



Programación Avanzada 2025

Lab. 1.2. Lectura y Escritura en Archivos de Texto en Python

Agosto 22, 2025

Cree una carpeta (folder) en el disco D, nómbrela con su apellido paterno seguido de su código. Ejemplo: LOPEZ12345

Tabla 1.1 Operaciones de Secuencias aplicables a String

| Nombre | Operación | Significado para S="¡Hola Como Estas!" |
|---------------|----------------|--|
| Length | len(str) | len(S) ---> 17 |
| Select | S[i] | S[6] ---> 'C' |
| Slice | S[i:j] | S[11:16]---> "Estas"(desde i hasta j-1) |
| | S[i:j:k] | S[1:12:5]---> "HCE" |
| Count | S.count(chr) | S.count('o') ---> 3 |
| Index | S.index(chr) | S.index('E') ---> 11 |
| Membership | 'chr' in S | 'p' in S ---> False |
| | 'chr' not in S | 'p' not in S ---> True |
| Concatenation | S + W | S + "!!" ---> "¡Hola Como Estas!!!" |
| | n*S (S*n) | 2*S-->"¡Hola Como Estas!¡Hola Como Estas!" |
| Minimum Value | min(S) | min(S) ---> " " (espacio) |
| Maximum Value | max(S) | max(S) ---> 't' |
| Comparison | S == W | S == "Hola" ---> False |

Tabla 1.2 Principales Métodos de String

Métodos para verificar el contenido de un String

| | | | |
|--------------------------------|---|------------|----------------------|
| str.isalpha() | Devuelve True si str contiene solo letras | S="Hello" | S.isalpha()--> True |
| | | S="Hello!" | S.isalpha()--> False |
| str.isdigit() | Devuelve True si str contiene solo dígitos | S="789" | S.isdigit()--> True |
| | | S="789W" | S.isdigit()--> False |
| str.islower() str.isupper() | Devuelve True si str contiene solo letras minúsculas (mayúsculas) | S="hello" | S.islower()--> True |
| | | S="Hello" | S.isupper()--> False |
| str.lower() str.upper() | Devuelve la versión minúsculas (mayúsculas) de str | S="Hello!" | S.lower()-->"hello!" |
| | | S="hello!" | S.upper()-->"HELLO!" |

Método de búsqueda en un String

| | | | |
|-------------|---|-------------|-------------------|
| str.find(w) | Retorna el índice de la primera ocurrencia de w en str. Retorna -1 si no lo encuentra | S="Hello!" | S.find('l')--> 2 |
| | | S="Goodbye" | S.find('l')--> -1 |

Método para reemplazar el contenido de un String

| | | | |
|-------------------|--|------------|--------------------------------|
| str.replace(w, t) | Retorna una copia de str con todas las ocurrencias de w reemplazadas con t | S="Hello!" | S.replace('H', 'J')--> "Jello" |
| | | S="Hello" | S.replace('ll', 'r')--> "Hero" |

Método para remover el contenido de un String

| | | | |
|--------------|--|--------------|---------------------------|
| str.strip(w) | Retorna una copia de str con todos los caracteres iniciales y finales que aparecen en w eliminados | S=" Hello! " | S.strip(' !') --> "Hello" |
| | | S="Hello\n" | S.strip('\n') --> "Hello" |

Método para dividir el contenido de un String

| | | | |
|--------------|--|-------------|---------------------------------|
| str.split(w) | Devuelve una lista que contiene todas las string de str delimitado por w | S="Lu,Chao" | S.split(',') --> ["Lu", "Chao"] |
|--------------|--|-------------|---------------------------------|

Uso de Archivos de Texto

Existen dos tipos de archivos, *archivo de texto* y *archivo binario*. Los archivos de texto se procesan línea por línea. Los archivos binarios se procesan byte por byte. Nosotros trabajaremos principalmente con archivos de texto.

Las operaciones fundamentales de todos los tipos de archivos incluyen *apertura* de un archivo, *lectura* desde un archivo, *escritura* en un archivo, y *cierre* de un archivo.

Apertura de Archivos de Texto

Todo archivo debe de ser abierto antes de ser usados. En Python, cuando se abre un archivo, se crea un *objeto archivo* el cual provee métodos para acceder al archivo.

Antes de *leer desde un archivo* o antes de *escribir en un archivo*, primero se debe *abrir el archivo*.

