

Nombre: Wilberth Alejandro Mbo Estrella
Seguridad de Datos

Grupo: IDYGS73

Materia:

INVESTIGACION MANIPULACION DE ARCHIVOS PDF, WORD Y EXCEL EN PYTHON

PDF

Python ofrece varias bibliotecas que facilitan la manipulación de archivos en formatos comunes como PDF, Word y Excel. A continuación, se proporciona una breve introducción a cómo trabajar con estos formatos: Abrir y Manipular PDF en Python Biblioteca: PyPDF2 Para abrir y manipular archivos PDF en Python, puedes utilizar la biblioteca PyPDF2. Con PyPDF2, puedes realizar tareas como leer texto, extraer páginas y combinar varios PDF.

Abrimos el documento PDF en modo lectura binario usando `open('document_path.PDF', 'rb')`. `PDFFileReader()` se utiliza para crear un objeto lector de PDF para leer el documento. Podemos extraer texto de las páginas del documento PDF usando los métodos `getPage()` y `extractText()`. Para obtener el número de páginas en el documento PDF dado, usamos `.numPages`.

WORD

Es necesario instalar el paquete `python-docx`, aunque existe un paquete que se llama `docx` este último es una versión legacy de `python-docx`.

```
from docx import Document
```

```
# Especifica la ruta del archivo .docx que deseas abrir
```

```
archivo_docx = "mi_archivo.docx"
```

```
# Abre el archivo .docx
```

```
documento = Document(archivo_docx)
```

Nombre: Wilberth Alejandro Mbo Estrella
Seguridad de Datos

Grupo: IDYGS73

Materia:

```
# Lee y muestra el contenido del archivo  
for paragraph in documento.paragraphs:  
    print(paragraph.text)
```

EXCEL

Biblioteca: pandas

Para trabajar con archivos Excel, pandas es una biblioteca muy útil. Pandas permite leer, escribir y realizar operaciones en hojas de cálculo. Aquí tienes un ejemplo de cómo abrir un archivo Excel:

```
import pandas as pd
```

```
# Leer un archivo Excel
```

```
df = pd.read_excel('archivo.xlsx')
```

```
# Realizar operaciones en el DataFrame (por ejemplo, filtrar datos)
```

```
filtered_data = df[df['Columna'] > 10]
```

```
# Guardar el resultado en un nuevo archivo Excel
```

```
filtered_data.to_excel('nuevo_archivo.xlsx', index=False)
```