)

Ejemplos de Archivos de Textos Con Estructura (Ficheros)

```
# Termina Programa
print("\n-----")
print("Programa Terminado.")
```

Ejemplos de Archivos Binarios Planos y Con Estructura (Ficheros)

```
Archivos_006_Binario_Plano_Lectura_Escritura_001
#-*- coding: utf-8 -*-
"""
Creado 24/04/2022
Grabación y lectura de un Archivo Binario con números contenidos en una lista
@author: Horacio
"""
print("-----")
print(" Crear archivo binario con valores numéricos contenidos")
print(" en una lista. Luego de finalizada la grabación del archivo, ")
print(" leer y cargar los datos del archivo en nueva lista finalmente")
print(" mostrarla por pantalla, primero elemento a elemento, y ")
print(" después recorrerla con un for mientras se muestran todos")
print(" los elementos.")
print("-----\n")
import time
import pickle
def guardar_puntajes(nombre_archivo, puntajes):
    """ La función, guarda en archivo Binario el contenido de una lista de números reales, recibida como parámetro """
    archivo = open(nombre_archivo, "wb+")
    pickle.dump(puntajes, archivo)
    archivo.close()
def recuperar_puntajes(nombre_archivo):
    """ La función, recupera de un archivo binario, los datos que son cargados en un listado (lista numérica), que al finalizar la carga retorna al programa principal """
    archivo = open(nombre_archivo, "rb+")
    puntajes = pickle.load(archivo)
    archivo.close()
    return puntajes
# definición de lista con sus elementos, esta es la que será # grababa en el archivo
valores = [10, 3.5, 2315, 5.3, 34, 17, 32.24]
# Nombre del Archivo que usaremos
Archivo = "Numeros.bin"
# Llamada a la función de guarda datos en el archivo
guardar_puntajes(Archivo, valores)
# Llamada a la función de RECUPERA datos del archivo
Lista = recuperar_puntajes(Archivo)
# Muestro elementos de la lista, uno por uno
print( Lista[0])
print( Lista[1])
print( Lista[2])
print( Lista[3])
print( Lista[4])
print( Lista[5])
print( Lista[6])
print("\n-----\n")
# Muestro los elementos de la lista mientras la recorro con un for
for i in range(len(Lista)):
    print(Lista[i])
    time.sleep(1) #Espera un segundo
```

```
Archivos_007_Binario_Estructura_Lect_Escrit_001
"""
Creado 24/04/2022
Grabación y lectura de un Archivo Binario con los datos de una lista matricial de Nx5. la lista puede ser ampliada a NxM filas y Columnas.
@author: Horacio
"""
print("-----")
print(" Crear un archivo con registro de 5 campos, luego recuperarlo")
print(" cargándolo en una lista, finalmente mostrar.")
print(" Presta mucha atención al formato de la lista.")
print("-----")
import time
import pickle
def guardar_puntajes(nombre_archivo, puntajes):
    """ La función Guarda en archivo Binario (con formato pickle) el contenido de una lista de números reales recibida como parámetro """
    archivo = open(nombre_archivo, "wb+")
    pickle.dump(puntajes, archivo)
    archivo.close()
def recuperar_puntajes(nombre_archivo):
    """ La función recupera de un archivo binario (con formato pickle) un listado de números, cargándolos en una lista que retorna al programa principal cuando termina. """
    archivo = open(nombre_archivo, "rb+")
    puntajes = pickle.load(archivo)
    archivo.close()
    return puntajes
# definición de lista (arreglo) con sus elementos, esta es la que será # grababa en el archivo
valores = [("Athos", 1001, "13:05", "El ideal de la nobleza", 3.5),
            ("Porthos", 1002, "8:22", "Mosquetero", 5.3),
            ("Aramis", 1003, "14:15", "Mosquetero", 7.24),
            ("D'Artagnan", 1004, "9:32", "Intrépido", 9.24),
            ("Milady de Winter", 1005, "15:25", "La Mala", 12.24),
            ("Constance Bonacieux", 1006, "10:42", "Guardarropas", 2.15),
            ("Ana de Austria", 1007, "16:35", "La Reina", 13.25),
            ("H.P.C", 1008, "11:52", "Todo un Misterio!?", 14.35)]
# Nombre del Archivo que usaremos
Archivo = "puntajes.bin"
# Llamada a la función de guarda datos en el archivo
guardar_puntajes(Archivo, valores)
# Llamada a la función de RECUPERA datos del archivo
Matriz = recuperar_puntajes(Archivo)
# Muestro elementos de la matriz, uno por uno
print( Matriz[0][0], Matriz[0][1], Matriz[0][2], Matriz[0][3])
print( Matriz[1][0], Matriz[1][1], Matriz[1][2], Matriz[1][3])
print( Matriz[2][0], Matriz[2][1], Matriz[2][2], Matriz[2][3])
print( Matriz[3][0], Matriz[3][1], Matriz[3][2], Matriz[3][3])
print( Matriz[4][0], Matriz[4][1], Matriz[4][2], Matriz[4][3])
print("\n-----\n")
# Muestro elementos de la matriz (lista) mientras recorro filas y # columnas con dos for anidados
for i in range(len(Matriz)):
    for j in range(len(Matriz[i])):
```

