



PyMuPDF Documentation

- [Introduction](#)
 - [Note on the Name *fitz*](#)
 - [License and Copyright](#)
 - [Covered Version](#)
 - [Installation](#)
 - [Notes](#)
 - [Install from source without using an sdist](#)
 - [Enabling Integrated OCR Support](#)
 - [Tutorial](#)
 - [Importing the Bindings](#)
 - [Opening a Document](#)
 - [Some Document Methods and Attributes](#)
 - [Accessing Meta Data](#)
 - [Working with Outlines](#)
 - [Working with Pages](#)
 - [Inspecting the Links, Annotations or Form Fields of a Page](#)
 - [Rendering a Page](#)
 - [Saving the Page Image in a File](#)
 - [Displaying the Image in GUIs](#)
 - [wxPython](#)
 - [Tkinter](#)
 - [PyQt4, PyQt5, PySide](#)
 - [Extracting Text and Images](#)
 - [Searching for Text](#)
 - [PDF Maintenance](#)
 - [Modifying, Creating, Re-arranging and Deleting Pages](#)
 - [Joining and Splitting PDF Documents](#)
 - [Embedding Data](#)
 - [Saving](#)
 - [Closing](#)
 - [Further Reading](#)
- [Recipes: Images](#)

- [How to Make Images from Document Pages](#)
- [How to Increase Image Resolution](#)
- [How to Create Partial Pixmaps \(Clips\)](#)
- [How to Zoom a Clip to a GUI Window](#)
- [How to Create or Suppress Annotation Images](#)
- [How to Extract Images: Non-PDF Documents](#)
- [How to Extract Images: PDF Documents](#)
- [How to Handle Image Masks](#)
- [How to Make one PDF of all your Pictures \(or Files\)](#)
- [How to Create Vector Images](#)
- [How to Convert Images](#)
- [How to Use Pixmaps: Glueing Images](#)
- [How to Use Pixmaps: Making a Fractal](#)
- [How to Interface with NumPy](#)
- [How to Add Images to a PDF Page](#)
- [Recipes: Text](#)
 - [How to Extract all Document Text](#)
 - [How to Extract Text from within a Rectangle](#)
 - [How to Extract Text in Natural Reading Order](#)
 - [How to Extract Tables from Documents](#)
 - [How to Mark Extracted Text](#)
 - [How to Mark Searched Text](#)
 - [How to Mark Non-horizontal Text](#)
 - [How to Analyze Font Characteristics](#)
 - [How to Insert Text](#)
 - [How to Write Text Lines](#)
 - [How to Fill a Text Box](#)
 - [How to Use Non-Standard Encoding](#)
- [Recipes: Annotations](#)
 - [How to Add and Modify Annotations](#)
 - [How to Use FreeText](#)
 - [Using Buttons and JavaScript](#)
 - [How to Use Ink Annotations](#)
- [Recipes: Drawing and Graphics](#)
 - [Extracting Drawings](#)
- [Recipes: Multiprocessing](#)
- [Recipes: General](#)
 - [How to Open with a Wrong File Extension](#)
 - [How to Embed or Attach Files](#)
 - [How to Delete and Re-Arrange Pages](#)

- [How to Join PDFs](#)
- [How to Add Pages](#)
- [How To Dynamically Clean Up Corrupt PDFs](#)
- [How to Split Single Pages](#)
- [How to Combine Single Pages](#)
- [How to Convert Any Document to PDF](#)
- [How to Deal with Messages Issued by MuPDF](#)
- [How to Deal with PDF Encryption](#)
- [Recipes: Common Issues and their Solutions](#)
 - [Changing Annotations: Unexpected Behaviour](#)
 - [Problem](#)
 - [Cause](#)
 - [Solutions](#)
 - [Misplaced Item Insertions on PDF Pages](#)
 - [Problem](#)
 - [Cause](#)
 - [Solutions](#)
 - [Missing or Unreadable Extracted Text](#)
 - [Problem: no text is extracted](#)
 - [Cause](#)
 - [Solution](#)
 - [Problem: unreadable text](#)
 - [Cause](#)
 - [Solution](#)
- [Recipes: Low-Level Interfaces](#)
 - [How to Iterate through the `xref` Table](#)
 - [How to Handle Object Streams](#)
 - [How to Handle Page Contents](#)
 - [How to Access the PDF Catalog](#)
 - [How to Access the PDF File Trailer](#)
 - [How to Access XML Metadata](#)
 - [How to Extend PDF Metadata](#)
 - [How to Read and Update PDF Objects](#)
- [Recipes: Journalling](#)
 - [Example Session 1](#)
 - [Example Session 2](#)
- [Module *fitz*](#)
 - [Invocation](#)

