
CyPDFTools

Release 0.0.1

Apr 11, 2021

Table of Contents

1	PDFFile	1
2	PDF Objects	3
3	PDFObjectsParser	5
4	PDFStructureObjects	7
5	Utilities	9
6	The MIT License (MIT)	11
7	Indices and tables	13
	Python Module Index	15
	Index	17

CHAPTER 1

PDFFile

```
class PDFFile.PDFFile (filePath, to_pickle=False)
    Loads the PDFFile to modify it

    delete_object (object_number: int)
        Removes an object from the objects dict

        Parameters object_number – Object number

    delete_page (page_number: int)
        Deletes the given page

        Parameters page_number – Number of the page

    extract_object (number)
        Extracts object from the pdf file

        Parameters number – XrefIndex of object

        Returns PDF Object

    getFromPDFDict (key: int)
        Get object from pdf

        Parameters key – object number

        Returns PDFObject

    get_RootOutline () → PDFStructureObjects.PDFObject
        Gets the root outline object

        Returns Outline PDFObject

    get_document_catalog ()
        Gets the document_catalog from the pdf

        Returns Document catalog

    get_firstOutlineItem () → PDFStructureObjects.PDFObject
        Get the first actual OutlineItem
```

Returns First actual OutlineItem

get_lastOutlineItem () → PDFStructureObjects.PDFObject
Get the last actual OutlineItem

Returns Last actual OutlineItem

get_page_root ()
Extracts the page root from the pdf

Returns PDF Root

get_pages () → list
Gathers all pages

Returns list of pages

has_outline ()
Checks if the document has an outline

increment_references (*n: int*) → None
Increments all Indirect Object references of the pdf

Parameters **n** – offset

removeFromPDFDict (*key: int*)
Removes object from pdf

Parameters **key** – object number

rotate_all (*rotation: int*)
Rotates all pages

Parameters **rotation** – Degrees

rotate_page (*index: int, rotation: int*)
Rotates the given page

Parameters

- **index** – Page index
- **rotation** – Degrees

save (*path*)
Writes the contents of the pdf to disk

seek_object (*number: int*) → None
Moves the pointer to the nth object

Parameters **number** – Object's index in XRefTable

class `PDFObjects.IndirectObjectRef` (*objectref*, *generation_number*)
Represents indirect object references 7.3.10 PDF 32000-1:2008

offset_references (*offset*: *int*) → *None*
Increments the reference objects inside the data structure

Parameters **offset** – offset value

class `PDFObjects.PDFArray` (*data*: *list*)
A Wrapper for PDF Arrays 7.3.6 PDF 32000-1:2008

offset_references (*offset*: *int*)
Increments the reference objects inside the data structure

Parameters **offset** – offset value

class `PDFObjects.PDFDict` (*data*: *dict*)
A wrapper for PDF Dictionaries 7.3.8 PDF 32000-1:2008

offset_references (*offset*: *int*) → *None*
Increments the reference objects inside the data structure

Parameters **offset** – offset value

PDFObjectsParser

`PDFObjectsParser.classify_stream` (*stream_iter: utils.ObjectIter*, *letter=None*)

Classifies and parses the given stream

Parameters

- **stream_iter** – A stream whose 1st character indicates its type
- **letter** – Passes the letter that was consumed elsewhere

Returns A PDF Object or a standard object

`PDFObjectsParser.extract_array` (*stream: utils.ObjectIter*) → `PDFObjects.PDFArray`

Extracts array from steam

Parameters **stream** – `ObjectIter`

Returns `PDFArray`

`PDFObjectsParser.extract_name` (*stream: utils.ObjectIter*) → `bytes`

Extracts the next name from the iterator (7.3.5 PDF 32000-1:2008)

Parameters **stream** – A stream whose forward slash / was just consumed

Returns Bytes containing the name

`PDFObjectsParser.parse_dictionary` (*pdf_stream*) → `PDFObjects.PDFDict`

Parses `PDFDictionary` objects

Parameters **pdf_stream** – `Object Stream`

Returns `PDFObjects.PDFDict` object

`PDFObjectsParser.parse_literalStrings` (*stream: utils.ObjectIter*) → `bytes`

Parses string literals (7.3.4.2) PDF 32000-1:2008

Parameters **stream** – A stream whose opening round bracket (was just consumed

Returns The string literal including the round brackets

`PDFObjectsParser.parse_numeric` (*init: bytes, stream: utils.ObjectIter*)

Parses numeric objects

Parameters

- **init** – The char that *PDFObjectsParser.classify_stream()* already consumed
- **stream** – Object Stream

Returns A number or a reference object

PDFStructureObjects

class PDFStructureObjects.**PDFStream**(*stream_dict: PDFObjects.PDFDict, object_number, object_rev, start_address, inuse, file, pdfObjectsFunc*)

Represents objects that contain a stream

offset_references (*offset: int*) → None

Increments the reference objects inside the data structure

Parameters **offset** – offset value

read_stream () → bytes

Read stream from file

Parameters **file** – file_reader

Returns streamobject

class PDFStructureObjects.**XRefTable**(*xref_table: list, parsed=False*)

