

Previa Clase5

April 9, 2021

0.0.1 Seminario de Lenguajes - Python

0.1 Cursada 2021

0.1.1 Previa Clase 5

1 Manejo de archivos

- Usamos archivos cuando necesitamos una estructura de datos que permita que los mismos puedan **persistir** cuando la ejecución del programa finalice.
- En Python los archivos se manejan como objetos que se crean usando la [función open](#).

2 Algunos ejemplos

```
[ ]: archi1 = open('archivo.txt', 'w')
```

```
[ ]: archi2 = open('archivo.txt', 'x')
```

```
[ ]: archi3 = open('archivo.txt', 'r')
```

3 ¿Cómo guardamos datos en un archivo?

- El caso más sencillo: guardando texto en un archivo.

```
[ ]: f = open('archivo.txt', 'w')  
f.write('Hola, ')  
f.write('Mundo!')  
f.close()
```

- **write(cadena)**: escribe *cadena* en el archivo y retorna cantidad de caracteres escritos.
- **close()**: cierra el archivo.

4 ¿Cómo leemos los datos guardados?

```
[ ]: f = open('archivo.txt', 'r')  
print(f.read(4))  
print(f.read())
```

- **read(cantidad_bytes):** lee *cantidad_bytes* del archivo.
- Si cantidad_bytes es <0 o no está, lee hasta fin de archivo.
- Retorna "" si EOF.
- **Tarea:** probar el siguiente ejemplo que muestran otras formas de leer caracteres desde un archivo de texto.

```
[ ]: def leo_caracteres():
    f = open("imagine.txt", "r")
    for x in f.read():
        print(x)
    f.close()

def leo_lineas():
    f = open("imagine.txt", "r")
    print(f.readlines())
    f.close()

def otra_forma():
    f = open("imagine.txt", "r")
    for linea in f:
        print(linea)
    f.close()

def main():
    print('Leo caracteres')
    leo_caracteres()
    print('-' * 20)
    print('Leo lineas')
    leo_lineas()
    print('-' * 20)
    print('Otra forma')
    otra_forma()

if __name__ == "__main__":
    otra_forma()
```

5 ¿Qué pasa si necesito guardar información que tiene una estructura?

- Pensemos en estos ejemplos:
 - Los puntajes cada vez que juego a un juego. Información tipo: nombre jugador, puntaje, fecha.
 - El banco de preguntas: tema, enunciado, respuesta correcta.
 - Los Python Plus de los estudiantes por turnos: turno, nombre, apellido, num_alumno, cantidad_puntos, etc.

- En estos casos también podría usar un archivo de texto: ¿cómo se les ocurre?

6 Algunas posibilidades

```
'equipo: Astralis - e-sport: CSGO - pais: Dinamarca'
```

```
---
```

```
equipo:Astralis
```

```
e-sport:CSGO
```

```
pais:Dinamarca
```

```
---
```

```
'Astralis-CSGO-Dinamarca'
```

```
'Astralis*CSGO*Dinamarca*'
```

- ¿Pros y contras?

7 Hay otras formas mejores...

8 JSON (JavaScript Object Notation)

- Es un formato de intercambio de datos muy popular. Por ejemplo:

```
{"equipo": "Astralis",
  "e-sport": "CSGO",
  "pais": "Dinamarca"}
```

```
o
```

```
[{"equipo": "Astralis",
  "e-sport": "CSGO",
  "pais": "Dinamarca"},
 {"equipo": "Nz",
  "e-sport": "CSGO",
  "pais": "Argentina"}]
```

- [+Info](#)
- Veamos este ejemplo: https://developers.mercadolibre.com.ar/es_ar/categorias-y-publicaciones#close
- ¿Conocen la [central meteorológica](#) de la facultad? Miremos estos [datos](#)

9 Módulo json

- Python tiene un módulo que permite tabajar con este formato.
- Para usarlo, debemos importarlo.

```
[ ]: import json
```

- Permite serializar objetos.

- serializamos con: **dumps()** y **dump()**.
- deserializamos con: **loads()** y **load()**.
- Más info en: <https://docs.python.org/3/library/json.html>

9.0.1 Veamos este ejemplo

- Generamos un archivo con bandas de distintas ciudades:
 - Tenemos: nombre de la banda, ciudad en la que se generó y una referencia a su trabajo.
 - Empecemos por La Plata...

```
[ ]: import json

archivo = open("bandas.txt", "w")
datos = [
    {"nombre": "William Campbell", "ciudad": "La Plata", "ref": "www.instagram.
    ↪com/williamcampbellok"},
    {"nombre": "Buendia", "ciudad": "La Plata", "ref": "https://buendia.bandcamp.
    ↪com/"},
    {"nombre": "Lúmine", "ciudad": "La Plata", "ref": "https://www.instagram.
    ↪com/luminelp/"}]
json.dump(datos, archivo)
archivo.close()
```

- ¿De qué tipo es la variable datos?

```
[ ]: # Ahora accedemos a los datos guardados
import json

archivo = open("bandas.txt", "r")
datos = json.load(archivo)

datos_a_mostrar = json.dumps(datos, indent=4)
print(datos_a_mostrar)
archivo.close()
```

- ¿De qué tipo de datos? ¿Y datos_a_mostrar?

