

Tarea 10

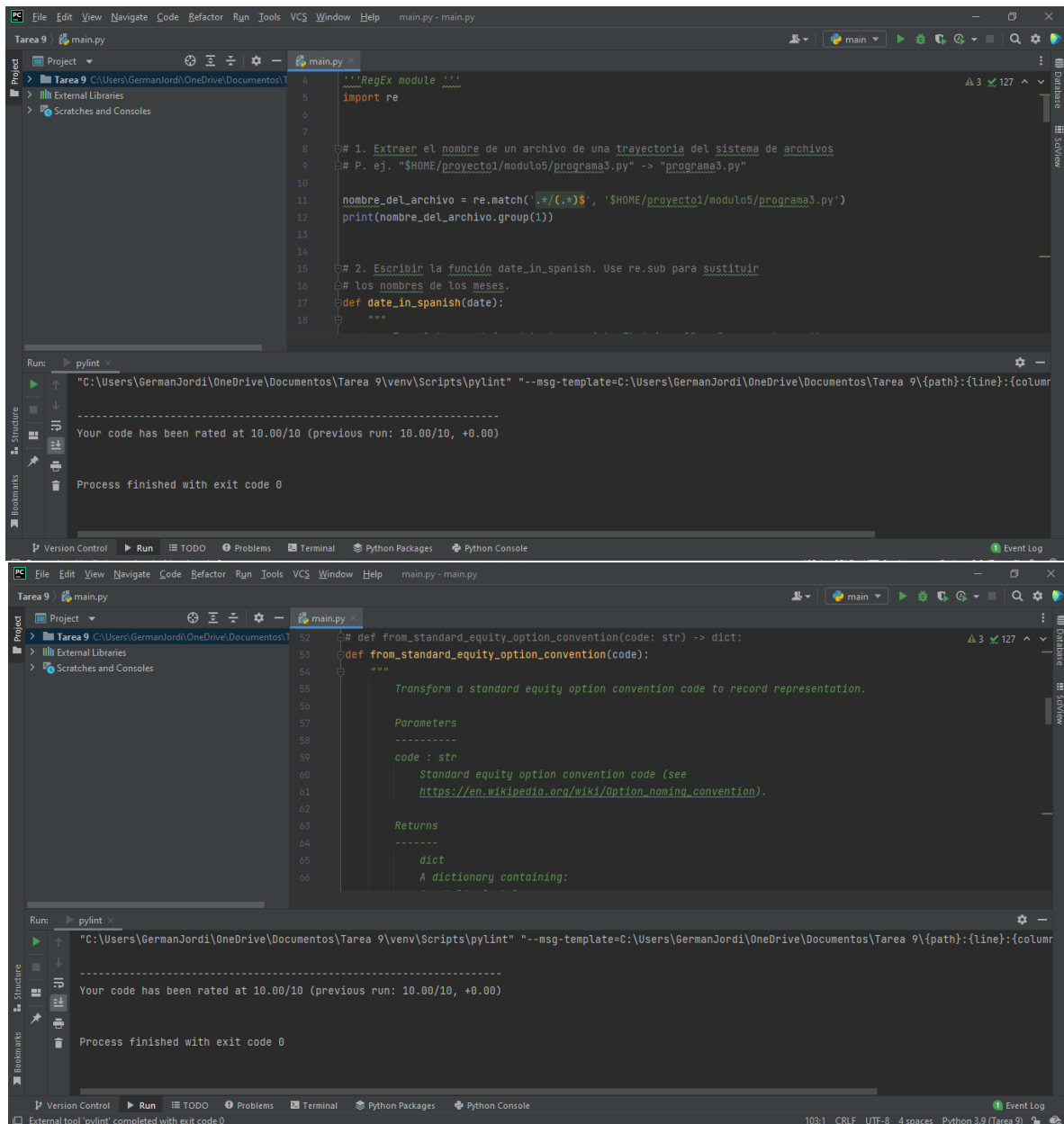
Nombre: German Jordi Arreortua Reyes.

Materia: Programación Avanzada

Fecha: martes, 24 de mayo de 2022.

1. Reporte generado por Pylint de la Tarea 9.

A continuación se muestra la captura de pantalla del reporte generado por Pylint de todas las actividades de la Tarea 9.



The image consists of two screenshots of a code editor (VS Code) showing Python code and the output of a Pylint command.

