

```
file = open('archivo.txt', 'w')
file.close()
```

Selecciona todos los que correspondan.

- ☐ La ruta donde se ubica el archivo es absoluta
- ☒ La ruta donde se ubica el archivo es relativa
- ☒ Cualquier información que exista en ese archivo se perderá
- ☒ El código es útil para crear un archivo nuevo

Lo que pasa es que si existe el archivo, lo sobrescribe, entonces se elimina la info que tenia previamente y se perderá. Si hubiese querido agregar info tenia que usar 'a' (append)

La direccion 'archivo.txt' es relativa porque no es una direccion completa, entonces se asume que la informacion esta en la carpeta donde estamos parados y no desde la raiz.

el resultado de ejecutar el siguiente código

```
file = open('archivo.txt', 'a')
file.write("")
texto = file.read()
print(texto)
```

Marca solo un óvalo.

- ☐ Se leerá un archivo vacío
- ☐ Se leerá lo que contenía el archivo antes de ser abierto
- ☐ Se leerá la primera línea del archivo
- ☒ Dará un error

Dará un error. Esto pasa porque faltó cerrarlo y luego volver a abrirlo

para poder leer esos cambios

deberiamos haber hecho:

```
file.close()
file = open(filename, "r")
```

el read() lee todo el archivo pero lo hace en una sola cadena

el readline() lee por cada linea

el readlines() almacena en una lista y cada elemento del archivo se comporta como una fila de la lista

.read() te da esto 'ola\ncasa\neco'

es como un text to columns de excel haciendo por separacion de espacios

```
lista1 = ['Un', 'dia', 'lindo']
```

```
file1 = open("myfile.txt", "w")
file1.writelines(lista1)
file1.close()
```

```
file1 = open("myfile.txt", "r")
texto = file1.readline()
for linea in file1.readlines():
    print(linea, end=' ')
```

```
print(texto, end=' ')
```

Marca solo un óvalo.

- ☐ un día lindo
- ☐ día lindo
- ☒ No se imprime nada

- ☐ un día lindo
☐ día lindo
☒ No se imprime nada
☐ Error de lectura

para mi tira error xq no podes agregar a fila1.writelines la lista, porque creo que solo podes agregar str --> NO

writelines es un iterable asi que va a escribirlo como Undialindo en el archivo

https://www.w3schools.com/python/ref_file_writelines.asp

DEBUGGING:

breakpoint es para ejecutar el programa hasta ese punto y apartir de ese punto ir linea por linea o entrar a una funcion

Apretar "Run and debug" y elegir python file

El step over es correr toda la funcion de una, sin pasar step by step

El step into es para entrar linea por linea

En watch se ponen las variables que siempre quiero saber que informacion tienen almacenadas. Click derecho sobre el codigo y elijo "add to watch".

el readline() es como un puntero que se va ubicando en las posiciones del archivo, entonces si tiene una sola linea lo leo y avanza a la siguiente linea. En el ejemplo como no hay nada despues de la primera linea, termina en el primer loop.

[https://www.tutorialspoint.com/python/file_next.htm#:~:text=Python%20file%20method%20next\(\),\(\)%20does%20not%20work%20right.](https://www.tutorialspoint.com/python/file_next.htm#:~:text=Python%20file%20method%20next(),()%20does%20not%20work%20right.)

4. Indique el resultado de ejecutar el siguiente código

```
try:
    resultado = 2 + '1'
except:
    resultado = 2 + int('1')
    print("2 + 1 es", resultado)
```

Marca solo un óvalo.

- ☐ Ocurre un error que interrumpe la ejecución
☒ Se imprime dos + uno es tres
☐ El código en el bloque except se ejecuta
☐ Ninguna de las anteriores

Salta un error porque esta sumando un int con un string, entonces va al except y luego el resultado hace el calculo = 3 y el print da "2+1 es 3"

5. Indique
Qué hace el siguiente código

```
while True:
    try:
        dni=int(input("ingrese su dni "))
        caddni=str(dni)
        loncaddni=len(caddni)
        break
    except ValueError as e:
        print("Este es el error.....",e)
```

Marca solo un óvalo.