- [Cleaning and Copying](#)
- [Extracting Fonts and Images](#)
- [Joining PDF Documents](#)
- [Low Level Information](#)
- [Embedded Files Commands](#)

- [Information](#)
- [Extraction](#)
- [Deletion](#)
- [Insertion](#)
- [Updates](#)
- [Copying](#)

- [Text Extraction](#)

- [Classes](#)

- [Annot](#)

- [Annot](#)
 - [Annot.get_pixmap\(\)](#)
 - [Annot.get_text\(\)](#)
 - [Annot.get_textbox\(\)](#)
 - [Annot.set_info\(\)](#)
 - [Annot.set_line_ends\(\)](#)
 - [Annot.set_oc\(\)](#)
 - [Annot.get_oc\(\)](#)
 - [Annot.set_irt_xref\(\)](#)
 - [Annot.set_open\(\)](#)
 - [Annot.set_popup\(\)](#)
 - [Annot.set_opacity\(\)](#)
 - [Annot.blendmode](#)
 - [Annot.set_blendmode\(\)](#)
 - [Annot.set_name\(\)](#)
 - [Annot.set_rect\(\)](#)
 - [Annot.set_rotation\(\)](#)
 - [Annot.set_border\(\)](#)
 - [Annot.set_flags\(\)](#)
 - [Annot.set_colors\(\)](#)
 - [Annot.delete_responses\(\)](#)
 - [Annot.update\(\)](#)
 - [Annot.file_info\(\)](#)
 - [Annot.get_file\(\)](#)
 - [Annot.update_file\(\)](#)
 - [Annot.get_sound\(\)](#)

- `Annot.opacity`
- `Annot.parent`
- `Annot.rotation`
- `Annot.rect`
- `Annot.next`
- `Annot.type`
- `Annot.info`
- `Annot.flags`
- `Annot.line_ends`
- `Annot.vertices`
- `Annot.colors`
- `Annot.xref`
- `Annot.irt_xref`
- `Annot.popup_xref`
- `Annot.has_popup`
- `Annot.is_open`
- `Annot.popup_rect`
- `Annot.border`

- [Annotation Icons in MuPDF](#)
- [Example](#)

◦ [Colorspace](#)

- `Colorspace`
 - `Colorspace.__init__()`
 - `Colorspace.name`
 - `Colorspace.n`

◦ [DisplayList](#)

- `DisplayList`
 - `DisplayList.__init__()`
 - `DisplayList.run()`
 - `DisplayList.get_pixmap()`
 - `DisplayList.get_textpage()`
 - `DisplayList.rect`

◦ [Document](#)

- `Document`
 - `Document.__init__()`
 - `Document.get_oc()`
 - `Document.set_oc()`

- `Document.get_layers()`
- `Document.add_layer()`
- `Document.switch_layer()`
- `Document.add_ocg()`
- `Document.set_ocmd()`
- `Document.get_ocmd()`
- `Document.get_layer()`
- `Document.set_layer()`
- `Document.get_ocgs()`
- `Document.layer_ui_configs()`
- `Document.set_layer_ui_config()`
- `Document.authenticate()`
- `Document.get_page_numbers()`
- `Document.get_page_labels()`
- `Document.set_page_labels()`
- `Document.make_bookmark()`
- `Document.find_bookmark()`
- `Document.chapter_page_count()`
- `Document.next_location()`
- `Document.prev_location()`
- `Document.load_page()`
- `Document.reload_page()`
- `Document.page_croptbox()`
- `Document.page_xref()`
- `Document.pages()`
- `Document.convert_to_pdf()`
- `Document.get_toc()`
- `Document.xref_get_keys()`
- `Document.xref_get_key()`
- `Document.xref_set_key()`
- `Document.get_page_pixmap()`
- `Document.get_page_xobjects()`
- `Document.get_page_images()`
- `Document.get_page_fonts()`
- `Document.get_page_text()`
- `Document.layout()`
- `Document.select()`
- `Document.set_metadata()`
- `Document.get_xml_metadata()`
- `Document.set_xml_metadata()`
- `Document.set_toc()`