Top Screenshot: The editor shows a file named `main.py` with the following code:

```
'''RegEx module'''
import re

# 1. Extraer el nombre de un archivo de una trayectoria del sistema de archivos
# P. ej. "$HOME/proyecto1/modulo5/programa3.py" -> "programa3.py"

nombre_del_archivo = re.match('^(.*)$', '$HOME/proyecto1/modulo5/programa3.py')
print(nombre_del_archivo.group(1))

# 2. Escribir la función date_in_spanish. Use re.sub para sustituir
# los nombres de los meses.
def date_in_spanish(date):
    """
```

The Run and Debug console shows the output of the command:

```
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 9\venv\Scripts\pylint" --msg-template=C:\Users\GermanJordi\OneDrive\Documentos\Tarea 9\{path}:{line}:{column}
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
Process finished with exit code 0
```

Bottom Screenshot: The editor shows a different file named `main.py` with the following code:

```
def from_standard_equity_option_convention(code: str) -> dict:
def from_standard_equity_option_convention(code):
    """
    Transform a standard equity option convention code to record representation.

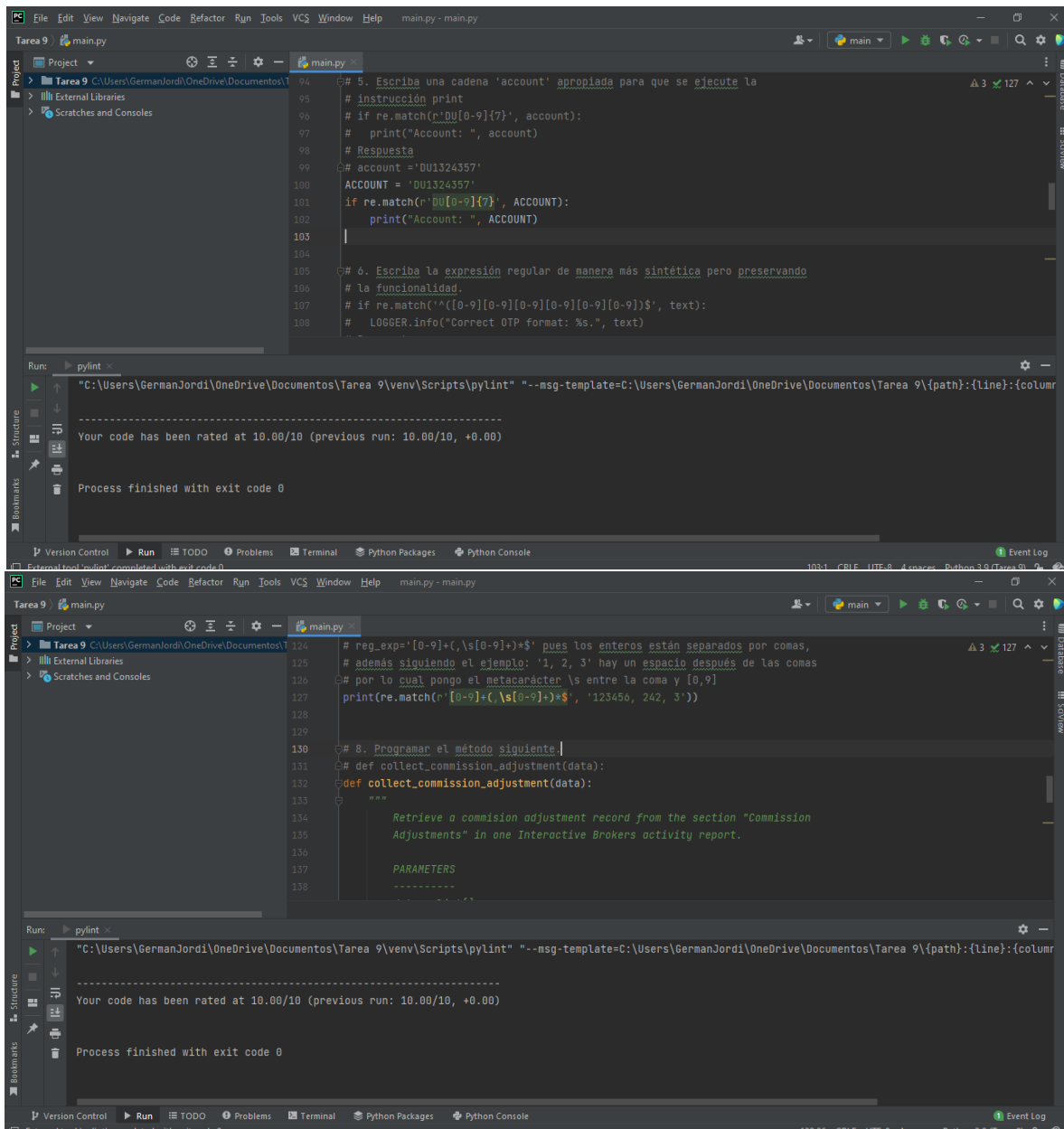
    Parameters
    -----
    code : str
        Standard equity option convention code (see
        https://en.wikipedia.org/wiki/Option_naming_convention).

    Returns
    -----
    dict
        A dictionary containing:
```

