



Instituto Politecnico Nacional  
Escuela Superior de Computo



Ing. Sistemas Computaciones

"Obtención de Información semi-pasiva"

Materia

GOBIERNO DE TI | COMPUTER SECURITY

Grupo

6CV2

Maestro

Aldama Coahuila Mario Alberto

Autor: Aldo Alcántara Martínez

Boleta 2019630578

Fecha: 22/03/2024

Reporte de los pasos que realice:

1.- Lo primero es designar una carpeta para trabajar nuestro script y no mezclarlo, en mi caso yo la llame All\_Meta

Nombre	Fecha de modificación	Tamaño	Clase
All_Meta	hoy, 12:56 a.m.	--	Carpeta
GetImgMetadata	hoy, 12:00 a.m.	--	Carpeta

2.- Despues, abro mi terminal y me posiciono en la carpeta que acabo de crear, luego procedo a crear una carpeta venv y así como se ve en la siguiente linea activo el modo venv, que ya en la tercer linea se ve como se activa en la parte izquierda, por ultimo reviso mi php List y compruebo que es lo que tengo instalado.

**source venv/bin/activate:** Esta instrucción activa el entorno virtual creado en el paso anterior. Una vez activado, cualquier instalación de paquetes de Python se realizará en el entorno virtual y no afectará al sistema global de Python.

```
aldoalcantara@MacBook-Pro-de-Aldo-2 All_Meta % python3 -m venv venv
aldoalcantara@MacBook-Pro-de-Aldo-2 All_Meta % source venv/bin/activate
(venv) aldoalcantara@MacBook-Pro-de-Aldo-2 All_Meta % pip list
Package Version
-----
pip      24.0
```

3.- Ya que comprobéste que tengo instalado, ahora instalo las bibliotecas que voy a ocupar, que son las que muestro en la imagen, junto con su proceso de descarga y por ultimo compruebo que se instalaron.

**pip install python-docx openpyxl PyPDF2:** Esta instrucción instala los paquetes necesarios para el script. python-docx se utiliza para trabajar con documentos DOCX, openpyxl para trabajar con archivos XLSX y PyPDF2 para trabajar con archivos PDF

```
(venv) aldoalcantara@MacBook-Pro-de-Aldo-2 All_Meta % pip install python-docx openpyxl PyPDF2
Collecting python-docx
  Downloading python_docx-1.1.0-py3-none-any.whl.metadata (2.0 kB)
Collecting openpyxl
  Downloading openpyxl-3.1.2-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Collecting lxml>=3.1.0 (from python-docx)
  Downloading lxml-5.1.0-cp312-cp312-macosx_10_9_x86_64.whl.metadata (3.5 kB)
Collecting typing-extensions (from python-docx)
  Downloading typing_extensions-4.10.0-py3-none-any.whl.metadata (3.0 kB)
Collecting et-xmlfile (from openpyxl)
  Downloading et_xmlfile-1.1.0-py3-none-any.whl.metadata (1.8 kB)
Downloading python_docx-1.1.0-py3-none-any.whl (239 kB)
 239.6/239.6 kB 2.1 MB/s eta 0:00:00
Downloading openpyxl-3.1.2-py2.py3-none-any.whl (249 kB)
 250.0/250.0 kB 6.1 MB/s eta 0:00:00
Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
 232.6/232.6 kB 7.4 MB/s eta 0:00:00
Downloading lxml-5.1.0-cp312-cp312-macosx_10_9_x86_64.whl (4.8 MB)
 4.8/4.8 MB 24.5 MB/s eta 0:00:00
Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
Downloading typing_extensions-4.10.0-py3-none-any.whl (33 kB)
Installing collected packages: typing-extensions, PyPDF2, lxml, et-xmlfile, python-docx, openpyxl
Successfully installed PyPDF2-3.0.1 et-xmlfile-1.1.0 lxml-5.1.0 openpyxl-3.1.2 python-docx-1.1.0 typing-extensions-4.10.0
(venv) aldoalcantara@MacBook-Pro-de-Aldo-2 All_Meta % pip list
Package            Version
-----
et-xmlfile         1.1.0
lxml               5.1.0
openpyxl           3.1.2
pip               24.0
PyPDF2             3.0.1
python-docx        1.1.0
typing_extensions  4.10.0
```

4.- Por ultimo procedo a correr el programa y verifico que todo funcione correctamente

```
(venv) aldoalcantara@MacBook-Pro-de-Aldo-2 All_Meta % python all_metadata_info.py
Metadatos del archivo SegundoMilestone.docx:
  author: Jhoana Monserrat Pimentel Lopez
  title:
  created: 2023-11-27 16:33:00
  modified: 2023-11-27 16:33:00

Metadatos del archivo Horario.xlsx:
  author: None
  title: None
  created: 2024-01-30 19:05:09
  modified: 2024-03-22 06:00:13

Metadatos del archivo Horario_Oficial.pdf:
  author: None
  creator: Crystal Reports
  producer: Powered By Crystal
  title: None
  subject: None
  created: None
  modified: None
```

## Codigo

```
import os
import re
from docx import Document
from openpyxl import load_workbook
from PyPDF2 import PdfReader

def extract_metadata_from_pdf(filename):
    pdf_file = PdfReader(filename)

    metadata = {
        "author": pdf_file.metadata.author,
        "creator": pdf_file.metadata.creator,
        "producer": pdf_file.metadata.producer,
        "title": pdf_file.metadata.title,
        "subject": pdf_file.metadata.subject,
        "created": pdf_file.metadata.get('/CreationDate', None),
        "modified": pdf_file.metadata.get('/ModDate', None)
    }
    return metadata

def extract_metadata_from_docx(filename):
    document = Document(filename)

    core_properties = document.core_properties

    metadata = {
        "author": core_properties.author,
        "title": core_properties.title,
        "created": core_properties.created,
        "modified": core_properties.modified
    }
    return metadata

def extract_metadata_from_xlsx(filename):
    wb = load_workbook(filename)

    author = None

    if wb.sheetnames: # Verifica si hay hojas de cálculo en el libro
        sheet = wb[wb.sheetnames[0]] # Obtiene la primera hoja de
        cálculo
```

```

        if sheet.dimensions: # Verifica si la hoja de cálculo contiene
datos
            author = sheet.cell(row=1, column=1).value # Obtiene el
valor de la primera celda

            metadata = {

                "author": author,

                "title": wb.properties.title,

                "created": wb.properties.created,

                "modified": wb.properties.modified

            }
            return metadata
def process_files(directory_path):
    for file_name in os.listdir(directory_path):
        file_path = os.path.join(directory_path, file_name)
        if os.path.isfile(file_path):
            file_extension = re.findall(r"\.(pdf|docx|xlsx)$", file_name)
            if file_extension:
                file_extension = file_extension[0]
                if file_extension == "pdf":
                    metadata = extract_metadata_from_pdf(file_path)
                elif file_extension == "docx":
                    metadata = extract_metadata_from_docx(file_path)
                elif file_extension == "xlsx":
                    metadata = extract_metadata_from_xlsx(file_path)
                print(f"Metadatos del archivo {file_name}:")
                for k, v in metadata.items():
                    print(f"    {k}: {v}")
                print()
# Ruta al directorio
directory_path = "/Users/aldoalcantara/Documents/ESCOM/Computer
Security/Python/All_Meta"
# Procesar los archivos en el directorio
process_files(directory_path)

```