- `Document.outline_xref()`
- `Document.del_toc_item()`
- `Document.set_toc_item()`
- `Document.can_save_incrementally()`
- `Document.scrub()`
- `Document.save()`
- `Document.ez_save()`
- `Document.saveIncr()`
- `Document.tobytes()`
- `Document.search_page_for()`
- `Document.insert_pdf()`
- `Document.new_page()`
- `Document.insert_page()`
- `Document.delete_page()`
- `Document.delete_pages()`
- `Document.copy_page()`
- `Document.fullcopy_page()`
- `Document.move_page()`
- `Document.need_appearances()`
- `Document.get_sigflags()`
- `Document.embfile_add()`
- `Document.embfile_count()`
- `Document.embfile_get()`
- `Document.embfile_del()`
- `Document.embfile_info()`
- `Document.embfile_names()`
- `Document.embfile_upd()`
- `Document.close()`
- `Document.xref_object()`
- `Document.pdf_catalog()`
- `Document.pdf_trailer()`
- `Document.xref_stream()`
- `Document.xref_stream_raw()`
- `Document.update_object()`
- `Document.update_stream()`
- `Document.xref_copy()`
- `Document.extract_image()`
- `Document.extract_font()`
- `Document.xref_xml_metadata()`
- `Document.has_links()`
- `Document.has_annots()`

- `Document.subset_fonts()`
- `Document.journal_enable()`
- `Document.journal_start_op()`
- `Document.journal_stop_op()`
- `Document.journal_position()`
- `Document.journal_op_name()`
- `Document.journal_can_do()`
- `Document.journal_undo()`
- `Document.journal_redo()`
- `Document.journal_save()`
- `Document.journal_load()`
- `Document.save_snapshot()`
- `Document.outline`
- `Document.is_closed`
- `Document.is_dirty`
- `Document.is_pdf`
- `Document.is_form_pdf`
- `Document.is_reflowable`
- `Document.is_repaired`
- `Document.needs_pass`
- `Document.is_encrypted`
- `Document.permissions`
- `Document.metadata`
- `Document.name`
- `Document.page_count`
- `Document.chapter_count`
- `Document.last_location`
- `Document.FormFonts`
- `set_metadata()` [Example](#)
- `set_toc()` [Demonstration](#)
- `insert_pdf()` [Examples](#)
- [Other Examples](#)

◦ [Font](#)

- `Font`
 - `Font.has_glyph()`
 - `Font.valid_codepoints()`
 - `Font.glyph_advance()`
 - `Font.glyph_name_to_unicode()`
 - `Font.glyph_bbox()`
 - `Font.unicode_to_glyph_name()`

- `Font.text_length()`
- `Font.char_lengths()`
- `Font.buffer`
- `Font.flags`
- `Font.name`
- `Font.bbox`
- `Font.glyph_count`
- `Font.ascender`
- `Font.descender`
- `Font.is_writable`

- Identity

- IRect

- IRect
 - `IRect.__init__()`
 - `IRect.__init__()`
 - `IRect.__init__()`
 - `IRect.__init__()`
 - `IRect.get_area()`
 - `IRect.intersect()`
 - `IRect.contains()`
 - `IRect.intersects()`
 - `IRect.torect()`
 - `IRect.morph()`
 - `IRect.norm()`
 - `IRect.normalize()`
 - `IRect.top_left`
 - `IRect.tl`
 - `IRect.top_right`
 - `IRect.tr`
 - `IRect.bottom_left`
 - `IRect.bl`
 - `IRect.bottom_right`
 - `IRect.br`
 - `IRect.rect`
 - `IRect.quad`
 - `IRect.width`
 - `IRect.height`
 - `IRect.x0`
 - `IRect.y0`
 - `IRect.x1`