The Run and Debug console shows the output of the command:

```
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 9\venv\Scripts\pylint" --msg-template=C:\Users\GermanJordi\OneDrive\Documentos\Tarea 9\{path}:{line}:{column}
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
Process finished with exit code 0
```

At the bottom of the IDE, a status bar indicates: `External tool 'pylint' completed with exit code 0` and `1031 CRLF UTF-8 4 spaces Python 3.9 (Tarea 9)`.



2. Resumen de los métodos del tipo file de Python.

En Python podemos manipular un archivos, para ello debemos asignar una variable un valor de tipo file mediante la función open(), la cual recibe dos parámetros el nombre del archivo y modo en el cual abrirlo. Cada vez que abrimos un archivo estamos creando un puntero, el cual se posicionará dentro del archivo en un lugar determinado y este puntero podrá moverse dentro de ese archivo, eligiendo su nueva posición.

A continuación se muestran los métodos del tipo file en Python.

Métodos del tipo file de Python

- **close:** Cierra un archivo abierto

Sintaxis: `file.close()`

Ejemplo



```
Tarea_10.py
1
2 archivo = open('Prueba_Tarea_10.txt', 'r')
3 print(archivo.read())
4 archivo.close()
5 print(archivo.read())

Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Hola, mi nombre es German Jordi
y esto es una prueba
...
Traceback (most recent call last):
  File "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py", line 5, in <module>
    print(archivo.read())
ValueError: I/O operation on closed file.
```

Ejemplo

- **closed:** Devuelve un valor booleano que indica si el archivo está cerrado.

Sintaxis: `file.closed`

Ejemplo



```
Tarea_10.py
1 archivo = open("Prueba_Tarea_10.txt", "r")
2 print(archivo.closed)
3 archivo.close()
4 print(archivo.closed)
5

Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
False
True
Process finished with exit code 0
```

- **read:** Lee todo el contenido de un archivo. Si se le pasa la longitud de bytes, leerá solo el contenido hasta la longitud indicada.

Sintaxis: `file.read(size)`

```
Tarea_10.py
1 archivo = open('Prueba_Tarea_10.txt', 'r')
2 print('31 bytes: ', archivo.read(31))
3
4 archivo = open('Prueba_Tarea_10.txt', 'r')
5 print('completo: ', archivo.read())
6

Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
31 bytes: Hola, mi nombre es German Jordi
completo: Hola, mi nombre es German Jordi
y esto es una prueba
...
Process finished with exit code 0
```

- **encoding:** retorna la codificación de caracteres de un archivo de texto.

Sintaxis: `file.encoding`

Ejemplo

```
Tarea_10.py
1 archivo = open('Prueba_Tarea_10.txt', 'r')
2 print('lectura del archivo:', archivo.read())
3 print('al aplicar encoding:', archivo.encoding)

Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
lectura del archivo: Hola, mi nombre es German Jordi
y esto es una prueba
...
al aplicar encoding: cp1252
Process finished with exit code 0
```

- **errors:** Devuelve el controlador de errores Unicode utilizado junto con la codificación.

Sintaxis: `file.errors`

- **fileno:** Devuelve el descriptor de archivo de la secuencia, como un número. Se producirá un error si el sistema del operador no utiliza un descriptor de archivo.

