
depdf

Release 0.2.2

Jun 12, 2020

Contents:

1	depdf	1
2	depdf	3
2.1	depdf package	3
3	Indices and tables	17
	Python Module Index	19
	Index	21

An ultimate pdf file disintegration tool. DePDF is designed to extract tables and paragraphs into structured markup language [eg. html] from embedding pdf pages. You can also use it to convert page/pdf to html.

`depdf.convert_pdf_to_html(pdf, **kwargs)`

Parameters

- **pdf** – pdf file path
- **kwargs** – config keyword arguments

Returns pdf html string

`depdf.convert_page_to_html(pdf, pid, **kwargs)`

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page html string

`depdf.extract_page_tables(pdf, pid, **kwargs)`

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page tables list

`depdf.extract_page_paragraphs(pdf, pid, **kwargs)`

Parameters

- **pdf** – pdf file path

- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page paragraphs list

2.1 depdf package

2.1.1 Subpackages

depdf.components package

Submodules

depdf.components.image module

```
class depdf.components.image.Image (bbox=None, src="", percent=100, pid='1', img_idx=1,
                                     scan=False, config=None)
    Bases: depdf.base.Base, depdf.base.Box
    object_type = 'image'
```

depdf.components.paragraph module

```
class depdf.components.paragraph.Paragraph (bbox=None, text="", pid='1', para_idx=1,
                                              config=None, inner_objects=None,
                                              style=None, align=None)
    Bases: depdf.base.InnerWrapper, depdf.base.Box
    object_type = 'paragraph'
    save_html ()
```

depdf.components.span module

```
class depdf.components.span.Span (bbox=None, span_text="", config=None, style=None)
    Bases: depdf.base.Base, depdf.base.Box

    object_type = 'span'
```

depdf.components.table module

```
class depdf.components.table.Cell (bbox=None, text="", inner_objects=None)
    Bases: depdf.base.InnerWrapper, depdf.base.Box

    object_type = 'cell'

    to_dict
```

```
class depdf.components.table.Table (rows, pid='1', tid=1, config=None, bbox=None)
    Bases: depdf.base.Base, depdf.base.Box

    html

    object_type = 'table'

    save_html ()

    to_dict

    to_html
```

```
depdf.components.table.convert_table_to_html (table_dict, pid='1', tid=1, tc_mt=5,
                                                table_class='pdf-table', skip_et=False)
```

```
depdf.components.table.gen_column_cell_sizes (t)
```

depdf.components.text module

```
class depdf.components.text.Text (bbox="", text=")
    Bases: depdf.base.Base, depdf.base.Box

    object_type = 'text'
```

Module contents

```
class depdf.components.Paragraph (bbox=None, text="", pid='1', para_idx=1, config=None, inner_objects=None, style=None, align=None)
    Bases: depdf.base.InnerWrapper, depdf.base.Box

    object_type = 'paragraph'

    save_html ()
```

```
class depdf.components.Table (rows, pid='1', tid=1, config=None, bbox=None)
    Bases: depdf.base.Base, depdf.base.Box

    html

    object_type = 'table'

    save_html ()
```



```

    to_dict
    to_html

class depdf.components.Span(bbox=None, span_text="", config=None, style=None)
    Bases: depdf.base.Base, depdf.base.Box

    object_type = 'span'

class depdf.components.Text(bbox="", text="")
    Bases: depdf.base.Base, depdf.base.Box

    object_type = 'text'

class depdf.components.Cell(bbox=None, text="", inner_objects=None)
    Bases: depdf.base.InnerWrapper, depdf.base.Box

    object_type = 'cell'

    to_dict

class depdf.components.Image(bbox=None, src="", percent=100, pid='1', img_idx=1, scan=False,
                             config=None)
    Bases: depdf.base.Base, depdf.base.Box

    object_type = 'image'

```

2.1.2 Submodules

2.1.3 depdf.api module

```

depdf.api.api_load_pdf(api_func)
depdf.api.convert_page_to_html(pdf, pid, **kwargs)