- `IRect.y1`
- `IRect.is_infinite`
- `IRect.is_empty`

- [Link](#)

- `Link`
 - `Link.set_border()`
 - `Link.set_colors()`
 - `Link.set_flags()`
 - `Link.flags`
 - `Link.colors`
 - `Link.border`
 - `Link.rect`
 - `Link.isExternal`
 - `Link.uri`
 - `Link.xref`
 - `Link.next`
 - `Link.dest`

- [linkDest](#)

- `linkDest`
 - `linkDest.dest`
 - `linkDest.fileSpec`
 - `linkDest.flags`
 - `linkDest.isMap`
 - `linkDest.isUri`
 - `linkDest.kind`
 - `linkDest.lt`
 - `linkDest.named`
 - `linkDest.newWindow`
 - `linkDest.page`
 - `linkDest.rb`
 - `linkDest.uri`

- [Matrix](#)

- `Matrix`
 - `Matrix.__init__()`
 - `Matrix.__init__()`
 - `Matrix.__init__()`
 - `Matrix.__init__()`

- `Matrix.__init__()`
- `Matrix.__init__()`
- `Matrix.__init__()`
- `Matrix.norm()`
- `Matrix.prerotate()`
- `Matrix.prescale()`
- `Matrix.preshear()`
- `Matrix.pretranslate()`
- `Matrix.concat()`
- `Matrix.invert()`
- `Matrix.a`
- `Matrix.b`
- `Matrix.c`
- `Matrix.d`
- `Matrix.e`
- `Matrix.f`
- `Matrix.is_rectilinear`

- [Examples](#)
- [Shifting](#)
- [Flipping](#)
- [Shearing](#)
- [Rotating](#)

- [Outline](#)

- `Outline`
 - `Outline.down`
 - `Outline.next`
 - `Outline.page`
 - `Outline.title`
 - `Outline.is_open`
 - `Outline.is_external`
 - `Outline.uri`
 - `Outline.dest`

- [Page](#)

- [Modifying Pages](#)
 - `Page`
- [Description of `get_links\(\)` Entries](#)
- [Notes on Supporting Links](#)
 - [Reading \(pertains to method `get_links\(\)` and the `first_link` property chain\)](#)

- [Writing](#)
- [Homologous Methods of Document and Page](#)
- [Pixmap](#)
 - [Pixmap](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.__init__\(\)](#)
 - [Pixmap.clear_with\(\)](#)
 - [Pixmap.tint_with\(\)](#)
 - [Pixmap.gamma_with\(\)](#)
 - [Pixmap.shrink\(\)](#)
 - [Pixmap.pixel\(\)](#)
 - [Pixmap.set_pixel\(\)](#)
 - [Pixmap.set_rect\(\)](#)
 - [Pixmap.set_origin\(\)](#)
 - [Pixmap.set_dpi\(\)](#)
 - [Pixmap.set_alpha\(\)](#)
 - [Pixmap.invert_irect\(\)](#)
 - [Pixmap.copy\(\)](#)
 - [Pixmap.save\(\)](#)
 - [Pixmap.pdfocr_save\(\)](#)
 - [Pixmap.pdfocr_tobytes\(\)](#)
 - [Pixmap.tobytes\(\)](#)
 - [Pixmap.pil_save\(\)](#)
 - [Pixmap.pil_tobytes\(\)](#)
 - [Pixmap.warp\(\)](#)
 - [Pixmap.color_count\(\)](#)
 - [Pixmap.color_topusage\(\)](#)
 - [Pixmap.alpha](#)
 - [Pixmap.digest](#)
 - [Pixmap.colorspace](#)
 - [Pixmap.stride](#)
 - [Pixmap.is_monochrome](#)
 - [Pixmap.is_unicolor](#)

- `Pixmap.irect`
 - `Pixmap.samples`
 - `Pixmap.samples_mv`
 - `Pixmap.samples_ptr`
 - `Pixmap.size`
 - `Pixmap.width`
 - `Pixmap.w`
 - `Pixmap.height`
 - `Pixmap.h`
 - `Pixmap.x`
 - `Pixmap.y`
 - `Pixmap.n`
 - `Pixmap.xres`
 - `Pixmap.yres`
 - `Pixmap.interpolate`
- [Supported Input Image Formats](#)
 - [Supported Output Image Formats](#)