Sintaxis: `file.fileno()`

Ejemplo

```
Tarea_10.py
1 archivo = open("Prueba_Tarea_10.txt", "r")
2 print(archivo.fileno())
3

Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
3
Process finished with exit code 0
```

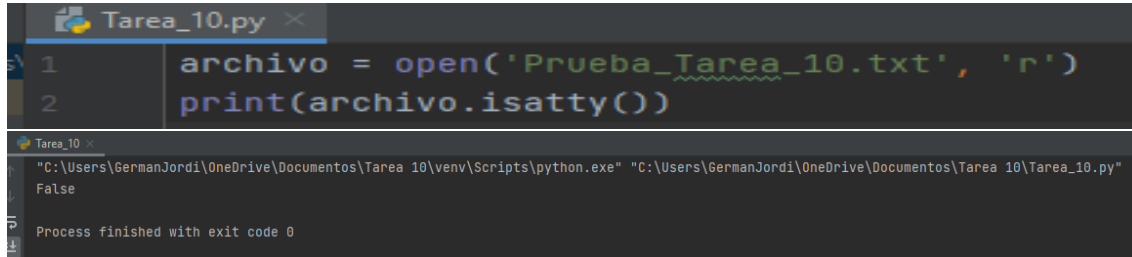
- **flush:** Limpia el búfer interno del archivo, esto ocurre automáticamente al cerrarlos, sin embargo, podemos limpiar el búfer interno del archivo antes de cerrarlo usando el método `flush()`.

Sintaxis: `file.flush()`

- **isatty:** Devuelve True si el flujo de archivos es interactivo es decir si está conectado a una terminal o un dispositivo tty, de lo contrario devuelve false.

Sintaxis: `file.isatty()`

Ejemplo



```
Tarea_10.py x
1 archivo = open('Prueba_Tarea_10.txt', 'r')
2 print(archivo.isatty())

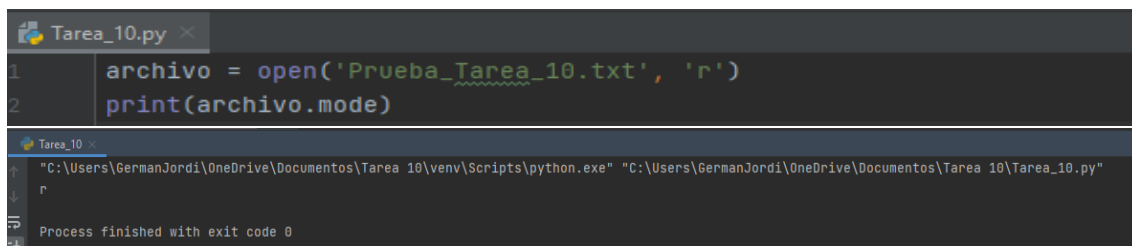
Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
False

Process finished with exit code 0
```

- **mode:** retorna el modo de apertura.

Sintaxis: `file.mode`

Ejemplo



```
Tarea_10.py x
1 archivo = open('Prueba_Tarea_10.txt', 'r')
2 print(archivo.mode)

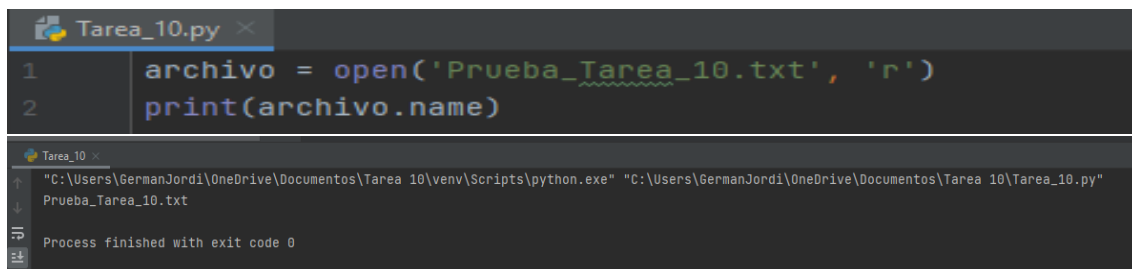
Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
r

Process finished with exit code 0
```

- **name:** retorna el nombre del archivo.

Sintaxis: `file.name`

Ejemplo



```
Tarea_10.py x
1 archivo = open('Prueba_Tarea_10.txt', 'r')
2 print(archivo.name)

Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Prueba_Tarea_10.txt

Process finished with exit code 0
```

- **newlines:** Devuelve el tipo de líneas nuevas encontradas al leer el archivo.

Sintaxis: `file.newlines`

- **next:** Para recorrer manualmente todos los renglones de un archivo.

Sintaxis: `file.__next__()`

Ejemplo

The screenshot shows a Python script named `Tarea_10.py` with the following code:

```
1 archivo = open('Prueba_Tarea_10.txt', 'r')
2 print('Renglon 1: ', archivo.__next__())
3 print('Renglon 2: ', archivo.__next__())
4 print('Renglon 3: ', archivo.__next__())
```

The output of the script is shown in the terminal window below:

```
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Renglon 1: Hola, mi nombre es German Jordi
Renglon 2: y esto es una prueba
Renglon 3: ...
Process finished with exit code 0
```

- **readline:** Devuelve una línea del archivo.