Lectura / Escritura de Archivos (Ficheros)

<pre># Termina Programa print("\n-----") print("Programa Terminado.") # Se recomienda ampliar información referente a los comandos # usados, puedes consultar el documento "Listado de Comandos".</pre>	<pre>print(Matriz[i][j], end = ' ') time.sleep(1) # Espera dos segundos print("\n") # Bajará dos renglones # Termina Programa print("\n-----") print("Programa Terminado.")</pre>
--	--

Ejemplos de Archivos: Búsquedas y Controles Especiales

<pre>Archivos_000_TXT_Especial_001_Verifico_Existencia # -*- coding: utf-8 -*- """ Creado 24/04/2022 Verifica Existencia del archivo de trabajo - Dos ejemplos. @author: Horacio """ # Uso del "import" time, p/generar una espera entre línea y línea import time import os # Armo dirección y nombre de Archivo para trabajar Camino = "d:/M_de_Falla/Programacion/Prog_2022/" Nombre = "probando.txt" Archivo = Camino + Nombre # Define lista vacía Comandos = [] print("-----") print("Verifica Existencia del archivo de trabajo y si existe, Lee") print("Archivo de texto (línea a línea) y lo cargas en una lista") print("Con espera de 1 segundo antes de leer próxima línea para") print("permitir que visualices lo que se lee desde archivo") print("-----") print("\n***** Ejemplo 01 *****") print("Verificación de la existencia del archivo de trabajo: ", Archivo) if os.path.isfile(Archivo): print("El Archivo Si Existe, procedemos a la apertura") f = open(Archivo, 'r') # Abro archivo físico y asigno print("-----\n") try: for linea in f: #Leo UNA Línea del Archivo Comandos.append(linea) #Guardo línea leída en Lista print(linea) #Muestro por pantalla línea leída time.sleep(1) #Tiempo de espera finally: print("\n-----") f.close() # Cierro archivo print("Termino Lectura del Archivo y lo Cierro") print("\n-----") # Muestro todos los elementos de la lista print("Muestro todos los elementos Cargados en la lista") for i in range(len(Comandos)): Cadena = "Elemento " + str(i+1) + ": " print(Cadena, Comandos[i]) else: print("El archivo no existe, se encontró un error tipo IOError ") print("***** Termina Ejemplo 01") print("*****") print("\n***** Ejemplo 02 *****") print("Nuevamente Verifica existencia del archivo: ", Archivo) try: f = open(Archivo) print("El Archivo Si Existe, procedemos a Trabajar Normalmente") #Aca se hace todo lo que se quiere hacer con datos del archivo f.close() except IOError as io: print("error tipo: ",io) #Muestro por pantalla el error else:</pre>	<pre>Archivos_000_TXT_Especial_002_Busco_Archivo # -*- coding: utf-8 -*- """ Creado 24/04/2022 Busco el Archivo en el Disco de la PC @author: Horacio """ print("-----") print("Busca archivo determinado en la carpeta o Unidad") print("Especificada dentro del Disco de la PC") print("-----") import os def Busca_Archivo(ArchiBuscado, AcaBusca): for dirpath, dirname, filename in os.walk(AcaBusca): print("Buscando en: ", os.path.join(dirpath, ArchiBuscado)) if ArchiBuscado in filename: return os.path.join(dirpath, ArchiBuscado) RutaEncontrada = Busca_Archivo("Probando.txt", "D:/M_de_Falla/") if RutaEncontrada == None: print("\n*****") print("***** Archivo NO ENCONTRADO") else: print("\n*****") print("***** Archivo Encontrado en: ", RutaEncontrada) # Termina Programa print("\n-----") print("Programa Terminado.") # Se recomienda ampliar información referente a los comandos # usados, puedes consultar el documento "Listado de Comandos".</pre>
--	--

Lectura / Escritura de Archivos (Ficheros)

```
print("Continúo trabajando, No Hay error.")
print("Podremos continuar procesando Datos del Archivo")

print("***** Termina Ejemplo 02")

# Termina Programa
print("\n-----")
print("Programa Terminado.")
```