- [Point](#)

- `Point`
 - `Point.__init__()`
 - `Point.__init__()`
 - `Point.__init__()`
 - `Point.__init__()`
 - `Point.distance_to()`
 - `Point.norm()`
 - `Point.transform()`
 - `Point.unit`
 - `Point.abs_unit`
 - `Point.x`
 - `Point.y`

- [Quad](#)

- `Quad`
 - `Quad.__init__()`
 - `Quad.__init__()`
 - `Quad.__init__()`
 - `Quad.__init__()`
 - `Quad.transform()`
 - `Quad.morph()`

- `Quad.rect`
- `Quad.ul`
- `Quad.ur`
- `Quad.ll`
- `Quad.lr`
- `Quad.is_convex`
- `Quad.is_empty`
- `Quad.is_rectangular`
- `Quad.width`
- `Quad.height`
- Remark
- Algebra and Containment Checks

- Rect

- `Rect`
 - `Rect.__init__()`
 - `Rect.__init__()`
 - `Rect.__init__()`
 - `Rect.__init__()`
 - `Rect.__init__()`
 - `Rect.__init__()`
 - `Rect.__init__()`
 - `Rect.__init__()`
 - `Rect.round()`
 - `Rect.transform()`
 - `Rect.intersect()`
 - `Rect.include_rect()`
 - `Rect.include_point()`
 - `Rect.get_area()`
 - `Rect.contains()`
 - `Rect.intersects()`
 - `Rect.torect()`
 - `Rect.morph()`
 - `Rect.norm()`
 - `Rect.normalize()`
 - `Rect.irect`
 - `Rect.top_left`
 - `Rect.tl`
 - `Rect.top_right`
 - `Rect.tr`
 - `Rect.bottom_left`
 - `Rect.bl`

- `Rect.bottom_right`
- `Rect.br`
- `Rect.quad`
- `Rect.width`
- `Rect.height`
- `Rect.x0`
- `Rect.y0`
- `Rect.x1`
- `Rect.y1`
- `Rect.is_infinite`
- `Rect.is_empty`
- `Rect.is_valid`

◦ Shape

- `Shape`
 - `Shape.__init__()`
 - `Shape.draw_line()`
 - `Shape.draw_squiggle()`
 - `Shape.draw_zigzag()`
 - `Shape.draw_polyline()`
 - `Shape.draw_bezier()`
 - `Shape.draw_oval()`
 - `Shape.draw_circle()`
 - `Shape.draw_curve()`
 - `Shape.draw_sector()`
 - `Shape.draw_rect()`
 - `Shape.draw_quad()`
 - `Shape.finish()`
 - `Shape.insert_text()`
 - `Shape.insert_textbox()`
 - `Shape.commit()`
 - `Shape.doc`
 - `Shape.page`
 - `Shape.height`
 - `Shape.width`
 - `Shape.draw_cont`
 - `Shape.text_cont`
 - `Shape.rect`
 - `Shape.totalcont`
 - `Shape.lastPoint`

▪ Usage

- [Examples](#)
- [Common Parameters](#)

- [TextPage](#)

- `TextPage`
 - `TextPage.extractText()`
 - `TextPage.extractTEXT()`
 - `TextPage.extractBLOCKS()`
 - `TextPage.extractWORDS()`
 - `TextPage.extractHTML()`
 - `TextPage.extractDICT()`
 - `TextPage.extractJSON()`
 - `TextPage.extractXHTML()`
 - `TextPage.extractXML()`
 - `TextPage.extractRAW_DICT()`
 - `TextPage.extractRAWJSON()`
 - `TextPage.search()`
 - `TextPage.rect`
- [Structure of Dictionary Outputs](#)
 - [Page Dictionary](#)
 - [Block Dictionaries](#)
 - [Line Dictionary](#)
 - [Span Dictionary](#)
 - [Character Dictionary for `extractRAW_DICT\(\)`](#)