Para abrir un archivo en Python se usa la función `open()`, esta función crea un *objeto archivo* y lo asocia con un archivo en el disco. Asimismo, `open()` crea un canal de comunicación entre un programa en la RAM y un archivo en el disco.

El formato general de `open()` es como sigue:

```
file_variable = open(filename, mode)
```

`file_variable` es el nombre de la variable que referencia al objeto archivo

`filename` es un string que especifica el nombre del archivo, el nombre incluye la ruta de ubicación del archivo en el disco.

`mode` es un string que especifica el modo en que el archivo será abierto (lectura, escritura, etc.)

Una vez realizada la apertura de un archivo, se puede realizar distintos procesamiento al archivo. Después de realizar los procesamiento al archivo, se debe de tener cuidado en *cerrar el archivo* apropiadamente, de lo contrario podrían presentarse inconvenientes indeseados. Se cierra un archivo con la función `close()` de la siguiente manera:

```
file_variable.close()
```

Python provee mecanismos para asegurarnos de cerrar un archivo.

Apertura para lectura

Para abrir un archivo para lectura, se usa la función incorporada `open` como sigue:

```
input_file = open('myfile.txt', 'r')
```

El primer argumento es el nombre del archivo, `'myfile.txt'`, que se encuentra en el directorio actual, también podemos usar la dirección absoluta del archivo. El segundo argumento, `'r'`, indica que el archivo será abierto para lectura. (*El segundo argumento es opcional cuando se abre un archivo para lectura*)

Como mencionamos anteriormente, una vez abierto el archivo, después de procesarlo, debemos cerrar el archivo usando `close()`, sin embargo, es responsabilidad del programador de usar apropiadamente la función `close()` y no olvidar colocarlo en el programa.

A fin de asegurarnos de cerrar el archivo, sin colocar explícitamente `close()`, lo recomendable es usar la sentencia `with as` de la siguiente manera:

```
with open("myfile.txt", "r") as input_file:
    # Aquí va el código para procesar el archivo
```

La sentencia `with` automáticamente se encarga de cerrar el archivo después de salir del bloque `with`.

Apertura para Escritura

```
output_file = open("mynewfile.txt", "w")
```

Si el segundo argumento es 'a', abrirá el archivo para agregar los datos al final del archivo si existe.

Usando `with as`:

```
with open("mynewfile.txt", "w") as output_file:  
    # Aquí va el código para guardar los datos en el archivo
```

Leyendo Archivos de Texto línea por línea con `readline()`

La lectura de un archivo de texto se realiza con la ayuda del método `readline`, el cual devuelve en forma de un string la siguiente línea de un archivo de texto, incluyendo el carácter de fin-de-línea, `\n`. Cuando llega al final del archivo, devuelve un string vacío.

| Datos del Archivo: myfile.txt | Código | Salida |
|--|---|--|
| Línea Uno\n Línea Dos\n Línea Tres\n | <pre>input_file = open('myfile.txt','r') linea=input_file.readline() while linea != '': print(linea) linea=input_file.readline() input_file.close()</pre> | Línea Uno Línea Dos Línea Tres |

Usando `with as`:

| Datos del Archivo: myfile.txt | Código | Salida |
|--|---|--|
| Línea Uno\n Línea Dos\n Línea Tres\n | <pre>with open("myfile.txt","r") as input_file: linea=input_file.readline() while linea != '': print(linea) linea=input_file.readline()</pre> | Línea Uno Línea Dos Línea Tres |

Leyendo todo el contenido del archivo con `readlines()` y `for`

Recordar que `readlines()` lee todo el contenido del archivo y lo guarda en una lista, en la que cada elemento es una línea del archivo en forma de string.

```
with open("myfile.txt","r") as input_file:  
    todoLista = input_file.readlines()  
    for linea in todoLista:  
        print(linea)
```

El código anterior se puede simplificar iterando sobre el mismo objeto archivo:

```
with open("myfile.txt","r") as input_file:
    for linea in input_file:
        print(linea)
```

Leyendo todo el contenido del archivo con `read()`

La función `read()` sin parámetros, lee todo el contenido del archivo y lo guarda en forma de un solo string.

```
with open("myfile.txt","r") as input_file:
    # Leyendo todo el contenido del archivo (string)
    todo = input_file.read()
    print(todo)
```

