

1. Importar los módulos necesarios: os, requests, lxml.html, bs4, y logging.
2. Configurar el registro de errores en un archivo scraping.log.
3. Definir la clase Scraping:
 - 3.1. Definir el método `__init__` que toma la URL como argumento y la asigna a `self.url`.
 - 3.2. Definir el método `download_file` que toma los argumentos `url` y `folder`:
 - 3.2.1. Intentar:
 - 3.2.1.1. Realizar una solicitud GET a la URL.
 - 3.2.1.2. Si el código de estado de la respuesta es 200 (éxito):
 - 3.2.1.2.1. Obtener el nombre del archivo de la URL.
 - 3.2.1.2.2. Construir la ruta del archivo en la carpeta especificada.
 - 3.2.1.2.3. Abrir el archivo en modo escritura binaria y escribir el contenido de la respuesta.
 - 3.2.1.2.4. Devolver la ruta del archivo.
 - 3.2.1.3. Si el código de estado no es 200, registrar un error.
 - 3.2.2. Capturar cualquier excepción y registrar un error.
4. En la clase Scraping, definir el método `scraping_beautiful_soup`:
 - 4.1. Intentar:
 - 4.1.1. Imprimir un mensaje indicando que se están obteniendo imágenes de la URL.
 - 4.1.2. Realizar una solicitud GET a la URL.
 - 4.1.3. Crear un objeto BeautifulSoup con el contenido de la respuesta.
 - 4.1.4. Crear el directorio "imagenes" si no existe.
 - 4.1.5. Inicializar una lista vacía `image_paths`.
 - 4.1.6. Iterar sobre las etiquetas `` en el HTML:
 - 4.1.6.1. Construir la URL completa para descargar la imagen.
 - 4.1.6.2. Llamar al método `download_file` con la URL y la carpeta "imagenes".

4.1.6.3. Si se descarga un archivo, agregar la URL a
image_paths.

4.1.7. Abrir el archivo "url_img.txt" en modo escritura.

4.1.8. Escribir cada URL de image_paths en el archivo "url_img.txt".

4.2. Capturar cualquier excepción y registrar un error.

5. En la clase Scraping, definir el método scraping_images:

5.1. Imprimir un mensaje indicando que se están obteniendo imágenes de la
URL.

5.2. Intentar:

5.2.1. Realizar una solicitud GET a la URL.

5.2.2. Crear un objeto lxml.html con el contenido de la respuesta.

5.2.3. Obtener las URLs de las imágenes utilizando XPath.

5.2.4. Crear el directorio "imagenes" si no existe.

5.2.5. Inicializar una lista vacía image_paths.

5.2.6. Iterar sobre las URLs de imágenes:

5.2.6.1. Construir la URL completa para descargar la imagen.

5.2.6.2. Llamar al método download_file con la URL y la
carpeta "imagenes".

5.2.6.3. Si se descarga un archivo, agregar la URL a
image_paths.

5.2.7. Abrir el archivo "url_img.txt" en modo escritura.

5.2.8. Escribir cada URL de image_paths en el archivo "url_img.txt".

5.3. Capturar cualquier excepción y registrar un error.

6. En la clase Scraping, definir el método scraping_pdf:

6.1. Imprimir un mensaje indicando que se están obteniendo PDFs de la URL.

6.2. Intentar:

6.2.1. Realizar una solicitud GET a la URL.

6.2.2. Crear un objeto lxml.html con el contenido de la respuesta.

6.2.3. Obtener las URLs de los PDFs utilizando XPath.

6.2.4. Crear el directorio "pdfs" si no existe.

6.2.5. Inicializar una lista vacía pdf_paths.

6.2.6. Iterar sobre las URLs de PDFs:

6.2.6.1. Construir la URL completa para descargar el PDF.

6.2.6.2. Llamar al método download_file con la URL y la carpeta "pdfs".

6.2.6.3. Si se descarga un archivo, agregar la URL a pdf_paths.

6.2.7. Abrir el archivo "url_pdf.txt" en modo escritura.

6.2.8. Escribir cada URL de pdf_paths en el archivo "url_pdf.txt".

6.3. Capturar cualquier excepción y registrar un error.

7. Definir la función scraping_web que toma la URL como argumento:

7.1. Crear una instancia de la clase Scraping con la URL.

7.2. Llamar al método scraping_beautiful_soup de la instancia.

7.3. Llamar al método scraping_images de la instancia.

7.4. Llamar al método scraping_pdf de la instancia.

8. Fin.