- [TextWriter](#)

- `TextWriter`
 - `TextWriter.__init__()`
 - `TextWriter.append()`
 - `TextWriter.appendv()`
 - `TextWriter.fill_textbox()`
 - `TextWriter.write_text()`
 - `TextWriter.text_rect`
 - `TextWriter.last_point`
 - `TextWriter.opacity`
 - `TextWriter.color`
 - `TextWriter.rect`

- [Tools](#)

- `Tools`

- `Tools.gen_id()`
- `Tools.set_annot_stem()`
- `Tools.set_small_glyph_heights()`
- `Tools.set_subset_fontnames()`
- `Tools.unset_quad_corrections()`
- `Tools.store_shrink()`
- `Tools.show_aa_level()`
- `Tools.set_aa_level()`
- `Tools.reset_mupdf_warnings()`
- `Tools.mupdf_display_errors()`
- `Tools.mupdf_warnings()`
- `Tools.fitz_config`
- `Tools.store_maxsize`
- `Tools.store_size`

- **Example Session**

- **Widget**

- `Widget`
 - `Widget.button_states()`
 - `Widget.update()`
 - `Widget.reset()`
 - `Widget.next`
 - `Widget.border_color`
 - `Widget.border_style`
 - `Widget.border_width`
 - `Widget.border_dashes`
 - `Widget.choice_values`
 - `Widget.field_name`
 - `Widget.field_label`
 - `Widget.field_value`
 - `Widget.field_flags`
 - `Widget.field_type`
 - `Widget.field_type_string`
 - `Widget.fill_color`
 - `Widget.button_caption`
 - `Widget.is_signed`
 - `Widget.rect`
 - `Widget.text_color`
 - `Widget.text_font`
 - `Widget.text_fontsize`
 - `Widget.text_maxlen`

- `Widget.text_type`
- `Widget.xref`
- `Widget.script`
- `Widget.script_stroke`
- `Widget.script_format`
- `Widget.script_change`
- `Widget.script_calc`

- [Standard Fonts for Widgets](#)
- [Supported Widget Types](#)

- [Operator Algebra for Geometry Objects](#)

- [General Remarks](#)
- [Unary Operations](#)
- [Binary Operations](#)
- [Some Examples](#)
 - [Manipulation with numbers](#)
 - [Manipulation with “like” Objects](#)

- [Low Level Functions and Classes](#)

- [Functions](#)

- `paper_size()`
- `paper_rect()`
- `sRGB_to_pdf()`
- `sRGB_to_rgb()`
- `glyph_name_to_unicode()`
- `unicode_to_glyph_name()`
- `adobe_glyph_names()`
- `adobe_glyph_unicodes()`
- `recover_quad()`
- `make_table()`
- `planish_line()`
- `paper_sizes()`
- `fitz_fontdescriptors`
- `TESSDATA_PREFIX`
- `pdfcolor`
- `get_pdf_now()`
- `get_text_length()`
- `get_pdf_str()`
- `image_profile()`
- `ConversionHeader()`
- `ConversionTrailer()`

- `Document.del_xml_metadata()`
- `Document.xml_metadata_xref()`
- `Page.run()`
- `Page.get_bboxlog()`
- `Page.get_texttrace()`
- `Page.wrap_contents()`
- `Page.is_wrapped`
- `Page.get_text_blocks()`
- `Page.get_text_words()`
- `Page.get_displaylist()`
- `Page.get_contents()`
- `Page.set_contents()`
- `Page.clean_contents()`
- `Page.read_contents()`
- `Annot.clean_contents()`
- `Document.get_char_widths()`
- `Document.is_stream()`
- `Document.get_new_xref()`
- `Document.xref_length()`
- `recover_quad()`
- `recover_char_quad()`
- `recover_span_quad()`
- `recover_line_quad()`
- `INFINITE_QUAD()`
- `INFINITE_RECT()`
- `INFINITE_IRECT()`
- `EMPTY_QUAD()`
- `EMPTY_RECT()`
- `EMPTY_IRECT()`

◦ Device

- `Device`
 - `Device.__init__()`
 - `Device.__init__()`

◦ Working together: DisplayList and TextPage

- Create a DisplayList
- Generate Pixmap
- Perform Text Search
- Extract Text
- Further Performance improvements