Sintaxis: `file.readline(size)`

Ejemplo

The screenshot shows a Python script named `Tarea_10.py` with the following code:

```
1
2 archivo = open('Prueba_Tarea_10.txt', 'r')
3 print(archivo.readlines())
4 print(archivo.readlines())
5
```

The output of the script is shown in the terminal window below:

```
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Hola, mi nombre es German Jordi
y esto es una prueba
...
Process finished with exit code 0
```

- **readlines:** Devuelve una lista conteniendo cada línea del archivo como un elemento de lista.

Sintaxis: `file.readlines(hint)`

El parámetro `hint`, es el número de bytes que limita el número de líneas devueltas, si el número total de bytes devueltos supera el número especificado, no se devuelven más líneas.

Ejemplo

The screenshot shows a Python script named `Tarea_10.py` with the following code:

```
archivo = open('Prueba_Tarea_10.txt', 'r')
print(archivo.readlines(31))
archivo = open('Prueba_Tarea_10.txt', 'r')
print(archivo.readlines(32))
archivo = open('Prueba_Tarea_10.txt', 'r')
print(archivo.readlines(50))
```

```
Tarea_10 x
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
['Hola, mi nombre es German Jordi\n']
['Hola, mi nombre es German Jordi\n', 'y esto es una prueba\n']
['Hola, mi nombre es German Jordi\n', 'y esto es una prueba\n']

Process finished with exit code 0
```

- **seek:** Mueve el puntero hacia el byte indicado.

Sintaxis: `file.seek(offset)`

offset es un número que representa la posición para establecer la posición actual del flujo de archivos.

Ejemplo

```
Tarea_10.py x
1
2 archivo = open('Prueba_Tarea_10.txt', 'r')
3 print(archivo.read())
4 print('imprimir nuevamente ', archivo.read())
5 archivo.seek(0)
6 print('imprimir despues de aplicar seek: ', archivo.read())
7

Tarea_10 x
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Hola, mi nombre es German Jordi
y esto es una prueba
...
imprimir nuevamente
imprimir despues de aplicar seek: Hola, mi nombre es German Jordi
y esto es una prueba
...

Process finished with exit code 0
```

- **softspace:** Devuelve un valor booleano que indica si se debe imprimir un carácter de espacio antes de otro valor cuando se usa la declaración de impresión, este método se eliminó de Python 3.
- **tell:** Devuelve la posición actual de la transmisión.

Sintaxis: `file.tell()`

Ejemplo

```
Tarea_10.py x
1
2 archivo = open('Prueba_Tarea_10.txt', 'r')
3 print(archivo.readline())
4 print(archivo.tell())
5

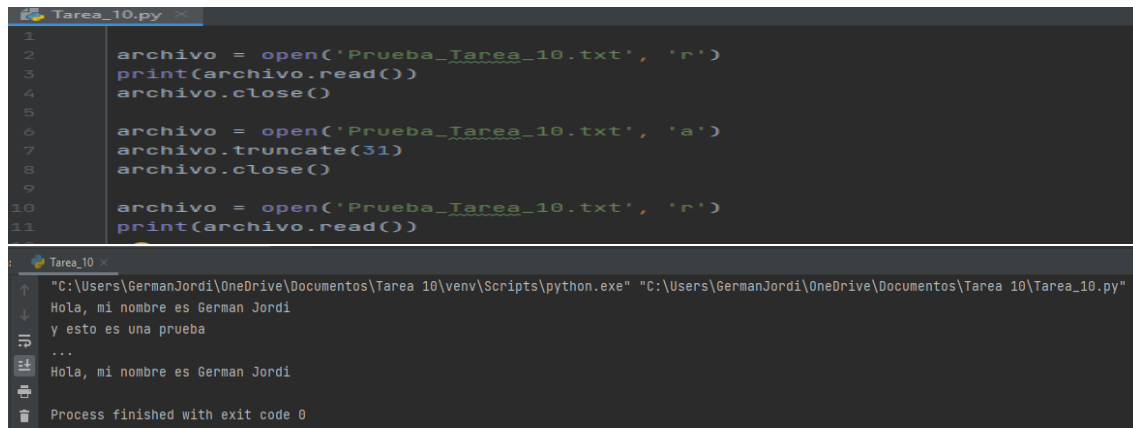
Tarea_10 x
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Hola, mi nombre es German Jordi
33

Process finished with exit code 0
```

- **truncate:** Trunca el archivo al tamaño de bytes.