```

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns

page html string

```
depdf.api.convert_pdf_to_html(pdf, **kwargs)
```

Parameters

- **pdf** – pdf file path
- **kwargs** – config keyword arguments

Returns

pdf html string

```
depdf.api.extract_page_paragraphs(pdf, pid, **kwargs)
```

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns

page paragraphs list

`depdf.api.extract_page_tables(pdf, pid, **kwargs)`

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page tables list

2.1.4 depdf.base module

```
class depdf.base.Base
    Bases: object

    html
    refresh()
    reset()
    soup
    to_dict
    to_soup(parser)
    write_to(file_name)

class depdf.base.Box
    Bases: object

    bbox
    bottom = Decimal('0')
    height
    static normalize_bbox(bbox)
    top = Decimal('0')
    width
    x0 = Decimal('0')
    x1 = Decimal('0')

class depdf.base.InnerWrapper
    Bases: depdf.base.Base

    inner_objects
    to_dict
```

2.1.5 depdf.config module

```
class depdf.config.Config(**kwargs)
    Bases: object

    add_horizontal_line_tolerance = Decimal('0.1')
    add_horizontal_lines_flag = False
```

```
add_line_flag = False
add_vertical_lines_flag = False
char_overlap_size = 3
char_size_lower = Decimal('3')
char_size_upper = Decimal('30')
column_region_half_width = 4
copy(**kwargs)
curved_line_flag = False
debug_flag = False
default_char_size = Decimal('12')
default_head_tail_page_offset_percent = 0.1
dotted_line_flag = True
header_footer_flag = True
image_class = 'pdf-image'
image_flag = True
image_resolution = 300
log_level = 30
logo_flag = True
main_frame_tolerance = None
max_columns = 3
max_double_line_tolerance = 3
min_column_region_objects = 1
min_double_line_tolerance = Decimal('0.05')
min_image_size = 80
mini_page_class = 'pdf-mini-page'
multiple_columns_flag = True
page_class = 'pdf-page'
page_num_left_fraction = Decimal('0.44')
page_num_right_fraction = Decimal('0.56')
page_num_top_fraction = Decimal('0.75')
paragraph_class = 'pdf-paragraph'
paragraph_flag = True
pdf_class = 'pdf-content'
resolution = 144
skip_empty_table = False
snap_flag = False
```

```

    span_class = 'pdf-span'
    table_cell_merge_tolerance = 5
    table_class = 'pdf-table'
    table_flag = True
    temp_dir_prefix = 'temp_depdf'
    to_dict
    unique_prefix = None
    update (**kwargs)
    verbose_flag = False
    vertical_double_line_tolerance = Decimal('2')
    x_tolerance = None
    y_tolerance = None
depdf.config.check_config(func)
depdf.config.check_config_type(config)

```

2.1.6 depdf.error module

```

exception depdf.error.BoxValueError(value)
    Bases: ValueError

exception depdf.error.ConfigTypeError(value)
    Bases: TypeError

exception depdf.error.PDFTypeError(value)
    Bases: TypeError

exception depdf.error.PageTypeError(value)
    Bases: TypeError

```

2.1.7 depdf.log module

```

depdf.log.logger_init(name)

```

2.1.8 depdf.page module

```

class depdf.page.DePage(page, pid='1', same=None, logo=None, config=None, columns=1,
                        mini=False)
    Bases: depdf.base.Base
    analyze_lines()
    analyze_main_frame()
    analyze_page_attributes()
    analyze_paragraph_border()
    ave_cs = 0
    ave_lh = 3