9.1 ¿Qué tipo de archivos manejamos hasta ahora?

9.2 Hasta ahora todos los archivos de nuestros ejemplos son archivos de texto.

10 El módulo pickle

- Es un protocolo **binario** para serializar objetos.
- El formato de datos que utiliza pickle es **específico de Python**.
- Se usan las funciones **load()** y **dump()**.
- También existen **loads()** y **dumps()**.
- Más info en: <https://docs.python.org/3/library/pickle.html>
- Veamos el ejemplo anterior, pero con este formato:

```
[ ]: import pickle

archivo = open("bandas.dat", "wb")
datos = [
    {"nombre": "William Campbell", "ciudad": "La Plata", "ref": "https://www.
    ↪instagram.com/williamcampbellok"},
    {"nombre": "Buendia", "ciudad": "La Plata", "ref": "https://buendia.bandcamp.
    ↪com/"},
    {"nombre": "Lúmine", "ciudad": "La Plata", "ref": "https://www.instagram.
    ↪com/luminelp/"}]

pickle.dump(datos, archivo)
archivo.close()
```

- Accedemos a los datos guardados:

```
[ ]: import pickle

archivo = open("bandas.dat", "rb")
datos = pickle.load(archivo)

for item in datos:
    for item, valor in item.items():
        print(f"{item}: {valor}")
    print("-"*30)
archivo.close()
```

- ¿De qué tipo de datos?
- ¿De qué forma estamos abriendo el archivo?

11 json vs. pickle

- ¿Qué consideraciones deberíamos analizar en cada caso?
- ¿Ventajas y desventajas?
- Si queremos interactuar con aplicaciones externas, ¿qué formato les parece mejor?
- Ambos nos permiten serializar-deserializar objetos.
- **pickle:**
 - Formato específico de Python. ¿Qué problema podría traer esto? ¿Qué ventaja?
 - Tiene una [advertencia de seguridad](#).
 - Genera archivos binarios.
- **json:**
 - Notación muy popular.
 - No dependemos de Python.
 - Genera archivos de texto.

12 Pensemos en las siguientes situaciones:

¿Qué hacemos si queremos:

- agregar nuevos datos a los datos ya existentes?;
- modificar el puntaje de un jugador específico en un juego?;
- agregar una nueva pregunta al banco de preguntas del juego de repaso?;
- incrementar los Python Plus de un estudiante determinado?

¿Qué tienen todas estas situaciones en común?

13 Acceso aleatorio

- Si el valor **desde_donde** no está, se asume 0.
- Si el archivo es de texto, solamente se considera desplazar desde el comienzo del archivo, es decir el valor **desde_donde** es 0.

14 Veamos el siguiente ejemplo

```
[ ]: archivo = open("ejemplos/clase5/imagen.txt", "rb")

archivo.seek(-18, 2)      #Esto da error si el archivo se abre en modo "r"
print(archivo.read().decode('UTF-8'))
print("*"*30)
archivo.seek(0, 0)
print(archivo.read().decode('UTF-8'))
archivo.close()
```

```
[ ]: archivo = open("ejemplos/clase5/imagen.txt", "rb+")

archivo.seek(0, 2)      #Esto da error si el archivo se abre en modo "r"
archivo.write(bytes("John Lennon", 'UTF-8'))
archivo.close()
```

- ¿Por qué convierto a bytes?

```
[ ]: archivo = open("ejemplos/clase5/imagen.txt", "rb")
print(archivo.read().decode('UTF-8'))
archivo.close()
```

- Probemos esto mismo abriendo trabajando con archivos de texto.
- ¿`archi.read(4).decode('UTF-8')`?

15 ¿Dónde estamos?

- `tell()`: retorna la posición actual.

```
[ ]: archivo = open("ejemplos/clase5/imagen.txt", "rb")
archivo.seek(0,2)
print(archivo.tell())
archivo.close()
```

16 Entrada y salida estándar

- Tres archivos estándares: **stdin**, **stdout**, **stderr**.
- Usados por el intérprete como entrada, salida y error estándar.
- Se los accede a través del módulo **sys**.

```
[ ]: import sys

sys.stdout.write("Hola \n\n que tal")
```

```
[ ]: nombre = input()
```

```
[ ]: import sys

nombre = sys.stdin.readline()
nombre
```

17 Retomamos los módulos estándares

17.1 Módulo OS

- Incluye funciones para:
 - Procesamiento de archivos.
 - Directorios.
 - Permisos.
- Más info: <http://docs.python.org/library/os.html#module-os>

```
[ ]: import os
#os.rename('ejemplos/clase5/imagen.txt', 'ejemplos/clase5/COPIA.TXT')
#os.remove('bandas.dat')
```

Veamos estos otros ejemplos

```
[ ]: lista = os.listdir("/home/claudia/git/")
print(lista)
print(os.sep)
```

```
[ ]: import os
print(os.access("/home/claudia/git/", os.W_OK))
```

18 Módulo os.path

- Provee algunas funciones útiles: exists(), isdir(), isfile()

```
[ ]: import os.path

print(os.path.exists("/home/claudia/git/"))
print(os.path.isdir("/home/clau/git/"))
```