Represents the XRef table of a PDF file

static parse_table (*table: list*) → list

Parses list of XRef bytes

Parameters **table** – A list containing XRef entries

Returns list of XRef Entries

class PDFStructureObjects.**XrefEntry**(*address, revision, in_use_entry*)

address

Alias for field number 0

in_use_entry

Alias for field number 2

revision

Alias for field number 1

class `utils.ObjectIter` (*stream: bytes, pointer=0*)
Used to iterate over objects

finish_number () → bytes
Move stream until a separating character is found
Returns Resulting int

move_pointer (*n: int*) → None
Moves the pointer *n* characters
Parameters *n* – Number of characters

move_to (*item: bytes*)
Moves the iterator to given item
Parameters *item* – item to move to
Returns Items since the beginning of iteration till end

peek (*n: int = 1*)
Returns the next *n* chars without incrementing the counter
Parameters *n* – number of characters
Returns Returns the next *n* chars without incrementing the counter

prev () → bytes
Decrements the counter
Returns Previous element

reversePeek (*n: int = 1*) → bytes
Returns the previous *n* chars without incrementing the counter
Parameters *n* – number of characters
Returns Returns the next *n* chars without incrementing the counter

skip_space () → None

Moves stream to the next non whitespace char

Parameters **stream** – Any iterable object

CHAPTER 6

The MIT License (MIT)

Copyright © 2020 John Sorial

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

p

PDFFile, [1](#)
PDFObjects, [3](#)
PDFObjectsParser, [5](#)
PDFStructureObjects, [7](#)

u

utils, [9](#)

A

address (*PDFStructureObjects.XrefEntry* attribute), 7

C

classify_stream() (in module *PDFObjectsParser*), 5

D

delete_object() (*PDFFile.PDFFile* method), 1
delete_page() (*PDFFile.PDFFile* method), 1

E

extract_array() (in module *PDFObjectsParser*), 5
extract_name() (in module *PDFObjectsParser*), 5
extract_object() (*PDFFile.PDFFile* method), 1

F

finish_number() (*utils.ObjectIter* method), 9

G

get_document_catalog() (*PDFFile.PDFFile* method), 1
get_firstOutlineItem() (*PDFFile.PDFFile* method), 1
get_lastOutlineItem() (*PDFFile.PDFFile* method), 2
get_page_root() (*PDFFile.PDFFile* method), 2
get_pages() (*PDFFile.PDFFile* method), 2
get_RootOutline() (*PDFFile.PDFFile* method), 1
getFromPDFDict() (*PDFFile.PDFFile* method), 1

H

has_outline() (*PDFFile.PDFFile* method), 2

I

in_use_entry (*PDFStructureObjects.XrefEntry* attribute), 7
increment_references() (*PDFFile.PDFFile* method), 2

IndirectObjectRef (class in *PDFObjects*), 3

M

move_pointer() (*utils.ObjectIter* method), 9
move_to() (*utils.ObjectIter* method), 9

O

ObjectIter (class in *utils*), 9
offset_references() (*PDFObjects.IndirectObjectRef* method), 3
offset_references() (*PDFObjects.PDFArray* method), 3
offset_references() (*PDFObjects.PDFDict* method), 3
offset_references() (*PDFStructureObjects.PDFStream* method), 7

P

parse_dictionary() (in module *PDFObjectsParser*), 5
parse_literalStrings() (in module *PDFObjectsParser*), 5
parse_numeric() (in module *PDFObjectsParser*), 5
parse_table() (*PDFStructureObjects.XrefTable* static method), 7
PDFArray (class in *PDFObjects*), 3
PDFDict (class in *PDFObjects*), 3
PDFFile (class in *PDFFile*), 1
PDFFile (module), 1
PDFObjects (module), 3
PDFObjectsParser (module), 5
PDFStream (class in *PDFStructureObjects*), 7
PDFStructureObjects (module), 7
peek() (*utils.ObjectIter* method), 9
prev() (*utils.ObjectIter* method), 9

R

read_stream() (*PDFStructureObjects.PDFStream* method), 7

`removeFromPDFDict()` (*PDFFile.PDFFile method*), [2](#)
`reversePeek()` (*utils.ObjectIter method*), [9](#)
`revision` (*PDFStructureObjects.XrefEntry attribute*), [7](#)
`rotate_all()` (*PDFFile.PDFFile method*), [2](#)
`rotate_page()` (*PDFFile.PDFFile method*), [2](#)

S

`save()` (*PDFFile.PDFFile method*), [2](#)
`seek_object()` (*PDFFile.PDFFile method*), [2](#)
`skip_space()` (*utils.ObjectIter method*), [9](#)

U

`utils` (*module*), [9](#)

X

`XrefEntry` (*class in PDFStructureObjects*), [7](#)
`XrefTable` (*class in PDFStructureObjects*), [7](#)