```

```
border = (0, 0, 0, 0)
chars
check_if_toc_page()
check_multi_column_page()
config
debug = False
extract_images()
extract_paragraph()
extract_phrases()
extract_tables()
frame_bottom = 0
frame_top = 0
h_edges = []
height
html
images
images_raw
min_cs = 0
new_para_end_flag = None
new_para_start_flag = None
object_key_list = ['_tables', '_paragraphs', '_images']
objects
orientation = ''
page
pagination_phrases = []
paragraphs
phrases = None
pid
prefix = UUID('599a28dd-1074-4438-8d7d-b54d15793514')
process_mini_page()
process_page()
refresh()
save_html()
screenshot
set_global()
tables
```

```

tables_raw
temp_dir = 'temp'
to_html
to_screenshot()
toc_flag = False
v_edges = []
verbose = False
width
x_tolerance = 3
y_tolerance = 3

class depdf.page.MiniDePage (page, pid='1', same=None, logo=None, config=None, columns=1,
                             mini=False)
    Bases: depdf.page.DePage
    save_html()
    to_html

depdf.page.check_page_type (page)
depdf.page.convert_plumber_table (pdf_page, table, pid='1', tid=1, config=None, min_cs=1)
depdf.page.extract_cell_region (cell_region, bbox, config=None, pid='1', tid=1, cid=1)

```

2.1.9 depdf.page_tools module

```

depdf.page_tools.add_horizontal_lines (v_lines, h_lines, vlts_tolerance=0.1)
depdf.page_tools.add_vertical_lines (v_lines, h_lines, page_rects, page, ave_cs)
depdf.page_tools.analyze_char_size (chars, char_size_upper=30, char_size_lower=3, de-
    fault_char_size=12)
depdf.page_tools.analyze_page_num_word (phrases, page_height, page_width,
    top_fraction=Decimal('0.7'),
    left_fraction=Decimal('0.4'),
    right_fraction=Decimal('0.6'))
depdf.page_tools.analyze_page_orientation (plumber_page)

```

Parameters `plumber_page` – pdfplumber.page.Page class

Returns

```

depdf.page_tools.calculate_paragraph_border (depdf_page_object)
depdf.page_tools.curve_to_lines (curves)
depdf.page_tools.edges_to_lines (edges)
depdf.page_tools.format_text (text)
depdf.page_tools.merge_page_figures (pdf_page, tables_raw=None, logo=None, min_width=3,
    min_height=3, pid='1')
depdf.page_tools.remove_duplicate_chars (chars, overlap_size=3)

```

```
depdf.page_tools.remove_single_lines (lines, max_double=3, min_double=0.05, vertical_double=2, m='h')
```

2.1.10 depdf.pdf module

```
class depdf.pdf.DePDF (pdf, config=None, **kwargs)
    Bases: depdf.base.Base

    close ()
    config
    extract_html_pages ()
    generate_pages ()
    get_prefix ()
    html
    html_pages
    classmethod load (file_name, config=None, **kwargs)
    logo
    classmethod open (*args, **kwargs)
    page_num
    pages
    pdf
    refresh ()
    same
    save_html ()
    to_html

depdf.pdf.check_pdf_type (pdf)
```

2.1.11 depdf.pdf_tools module

```
depdf.pdf_tools.check_page_orientation (pdf, pid)
```

Parameters

- **pdf** – pdfplumber class
- **pid** – page number starts from 0

Returns

```
depdf.pdf_tools.pdf_head_tail (pdf, config=None)
```

Parameters

- **pdf** – plumber pdf object
- **config** – depdf config class

Returns

 PDF

```
depdf.pdf_tools.pdf_logo(pdf)
```

2.1.12 depdf.settings module

2.1.13 depdf.utils module

```
depdf.utils.calc_bbox(objects)
depdf.utils.calc_overlap(a, b)
    :param a: [a_lower, a_upper] :param b: [b_lower, b_upper] :return: overlapping length
depdf.utils.construct_style(style=None)
depdf.utils.convert_html_to_soup(html, parser='html.parser')
depdf.utils.convert_soup_to_html(soup)
depdf.utils.repr_str(text, max_length=5)
```

