Extracting_URLs_From_PDF_&_Downloading_Files_From_URLs

This notebook divided into 3 parts

- Step - 1: Getting PDF file paths from different directories
- Step - 2: Extracting the URLs from PDF files
- Step - 3: Downloading the data from the Extracted URL

Step - 1: Getting PDF file paths from different directories

In [1]:

```
main_dir = "G:\Learnings\Practice\Extracting_URLs_From_PDF_&_Downloading_Files_From_URLs"

req_file_paths = []

import os

for root, directory, files in os.walk(main_dir):
    for file in files: # Iterating all the files
        if file.endswith(".pdf"): # req file Extension
            req_file_paths.append(os.path.join(root, file)) #Joining root path and file pat
```

Step - 2: Extracting the URLs from PDF files

Usinf pdfx package we can extract the all the urls in the pdf file in the dictionary format

In [2]:

```
import pdfx
```

In [3]:

```
urls = []
for path in req_file_paths:
    pdf = pdfx.PDFx(path)

    # displaying pdf object
    print('-----PDF object-----')

    print(pdf)

    # displaying Metadata of the pdf like creator Name, Data, #of Pages etc.,

    print('-----Meta Data-----')
    print(pdf.get_metadata())

    # displaying File Names

    print('----FileName----')
    print(path.split('\\')[-1])

    # Extracting the URLs from the PDF files
    print('----Extraxted URL from the file----')
    print(pdf.get_references_as_dict())

    # Savong the urls in a list
    urls += pdf.get_references_as_dict()['url']
    print()
```

```
-----PDF object-----
<pdfx.PDFx object at 0x000002084F7B3790>
-----Meta Data-----
{'Producer': 'Skia/PDF m89', 'Pages': 1}
----FileName----
Python_Home_Site.pdf
----Extraxted URL from the file----
{'url': ['https://www.python.org/doc/']}

-----PDF object-----
<pdfx.PDFx object at 0x000002084F804250>
-----Meta Data-----
{'Producer': 'Skia/PDF m89', 'Pages': 1}
----FileName----
About_Python.pdf
----Extraxted URL from the file----
{'url': ['https://www.python.org/about/']}

-----PDF object-----
<pdfx.PDFx object at 0x000002084F809B50>
-----Meta Data-----
{'Producer': 'Skia/PDF m89', 'Pages': 1}
----FileName----
Download_Python.pdf
----Extraxted URL from the file----
{'url': ['https://www.python.org/downloads/']}

-----PDF object-----
<pdfx.PDFx object at 0x000002084F809AF0>
-----Meta Data-----
```

```
{'Producer': 'Skia/PDF m89', 'Pages': 1}
----FileName----
Python_books.pdf
----Extraxted URL from the file----
{'url': ['https://wiki.python.org/moin/PythonBooks']}

-----PDF object-----
<pdfx.PDFx object at 0x000002084F804C40>
-----Meta Data-----
{'Producer': 'Skia/PDF m89', 'Pages': 1}
----FileName----
Python_docs.pdf
----Extraxted URL from the file----
{'url': ['https://www.python.org/doc/']}

-----PDF object-----
<pdfx.PDFx object at 0x000002084F8090D0>
-----Meta Data-----
{'Producer': 'Skia/PDF m89', 'Pages': 2}
----FileName----
Python_docs_Beginners_Guide.pdf
----Extraxted URL from the file----
{'url': ['https://wiki.python.org/moin/BeginnersGuide/NonProgrammers', 'http
s://wiki.python.org/moin/BeginnersGuide/Overview', 'https://wiki.python.org/
moin/BeginnersGuide', 'python.org']}
```

Step - 3: Downloading the data from the Extracted URL

In [4]:

```
import requests
for url in range(len(urls)):
    if urls[url].startswith('http'):

        # URL of the file to be downloaded is defined as url
        response = requests.get(urls[url]) # create HTTP response object

        # send a HTTP request to the server and save
        # the HTTP response in a response object called res
        file_name = 'downloaded_files/file{}.html'.format(url) # Assuming all the url are h
        with open(file_name,'wb') as file:

            # Saving received content as a file in binary format

            # write the contents of the response (r.content)
            # to a new file in binary mode.
            file.write(response.content)

print('Files Downloaded Successfully 🐯')
```

```
Files Downloaded Successfully 🐯
```

Downloading Files from Google Drive

If the URL file is located in the Google Drive then follow the below procedure

original share link of a file in Google Drive will be as below:

https://docs.google.com/document/d/FILE_ID (https://docs.google.com/document/d/FILE_ID)

The FILE_ID is unique for every file in Google Drive. If you copy this FILE_ID and use it in the URL below, you'll get a direct link to download the file from Google Drive.

https://docs.google.com/document/d/DOC_FILE_ID/export?format=pdf (https://docs.google.com/document/d/DOC_FILE_ID/export?format=pdf) <-- format for docs file to be downloaded as pdf https://docs.google.com/document/d/DOC_FILE_ID/export?format=doc (https://docs.google.com/document/d/DOC_FILE_ID/export?format=doc) <-- format for docs file to be downloaded as doc https://drive.google.com/file/d/uc?export=download&id=DRIVE_FILE_ID (https://drive.google.com/file/d/uc?export=download&id=DRIVE_FILE_ID) <-- format for files to be downloaded from google drive

After creating the direct link to download now get the response of the url and download the content

For downloading the google doc file in .doc format

In [5]:

```
# Changing the original url to downloadable doc

org_url = "https://docs.google.com/document/d/1VTMfaT9oFXbjr0_yS9gvcnP5daVZV1OHBi0UhbjGeNc"
changed_url = org_url + '/export?format=doc'
import requests

# URL of the file to be downloaded is defined as url
response = requests.get(changed_url) # create HTTP response object

# send a HTTP request to the server and save
# the HTTP response in a response object called response

file_name = 'downloaded_files/sample.doc'
with open(file_name,'wb') as file:

    # Saving received content as a file in binary format

    # write the contents of the response (r.content)
    # to a new file in binary mode.
    file.write(response.content)

print('File Downloaded Successfully 🐯')
```

```
File Downloaded Successfully 🐯
```

For downloading the google doc file in .pdf format

In [6]:

```
# Changing the original url to downloadable doc

org_url = "https://docs.google.com/document/d/1VTMfaT9oFXbjr0_yS9gvcnP5daVZV1OHBi0UhbjGeNc"
changed_url = org_url + '/export?format=doc'

import requests

response = requests.get(changed_url, stream = True)

with open("downloaded_files/sample.pdf","wb") as pdf:
    for chunk in response.iter_content(chunk_size=1024):

        # writing one chunk at a time to pdf file
        if chunk:
            pdf.write(chunk)

print('File Downloaded Successfully 😺')
```

```
File Downloaded Successfully 😺
```