- [Pixmap](#)
- [TextPage](#)

- [Glossary](#)

- [matrix_like](#)
- [rect_like](#)
- [irect_like](#)
- [point_like](#)
- [quad_like](#)
- [inheritable](#)
- [MediaBox](#)
- [CropBox](#)
- [catalog](#)
- [trailer](#)
- [contents](#)
- [resources](#)
- [dictionary](#)
- [page](#)
- [pagetree](#)
- [object](#)
- [stream](#)
- [unitvector](#)
- [xref](#)
- [resolution](#)
- [OCPD](#)
- [OCCD](#)
- [OCG](#)
- [OCMD](#)
- [ligature](#)

- [Constants and Enumerations](#)

- [Constants](#)

- [Base14_Fonts](#)
- [csRGB](#)
- [csGRAY](#)
- [csCMYK](#)
- [CS_RGB](#)
- [CS_GRAY](#)
- [CS_CMYK](#)
- [VersionBind](#)
- [VersionFitz](#)

- `VersionDate`
- `version`
- Document Permissions
- PDF encryption method codes
- Font File Extensions
- Text Alignment

- `TEXT_ALIGN_LEFT`
- `TEXT_ALIGN_CENTER`
- `TEXT_ALIGN_RIGHT`
- `TEXT_ALIGN_JUSTIFY`

- Text Extraction Flags

- `TEXT_PRESERVE_LIGATURES`
- `TEXT_PRESERVE_WHITESPACE`
- `TEXT_PRESERVE_IMAGES`
- `TEXT_INHIBIT_SPACES`
- `TEXT_DEHYPHENATE`
- `TEXT_PRESERVE_SPANS`
- `TEXT_MEDIABOX_CLIP`
- `TEXTFLAGS_TEXT`
- `TEXTFLAGS_WORDS`
- `TEXTFLAGS_BLOCKS`
- `TEXTFLAGS_DICT`
- `TEXTFLAGS_RAWDICT`
- `TEXTFLAGS_HTML`
- `TEXTFLAGS_XHTML`
- `TEXTFLAGS_XML`
- `TEXTFLAGS_SEARCH`

- Link Destination Kinds

- `LINK_NONE`
- `LINK_GOTO`
- `LINK_URI`
- `LINK_LAUNCH`
- `LINK_NAMED`
- `LINK_GOTOR`

- Link Destination Flags

- `LINK_FLAG_L_VALID`
- `LINK_FLAG_T_VALID`
- `LINK_FLAG_R_VALID`

- `LINK_FLAG_B_VALID`
- `LINK_FLAG_FIT_H`
- `LINK_FLAG_FIT_V`
- `LINK_FLAG_R_IS_ZOOM`

- Annotation Related Constants

- Annotation Types
- Annotation Flag Bits
- Annotation Line Ending Styles

- Widget Constants

- Widget Types (*field_type*)
- Text Widget Subtypes (*text_format*)
- Widget flags (*field_flags*)

- PDF Standard Blend Modes

- Stamp Annotation Icons

- Color Database

- Function *getColor()*
- Printing the Color Database

- Appendix 1: Details on Text Extraction

- General structure of a TextPage
- Plain Text
- BLOCKS
- WORDS
- HTML
- Controlling Quality of HTML Output
- DICT (or JSON)
- RAWDICT (or RAWJSON)
- XML
- XHTML
- Text Extraction Flags Defaults
- Performance

- Appendix 2: Considerations on Embedded Files

- General
- MuPDF Support
- PyMuPDF Support

- Appendix 3: Assorted Technical Information

- Image Transformation Matrix
- PDF Base 14 Fonts
- Adobe PDF References

- [Using Python Sequences as Arguments in PyMuPDF](#)
- [Ensuring Consistency of Important Objects in PyMuPDF](#)
- [Design of Method](#) `Page.show_pdf_page()`
 - [Purpose and Capabilities](#)
 - [Technical Implementation](#)
- [Redirecting Error and Warning Messages](#)
- [Change Log](#)
- [Deprecated Names](#)