2.1.14 depdf.version module

2.1.15 Module contents

depdf

An ultimate pdf file disintegration tool. DePDF is designed to extract tables and paragraphs into structured markup language [eg. html] from embedding pdf pages. You can also use it to convert page/pdf to html.

```
class depdf.Config(**kwargs)
    Bases: object

    add_horizontal_line_tolerance = Decimal('0.1')
    add_horizontal_lines_flag = False
    add_line_flag = False
    add_vertical_lines_flag = False
    char_overlap_size = 3
    char_size_lower = Decimal('3')
    char_size_upper = Decimal('30')
    column_region_half_width = 4
    copy(**kwargs)
    curved_line_flag = False
    debug_flag = False
    default_char_size = Decimal('12')
    default_head_tail_page_offset_percent = 0.1
    dotted_line_flag = True
    header_footer_flag = True
    image_class = 'pdf-image'
```



```

    image_flag = True
    image_resolution = 300
    log_level = 30
    logo_flag = True
    main_frame_tolerance = None
    max_columns = 3
    max_double_line_tolerance = 3
    min_column_region_objects = 1
    min_double_line_tolerance = Decimal('0.05')
    min_image_size = 80
    mini_page_class = 'pdf-mini-page'
    multiple_columns_flag = True
    page_class = 'pdf-page'
    page_num_left_fraction = Decimal('0.44')
    page_num_right_fraction = Decimal('0.56')
    page_num_top_fraction = Decimal('0.75')
    paragraph_class = 'pdf-paragraph'
    paragraph_flag = True
    pdf_class = 'pdf-content'
    resolution = 144
    skip_empty_table = False
    snap_flag = False
    span_class = 'pdf-span'
    table_cell_merge_tolerance = 5
    table_class = 'pdf-table'
    table_flag = True
    temp_dir_prefix = 'temp_depdf'
    to_dict
    unique_prefix = None
    update(**kwargs)
    verbose_flag = False
    vertical_double_line_tolerance = Decimal('2')
    x_tolerance = None
    y_tolerance = None
class depdf.DePDF(pdf, config=None, **kwargs)
    Bases: depdf.base.Base

```

```
close()
config
extract_html_pages()
generate_pages()
get_prefix()
html
html_pages
classmethod load(file_name, config=None, **kwargs)
logo
classmethod open(*args, **kwargs)
page_num
pages
pdf
refresh()
same
save_html()
to_html

class depdf.DePage(page, pid='1', same=None, logo=None, config=None, columns=1, mini=False)
    Bases: depdf.base.Base
    analyze_lines()
    analyze_main_frame()
    analyze_page_attributes()
    analyze_paragraph_border()
    ave_cs = 0
    ave_lh = 3
    border = (0, 0, 0, 0)
    chars
    check_if_toc_page()
    check_multi_column_page()
    config
    debug = False
    extract_images()
    extract_paragraph()
    extract_phrases()
    extract_tables()
    frame_bottom = 0
```

```

frame_top = 0
h_edges = []
height
html
images
images_raw
min_cs = 0
new_para_end_flag = None
new_para_start_flag = None
object_key_list = ['_tables', '_paragraphs', '_images']
objects
orientation = ''
page
pagination_phrases = []
paragraphs
phrases = None
pid
prefix = UUID('599a28dd-1074-4438-8d7d-b54d15793514')
process_mini_page()
process_page()
refresh()
save_html()
screenshot
set_global()
tables
tables_raw
temp_dir = 'temp'
to_html
to_screenshot()
toc_flag = False
v_edges = []
verbose = False
width
x_tolerance = 3
y_tolerance = 3
depdf.convert_pdf_to_html(pdf, **kwargs)

```

Parameters

- **pdf** – pdf file path
- **kwargs** – config keyword arguments

Returns pdf html string

`depdf.convert_page_to_html(pdf, pid, **kwargs)`

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page html string

`depdf.extract_page_tables(pdf, pid, **kwargs)`

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page tables list

`depdf.extract_page_paragraphs(pdf, pid, **kwargs)`

Parameters

- **pdf** – pdf file path
- **pid** – page number start from 1
- **kwargs** – config keyword arguments

Returns page paragraphs list

CHAPTER 3

Indices and tables

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

d

- `depdf, ??`
- `depdf.api, 5`
- `depdf.base, 6`
- `depdf.components, 4`
- `depdf.components.image, 3`
- `depdf.components.paragraph, 3`
- `depdf.components.span, 4`
- `depdf.components.table, 4`
- `depdf.components.text, 4`
- `depdf.config, 6`
- `depdf.error, 8`
- `depdf.log, 8`
- `depdf.page, 8`
- `depdf.page_tools, 10`
- `depdf.pdf, 11`
- `depdf.pdf_tools, 11`
- `depdf.settings, 12`
- `depdf.utils, 12`
- `depdf.version, 12`