Leyendo Archivos de Texto línea por línea con `read()` y `for`

```
with open("myfile.txt", "r") as input_file:
    # Leyendo todo el contenido del archivo (string)
    todo = input_file.read()
    todoLista = todo.split("\n")
    print(todoLista)
    # Imprimiendo línea por línea
    for linea in todoLista:
        print(f"Linea:{linea}")
```

Escritura en un Archivo de Texto

Para escribir string en un archivo se usa el método `write`.

| Archivo: myfile.txt | Código | Archivo: myfile_copy.txt |
|------------------------------------|---|--------------------------------------|
| Línea Uno\nLínea Dos\nLínea Tres\n | <pre>input_file = open('myfile.txt','r') output_file = open('myfile_copy.txt','w') linea=input_file.readline() while linea != '': print(linea) output_file.write(linea) linea=input_file.readline() output_file.close()</pre> | Línea Uno Línea Dos Línea Tres |

Usando `with` `as` y `write()`:

```
with open("myfile.txt","r") as input_file:
    todoLista = input_file.readlines()

with open("myfile_copy.txt","w") as output_file:
    # Writing multiple lines to the file
    for linea in todoLista:
        output_file.write(linea)
```

Usando `with` `as` y `writelines()`:

```
with open("myfile_copy2.txt","w") as output_file:
    # Writing multiple lines to the file
    Output_file.writelines(todoLista)
```

Para los siguientes ejercicios, elabore un programa en Python. Verifique los programas ejecutando y probando para distintas entradas.

1. El archivo `datos_estudiante.txt` contiene, en cada línea, información de un estudiante de un curso determinado. El nombre del estudiante es la primera información en cada línea, seguida por sus notas en el curso.
Usando el archivo `datos_estudiante.txt` escriba un programa y realice lo siguiente:
 - a) Imprimir los nombres de los estudiantes que tengan más de seis notas en el curso.
 - b) Agregue el código para que imprima el nombre del archivo
 - c) Agregue el código para que imprima el contenido total del archivo (lista de alumnos)
2. Usando el archivo `datos_estudiante.txt`, escriba un programa que determine el promedio de cada estudiante e imprima el nombre del estudiante y su promedio.
3. Escriba un programa que elimine, de un archivo de texto, todas las ocurrencias de un string especificado. Vuestro programa debe solicitar que ingrese el nombre del archivo y el string a ser eliminado.
 - a) Use el archivo `alumnos2025.txt` (mantener dos copias del archivo en el disco).
 - b) Muestre el contenido inicial del archivo
 - c) Genere un nuevo archivo `alumnos2025_new.txt`, a partir de `alumnos2025.txt`, eliminando el string especificado. Verifique físicamente el contenido del archivo
 - d) Modifique el archivo `alumnos2025.txt`, eliminando el string especificado. Muestre el contenido del archivo actualizado.
4. Escriba un programa que guarde en un archivo la siguiente tabla que muestra la conversión de kilogramos a libras y de libras a kilogramos. Considere 1 kilogramo igual a 2.2 libras. El programa debe guardar la información en forma de una tabla como se muestra en el ejemplo.

| Kilogramos | Libras | Libras | Kilogramos |
|------------|--------|--------|------------|
| 1 | 2.2 | 20 | 9.09 |
| 3 | 6.6 | 25 | 11.36 |
| 5 | 11.0 | 30 | 13.64 |
| ... | ... | ... | ... |
| 197 | 433.4 | 510 | 231.82 |
| 199 | 437.8 | 515 | 234.09 |

- a) Muestre el contenido del archivo en pantalla.
- b) Determine cuántas líneas tiene la tabla
- c) Solicite al usuario que ingrese un número n entre 1 y 100 ambos inclusive, muestre la línea n del archivo.

OPCIONAL

5. Modifique el programa del Ejercicio N° 2 a fin de que el programa muestre también la mayor nota del curso y el alumno que lo obtuvo.
6. Modifique el programa del Ejercicio N° 2 a fin de que permita ingresar del teclado los datos de un nuevo alumno y lo agregue al final del archivo `datos_estudiante.txt`.
7. Modifique el programa del Ejercicio N° 3 a fin de eliminar la línea vacía que se generó al eliminar el string especificado en el archivo inicial.

Guarde todos vuestros programas en una carpeta con el nombre su **Apellido** paterno seguido de vuestro **DNI**, luego comprima esta carpeta. Envíe este archivo a:
Katherine Navarro katherine.navarro@upch.pe especificando como asunto **Lab1.2**.