Sintaxis: `file.truncate(tamaño)`

Ejemplo



```
Tarea_10.py
1
2  archivo = open('Prueba_Tarea_10.txt', 'r')
3  print(archivo.read())
4  archivo.close()
5
6  archivo = open('Prueba_Tarea_10.txt', 'a')
7  archivo.truncate(31)
8  archivo.close()
9
10 archivo = open('Prueba_Tarea_10.txt', 'r')
11 print(archivo.read())

Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Hola, mi nombre es German Jordi
y esto es una prueba
...
Hola, mi nombre es German Jordi
Process finished with exit code 0
```

- **write:** Escribe un texto especificado en el archivo.

El lugar donde se insertará el texto especificado depende del modo de archivo y la posición de la secuencia.

Sintaxis: `file.write(text)`

Ejemplo



```
Tarea_10.py
1
2  archivo = open('Prueba_Tarea_10.txt', 'a')
3  archivo.write('\n nueva línea')
4  archivo.close()
5  archivo = open('Prueba_Tarea_10.txt', 'r')
6  print(archivo.read())
7

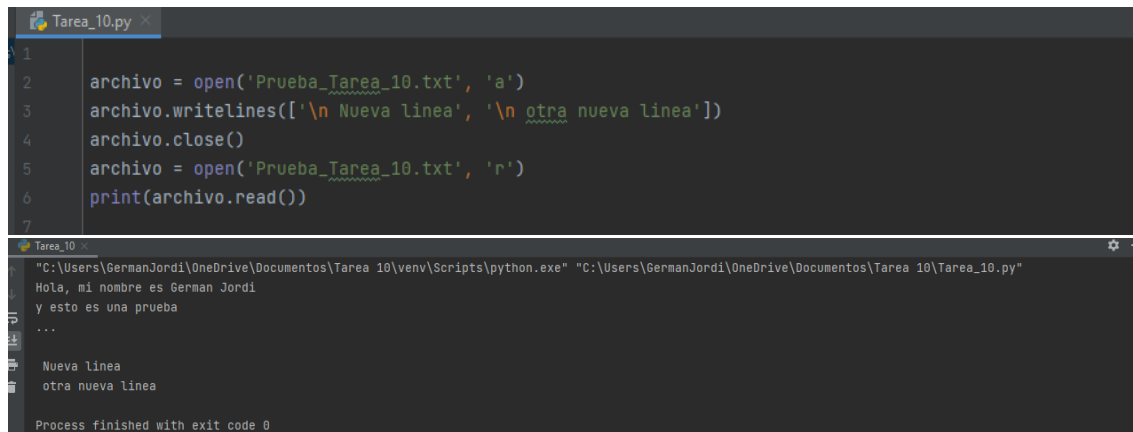
Tarea_10
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Hola, mi nombre es German Jordi
y esto es una prueba
...
nueva línea
Process finished with exit code 0
```

- **writelines:** Escribe los elementos de una lista en el archivo.

El lugar donde se insertarán los textos depende del modo de archivo y la posición del puntero.

Sintaxis: `file.writelines(list)`

Ejemplo



The image shows a screenshot of a Python IDE with two windows. The top window, titled 'Tarea_10.py', contains the following Python code:

```
1
2  archivo = open('Prueba_Tarea_10.txt', 'a')
3  archivo.writelines(['\n Nueva línea', '\n otra nueva línea'])
4  archivo.close()
5  archivo = open('Prueba_Tarea_10.txt', 'r')
6  print(archivo.read())
7
```

The bottom window, titled 'Tarea_10', shows the command prompt output of running the script. It displays the file path, the script's execution, and the contents of the file 'Prueba_Tarea_10.txt' as read by the program.

```
"C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\venv\Scripts\python.exe" "C:\Users\GermanJordi\OneDrive\Documentos\Tarea 10\Tarea_10.py"
Hola, mi nombre es German Jordi
y esto es una prueba
...
Nueva línea
otra nueva línea

Process finished with exit code 0
```

- **xreadlines:** Devuelve un iterador que lee una línea del archivo a la vez, esto se eliminó en Python 3, en lugar de eso podemos usar el método `next` con el archivo.