A

`add_horizontal_line_tolerance` (de-
pdf.Config attribute), 12
`add_horizontal_line_tolerance` (de-
pdf.config.Config attribute), 6
`add_horizontal_lines()` (in module de-
pdf.page_tools), 10
`add_horizontal_lines_flag` (*depdf.Config at-*
tribute), 12
`add_horizontal_lines_flag` (de-
pdf.config.Config attribute), 6
`add_line_flag` (*depdf.Config attribute*), 12
`add_line_flag` (*depdf.config.Config attribute*), 6
`add_vertical_lines()` (in module de-
pdf.page_tools), 10
`add_vertical_lines_flag` (*depdf.Config at-*
tribute), 12
`add_vertical_lines_flag` (*depdf.config.Config*
attribute), 7
`analyze_char_size()` (in module de-
pdf.page_tools), 10
`analyze_lines()` (*depdf.DePage method*), 14
`analyze_lines()` (*depdf.page.DePage method*), 8
`analyze_main_frame()` (*depdf.DePage method*),
14
`analyze_main_frame()` (*depdf.page.DePage*
method), 8
`analyze_page_attributes()` (*depdf.DePage*
method), 14
`analyze_page_attributes()` (de-
pdf.page.DePage method), 8
`analyze_page_num_word()` (in module de-
pdf.page_tools), 10
`analyze_page_orientation()` (in module de-
pdf.page_tools), 10
`analyze_paragraph_border()` (*depdf.DePage*
method), 14
`analyze_paragraph_border()` (de-
pdf.page.DePage method), 8

`api_load_pdf()` (in module *depdf.api*), 5
`ave_cs` (*depdf.DePage attribute*), 14
`ave_cs` (*depdf.page.DePage attribute*), 8
`ave_lh` (*depdf.DePage attribute*), 14
`ave_lh` (*depdf.page.DePage attribute*), 8

B

`Base` (class in *depdf.base*), 6
`bbox` (*depdf.base.Box attribute*), 6
`border` (*depdf.DePage attribute*), 14
`border` (*depdf.page.DePage attribute*), 9
`bottom` (*depdf.base.Box attribute*), 6
`Box` (class in *depdf.base*), 6
`BoxValueError`, 8

C

`calc_bbox()` (in module *depdf.utils*), 12
`calc_overlap()` (in module *depdf.utils*), 12
`calculate_paragraph_border()` (in module de-
pdf.page_tools), 10
`Cell` (class in *depdf.components*), 5
`Cell` (class in *depdf.components.table*), 4
`char_overlap_size` (*depdf.Config attribute*), 12
`char_overlap_size` (*depdf.config.Config attribute*),
7
`char_size_lower` (*depdf.Config attribute*), 12
`char_size_lower` (*depdf.config.Config attribute*), 7
`char_size_upper` (*depdf.Config attribute*), 12
`char_size_upper` (*depdf.config.Config attribute*), 7
`chars` (*depdf.DePage attribute*), 14
`chars` (*depdf.page.DePage attribute*), 9
`check_config()` (in module *depdf.config*), 8
`check_config_type()` (in module *depdf.config*), 8
`check_if_toc_page()` (*depdf.DePage method*), 14
`check_if_toc_page()` (*depdf.page.DePage*
method), 9
`check_multi_column_page()` (*depdf.DePage*
method), 14
`check_multi_column_page()` (de-
pdf.page.DePage method), 9

