PCATParser

Table of contents

Introduction

It is not enough to find a set of web pages, we need to extract the information from each of the web pages we find. This is done by the **PCATParser**. The **PCATParser** is a general purpose web scraper for getting the visible text from a web page. We also integrated some site-specific web scrapers to get more structured data into Site_Crawler_Parser_All.

Documentation

eightk_parser(link)

Parses an SEC document known as an 8-K

Parameters

- link (string) : the URL for the 8-K

Returns

- string : the important text for the 8-K

ex21_parser(link)

Parses an SEC document known as an EX-21

Parameters

- link (string) : the URL for the EX-21

Returns

- list of strings : the subsidiaries in the company listed in the EX-21

get_PDF_content(query_string, link)

Gets all of the text from a PDF document

Parameters

- query_string (string) : the query that generated the PDF document
- link (string) : the URL for the document

Returns

- string : the visible text in the PDF

parser_iter(query_string, linkList)

Parses the URLs in linkList using a timeout of 60 seconds on each page (a la try_one) and yields them as dictionaries.

Parameters

- query_string (string) : the generating query, default is "test"
- linkList (list of strings) : list of URLs for the documents you would like to parse

Returns

- dict (yields many)
    - dict'text' : the visible text on the web page
    - dict'html' : the HTML code of the page (if it is HTML based)
    - dict'pdf' : the PDF code of the page (if it is PDF based)

parse_single_page(link, query_string = "test")

Gets all of the text from web page

Parameters

- link (string) : the URL for the document
- query_string (string) : the generating query, default is "test"

Returns

- tuple (bytes, string) : the source code (HTML/PDF) of the web page and the visible text

tag_visible(element)

Determines if an HTML tag is visible

Parameters

- element (BeautifulSoup.element) : an HTML element

Returns

- bool : True if the element is visible, False else

tenk_parser(link)

Parses an SEC document known as an 10-K

Parameters

- link (string) : the URL for the 10-K

Returns

- string : the important information in the 10-K

text_from_html(body)

Gets all of the visible text from the body of an HTML document

Parameters

- body (string) : the body of an HTML document

Returns

- string : the visible text in the body

try_one(func, t, **kwargs)

Calls the function with the keyword arguments and after t seconds, interupts the call and moves on

Parameters

- func (function) : the function to be called
- t (int) : the number of seconds
- **kwargs (keyword-arguments) : arguments you'd like to pass to func

Returns

- func's return type

wikiParser(company)

Search the Wikipedia page for a company and get wikipedia infobox together with all other contents

Parameters

- company (str) : the company you would like to query Wikipedia for

Returns

- tuple
    - dict : a dictionary of all other contents on wikipedia
    - dict : a dictionary of wikipedia infobox
    - str : page title
    - str : page url
    - beautifulsoup.table : wikipedia infobox HTML