- ☒ Produce un error al ingresar un valor errado
☐ Nunca produce el error ValueError
☐ El dni no debe ser una cadena
☐ Ninguno de los anteriores. Paso

Tira un error al ingresar un valor errado. Si ingreso bien el DNI, se produce un ValueError, lo atrapa y lo muestra.

porque si ingresas un dato mal tipo "aaa" no lo convierte a int

Si no pongo el break se hace un loop infinito porque siempre es correcta la info. Ese while es necesario porque permite poder volver a ingresar el valor en input de nuevo.

```
from io import UnsupportedOperation

def existe(patharchivo):
    try:
        archivo=open(patharchivo,"r")
        return archivo
    except FileNotFoundError :
        return None
    except UnsupportedOperation:
        print("No puede hacer esta tarea con este archivo")
```

Marca solo un óvalo.

- ☐ No hace nada... no le gusta trabajar
- ☐ Verifica si se existe un archivo, si existe devuelve None
- ☐ Siempre devuelve None
- ☐ No funciona
- ☒ Ninguna de las anteriores

La funcion agarra la ruta del archivo, si lo puede leer, lo muestra, y sino, si no lo encuentra, devuelve None. Si no lo puede abrir (unsopported), entonces devuelve "no puede hacer...". Entonces hace algo.

7. Cual es la diferencia entre abrir los archivos usando with y open

Marca solo un óvalo.

- ☐ No hay ninguna diferencia
- ☐ Una empieza con w y la otra con o
- ☐ open te obliga a cerrar el archivo
- ☒ with te obliga a cerrar el archivo
- ☐ Ninguna de las anteriores

31/10/22, 10:53

ARCHIVOS / EXCEPCIONES

B. Con archivos csv estamos simulando el comportamiento de una base de datos o una table excel

Marca solo un óvalo.

- ☒ True
- ☐ False
- ☐ Paso, no me acuerdo

9. Que hace este codigo:

```
with open('archivoscsv.csv','w',newline='') as csvfile:
    encabezados=['NOMBRE','APELLIDO','RESULTADO']
    writer=csv.DictWriter(csvfile,fieldnames=encabezados)
    writer.writeheader()
    writer.writerow(
        {'NOMBRE': 'Mauricio',
        'APELLIDO': 'Jaimes',
        'RESULTADO': 100}
    )
```

Marca solo un óvalo.

- ☐ Abre un archivo y escribe NOMBRE,APELLIDO,RESULTADO
- ☐ Abre un archivo y escribe en el NOMBRE:'Mauricio', APELLIDO
Jaimes',RESULTADO:100
- ☐ Abre un archivo y escribe NOMBRE:'Mauricio', APELLIDO:'
Jaimes',RESULTADO:'100'
- ☐ No puede abrir el archivo, sintaxis incorrecta
- ☐ Este código produce Error
- ☒ Ninguna de las anteriores

Es ninguna es correcta porque en realidad estamos escribiendo (1) los encabezados y (2) los datos de la primera fila. En las opciones tenemos o los encabezados o los datos de la primera fila.

```
import csv
labels=['Book','Author','Year Released']
cuantas=int(input('cuantas datos quiere agregar'))
with open('libros.csv','a') as archivo:

    writer=csv.DictWriter(archivo,fieldnames=labels)
    writer.writeheader()
    for i in range (cuantas):
        nombre=input('Ingrese nombre del libro')
        autor=input('Ingrese el autor')
        fecha=input('Year Released ')
        writer.writerow({'Book':nombre,'Author':autor,'Year Released':fecha})
```

Marca solo un óvalo.

- ☐ Escribe correctamente la información de los nuevos libros
- ☐ No escribe la información de los nuevos libros pedidos
- ☐ Daña el formato del archivo porque escribe en el archivo información que no es un
nuevo libro
- ☐ No he mirado el tema
- ☐ Ninguna de las anteriores