check_page_orientation() (in module *depdf.pdf_tools*), 11
 check_page_type() (in module *depdf.page*), 10
 check_pdf_type() (in module *depdf.pdf*), 11
 close() (*depdf.DePDF* method), 13
 close() (*depdf.pdf.DePDF* method), 11
 column_region_half_width (*depdf.Config* attribute), 12
 column_region_half_width (*depdf.config.Config* attribute), 7
 Config (class in *depdf*), 12
 Config (class in *depdf.config*), 6
 config (*depdf.DePage* attribute), 14
 config (*depdf.DePDF* attribute), 14
 config (*depdf.page.DePage* attribute), 9
 config (*depdf.pdf.DePDF* attribute), 11
 ConfigTypeError, 8
 construct_style() (in module *depdf.utils*), 12
 convert_html_to_soup() (in module *depdf.utils*), 12
 convert_page_to_html() (in module *depdf*), 1, 16
 convert_page_to_html() (in module *depdf.api*), 5
 convert_pdf_to_html() (in module *depdf*), 1, 15
 convert_pdf_to_html() (in module *depdf.api*), 5
 convert_plumber_table() (in module *depdf.page*), 10
 convert_soup_to_html() (in module *depdf.utils*), 12
 convert_table_to_html() (in module *depdf.components.table*), 4
 copy() (*depdf.Config* method), 12
 copy() (*depdf.config.Config* method), 7
 curve_to_lines() (in module *depdf.page_tools*), 10
 curved_line_flag (*depdf.Config* attribute), 12
 curved_line_flag (*depdf.config.Config* attribute), 7

D

debug (*depdf.DePage* attribute), 14
 debug (*depdf.page.DePage* attribute), 9
 debug_flag (*depdf.Config* attribute), 12
 debug_flag (*depdf.config.Config* attribute), 7
 default_char_size (*depdf.Config* attribute), 12
 default_char_size (*depdf.config.Config* attribute), 7
 default_head_tail_page_offset_percent (*depdf.Config* attribute), 12
 default_head_tail_page_offset_percent (*depdf.config.Config* attribute), 7
 DePage (class in *depdf*), 14
 DePage (class in *depdf.page*), 8
 DePDF (class in *depdf*), 13
 DePDF (class in *depdf.pdf*), 11
 depdf (module), 1, 12
 depdf.api (module), 5
 depdf.base (module), 6
 depdf.components (module), 4
 depdf.components.image (module), 3
 depdf.components.paragraph (module), 3
 depdf.components.span (module), 4
 depdf.components.table (module), 4
 depdf.components.text (module), 4
 depdf.config (module), 6
 depdf.error (module), 8
 depdf.log (module), 8
 depdf.page (module), 8
 depdf.page_tools (module), 10
 depdf.pdf (module), 11
 depdf.pdf_tools (module), 11
 depdf.settings (module), 12
 depdf.utils (module), 12
 depdf.version (module), 12
 dotted_line_flag (*depdf.Config* attribute), 12
 dotted_line_flag (*depdf.config.Config* attribute), 7

E

edges_to_lines() (in module *depdf.page_tools*), 10
 extract_cell_region() (in module *depdf.page*), 10
 extract_html_pages() (*depdf.DePDF* method), 14
 extract_html_pages() (*depdf.pdf.DePDF* method), 11
 extract_images() (*depdf.DePage* method), 14
 extract_images() (*depdf.page.DePage* method), 9
 extract_page_paragraphs() (in module *depdf*), 1, 16
 extract_page_paragraphs() (in module *depdf.api*), 5
 extract_page_tables() (in module *depdf*), 1, 16
 extract_page_tables() (in module *depdf.api*), 5
 extract_paragraph() (*depdf.DePage* method), 14
 extract_paragraph() (*depdf.page.DePage* method), 9
 extract_phrases() (*depdf.DePage* method), 14
 extract_phrases() (*depdf.page.DePage* method), 9
 extract_tables() (*depdf.DePage* method), 14
 extract_tables() (*depdf.page.DePage* method), 9

F

format_text() (in module *depdf.page_tools*), 10
 frame_bottom (*depdf.DePage* attribute), 14
 frame_bottom (*depdf.page.DePage* attribute), 9
 frame_top (*depdf.DePage* attribute), 14
 frame_top (*depdf.page.DePage* attribute), 9

G

gen_column_cell_sizes() (in module *depdf.components.table*), 4

`generate_pages()` (*depdf.DePDF method*), 14
`generate_pages()` (*depdf.pdf.DePDF method*), 11
`get_prefix()` (*depdf.DePDF method*), 14
`get_prefix()` (*depdf.pdf.DePDF method*), 11

