

The PdfFileReader Class

`class PyPDF2.PdfFileReader(stream, strict=True, warndest=None, overwriteWarnings=True)`

Initializes a PdfFileReader object. This operation can take some time, as the PDF stream's cross-reference tables are read into memory.

Parameters:

- **stream** – A File object or an object that supports the standard read and seek methods similar to a File object. Could also be a string representing a path to a PDF file.
- **strict** (*bool*) – Determines whether user should be warned of all problems and also causes some correctable problems to be fatal. Defaults to `True`.
- **warndest** – Destination for logging warnings (defaults to `sys.stderr`).
- **overwriteWarnings** (*bool*) – Determines whether to override Python's `warnings.py` module with a custom implementation (defaults to `True`).

decrypt(*password*)

When using an encrypted / secured PDF file with the PDF Standard encryption handler, this function will allow the file to be decrypted. It checks the given password against the document's user password and owner password, and then stores the resulting decryption key if either password is correct.

It does not matter which password was matched. Both passwords provide the correct decryption key that will allow the document to be used with this library.

Parameters: **password** (*str*) – The password to match.

Returns: `0` if the password failed, `1` if the password matched the user password, and `2` if the password matched the owner password.

Return `int`

type:

Raises NotImplementedError:

if document uses an unsupported encryption method.

documentInfo

Read-only property that accesses the `getDocumentInfo()` function.

getDestinationPageNumber(*destination*)

Retrieve page number of a given Destination object

Parameters: **destination** (*Destination*) – The destination to get page number. Should be an instance of `Destination`

Returns: the page number or `-1` if page not found

Return `int`

type:

getDocumentInfo()

Retrieves the PDF file's document information dictionary, if it exists. Note that some PDF files use metadata streams instead of docinfo dictionaries, and these metadata streams will not be accessed by this function.

Returns: the document information of this PDF file

Return type: `DocumentInformation` or `None` if none exists.

getFields(*tree=None, retval=None, fileobj=None*)

Extracts field data if this PDF contains interactive form fields. The *tree* and *retval* parameters are for recursive use.

Parameters: `fileobj` – A file object (usually a text file) to write a report to on all interactive form fields found.

Returns: A dictionary where each key is a field name, and each value is a [Field](#) object. By default, the mapping name is used for keys.

Return type: dict, or `None` if form data could not be located.

getFormTextFields()

Retrieves form fields from the document with textual data (inputs, dropdowns)

getNamedDestinations(*tree=None, retval=None*)

Retrieves the named destinations present in the document.

Returns: a dictionary which maps names to [Destinations](#).

Return type: dict

getNumPages()

Calculates the number of pages in this PDF file.

Returns: number of pages

Return type: int

Raises PdfReadError:

if file is encrypted and restrictions prevent this action.

getOutlines(*node=None, outlines=None*)

Retrieves the document outline present in the document.

Returns: a nested list of [Destinations](#).

getPage(*pageNumber*)

Retrieves a page by number from this PDF file.

Parameters: `pageNumber` (*int*) – The page number to retrieve (pages begin at zero)

Returns: a [PageObject](#) instance.

Return type: [PageObject](#)

getPageLayout()

Get the page layout. See [setPageLayout\(.\)](#) for a description of valid layouts.

Returns: Page layout currently being used.

Return type: `str`, `None` if not specified

getPageMode()

Get the page mode. See [setPageMode\(.\)](#) for a description of valid modes.

Returns: Page mode currently being used.

Return type: `str`, `None` if not specified

getPageNumber(*page*)

Retrieve page number of a given PageObject

Parameters: `page` ([PageObject](#)) – The page to get page number. Should be an instance of [PageObject](#)

Returns: the page number or -1 if page not found

Return type: int

getXmpMetadata()

Retrieves XMP (Extensible Metadata Platform) data from the PDF document root.

Returns: a [XmpInformation](#) instance that can be used to access XMP metadata from the document.

Return type: [XmpInformation](#) or `None` if no metadata was found on the document root.

isEncrypted

Read-only boolean property showing whether this PDF file is encrypted. Note that this property, if true, will remain true even after the [decrypt\(\)](#) method is called.

namedDestinations

Read-only property that accesses the [getNamedDestinations\(\)](#) function.

numPages

Read-only property that accesses the [getNumPages\(\)](#) function.

outlines

Read-only property that accesses the [getOutlines\(\)](#) function.

pageLayout

Read-only property accessing the [getPageLayout\(\)](#) method.

pageMode

Read-only property accessing the [getPageMode\(\)](#) method.

pages

Read-only property that emulates a list based upon the [getNumPages\(\)](#) and [getPage\(\)](#) methods.

xmpMetadata

Read-only property that accesses the [getXmpMetadata\(\)](#) function.