H

`h_edges` (*depdf.DePage attribute*), 15
`h_edges` (*depdf.page.DePage attribute*), 9
`header_footer_flag` (*depdf.Config attribute*), 12
`header_footer_flag` (*depdf.config.Config attribute*), 7
`height` (*depdf.base.Box attribute*), 6
`height` (*depdf.DePage attribute*), 15
`height` (*depdf.page.DePage attribute*), 9
`html` (*depdf.base.Base attribute*), 6
`html` (*depdf.components.Table attribute*), 4
`html` (*depdf.components.table.Table attribute*), 4
`html` (*depdf.DePage attribute*), 15
`html` (*depdf.DePDF attribute*), 14
`html` (*depdf.page.DePage attribute*), 9
`html` (*depdf.pdf.DePDF attribute*), 11
`html_pages` (*depdf.DePDF attribute*), 14
`html_pages` (*depdf.pdf.DePDF attribute*), 11

I

`Image` (*class in depdf.components*), 5
`Image` (*class in depdf.components.image*), 3
`image_class` (*depdf.Config attribute*), 12
`image_class` (*depdf.config.Config attribute*), 7
`image_flag` (*depdf.Config attribute*), 12
`image_flag` (*depdf.config.Config attribute*), 7
`image_resolution` (*depdf.Config attribute*), 13
`image_resolution` (*depdf.config.Config attribute*), 7
`images` (*depdf.DePage attribute*), 15
`images` (*depdf.page.DePage attribute*), 9
`images_raw` (*depdf.DePage attribute*), 15
`images_raw` (*depdf.page.DePage attribute*), 9
`inner_objects` (*depdf.base.InnerWrapper attribute*), 6
`InnerWrapper` (*class in depdf.base*), 6

L

`load()` (*depdf.DePDF class method*), 14
`load()` (*depdf.pdf.DePDF class method*), 11
`log_level` (*depdf.Config attribute*), 13
`log_level` (*depdf.config.Config attribute*), 7
`logger_init()` (*in module depdf.log*), 8
`logo` (*depdf.DePDF attribute*), 14
`logo` (*depdf.pdf.DePDF attribute*), 11
`logo_flag` (*depdf.Config attribute*), 13
`logo_flag` (*depdf.config.Config attribute*), 7

M

`main_frame_tolerance` (*depdf.Config attribute*), 13
`main_frame_tolerance` (*depdf.config.Config attribute*), 7
`max_columns` (*depdf.Config attribute*), 13
`max_columns` (*depdf.config.Config attribute*), 7
`max_double_line_tolerance` (*depdf.Config attribute*), 13
`max_double_line_tolerance` (*depdf.config.Config attribute*), 7
`merge_page_figures()` (*in module depdf.page_tools*), 10
`min_column_region_objects` (*depdf.Config attribute*), 13
`min_column_region_objects` (*depdf.config.Config attribute*), 7
`min_cs` (*depdf.DePage attribute*), 15
`min_cs` (*depdf.page.DePage attribute*), 9
`min_double_line_tolerance` (*depdf.Config attribute*), 13
`min_double_line_tolerance` (*depdf.config.Config attribute*), 7
`min_image_size` (*depdf.Config attribute*), 13
`min_image_size` (*depdf.config.Config attribute*), 7
`mini_page_class` (*depdf.Config attribute*), 13
`mini_page_class` (*depdf.config.Config attribute*), 7
`MiniDePage` (*class in depdf.page*), 10
`multiple_columns_flag` (*depdf.Config attribute*), 13
`multiple_columns_flag` (*depdf.config.Config attribute*), 7

N

`new_para_end_flag` (*depdf.DePage attribute*), 15
`new_para_end_flag` (*depdf.page.DePage attribute*), 9
`new_para_start_flag` (*depdf.DePage attribute*), 15
`new_para_start_flag` (*depdf.page.DePage attribute*), 9
`normalize_bbox()` (*depdf.base.Box static method*), 6

O

`object_key_list` (*depdf.DePage attribute*), 15
`object_key_list` (*depdf.page.DePage attribute*), 9
`object_type` (*depdf.components.Cell attribute*), 5
`object_type` (*depdf.components.Image attribute*), 5
`object_type` (*depdf.components.image.Image attribute*), 3
`object_type` (*depdf.components.Paragraph attribute*), 4
`object_type` (*depdf.components.paragraph.Paragraph attribute*), 3

object_type (*depdf.components.Span* attribute), 5
 object_type (*depdf.components.span.Span* attribute), 4
 object_type (*depdf.components.Table* attribute), 4
 object_type (*depdf.components.table.Cell* attribute), 4
 object_type (*depdf.components.table.Table* attribute), 4
 object_type (*depdf.components.Text* attribute), 5
 object_type (*depdf.components.text.Text* attribute), 4
 objects (*depdf.DePage* attribute), 15
 objects (*depdf.page.DePage* attribute), 9
 open () (*depdf.DePDF* class method), 14
 open () (*depdf.pdf.DePDF* class method), 11
 orientation (*depdf.DePage* attribute), 15
 orientation (*depdf.page.DePage* attribute), 9

P

page (*depdf.DePage* attribute), 15
 page (*depdf.page.DePage* attribute), 9
 page_class (*depdf.Config* attribute), 13
 page_class (*depdf.config.Config* attribute), 7
 page_num (*depdf.DePDF* attribute), 14
 page_num (*depdf.pdf.DePDF* attribute), 11
 page_num_left_fraction (*depdf.Config* attribute), 13
 page_num_left_fraction (*depdf.config.Config* attribute), 7
 page_num_right_fraction (*depdf.Config* attribute), 13
 page_num_right_fraction (*depdf.config.Config* attribute), 7
 page_num_top_fraction (*depdf.Config* attribute), 13
 page_num_top_fraction (*depdf.config.Config* attribute), 7
 pages (*depdf.DePDF* attribute), 14
 pages (*depdf.pdf.DePDF* attribute), 11
 PageTypeError, 8
 pagination_phrases (*depdf.DePage* attribute), 15
 pagination_phrases (*depdf.page.DePage* attribute), 9
 Paragraph (class in *depdf.components*), 4
 Paragraph (class in *depdf.components.paragraph*), 3
 paragraph_class (*depdf.Config* attribute), 13
 paragraph_class (*depdf.config.Config* attribute), 7
 paragraph_flag (*depdf.Config* attribute), 13
 paragraph_flag (*depdf.config.Config* attribute), 7
 paragraphs (*depdf.DePage* attribute), 15
 paragraphs (*depdf.page.DePage* attribute), 9
 pdf (*depdf.DePDF* attribute), 14
 pdf (*depdf.pdf.DePDF* attribute), 11
 pdf_class (*depdf.Config* attribute), 13
 pdf_class (*depdf.config.Config* attribute), 7

pdf_head_tail () (in module *depdf.pdf_tools*), 11
 pdf_logo () (in module *depdf.pdf_tools*), 11
 PDFTypeError, 8
 phrases (*depdf.DePage* attribute), 15
 phrases (*depdf.page.DePage* attribute), 9
 pid (*depdf.DePage* attribute), 15
 pid (*depdf.page.DePage* attribute), 9
 prefix (*depdf.DePage* attribute), 15
 prefix (*depdf.page.DePage* attribute), 9
 process_mini_page () (*depdf.DePage* method), 15
 process_mini_page () (*depdf.page.DePage* method), 9
 process_page () (*depdf.DePage* method), 15
 process_page () (*depdf.page.DePage* method), 9

R

refresh () (*depdf.base.Base* method), 6
 refresh () (*depdf.DePage* method), 15
 refresh () (*depdf.DePDF* method), 14
 refresh () (*depdf.page.DePage* method), 9
 refresh () (*depdf.pdf.DePDF* method), 11
 remove_duplicate_chars () (in module *depdf.page_tools*), 10
 remove_single_lines () (in module *depdf.page_tools*), 10
 repr_str () (in module *depdf.utils*), 12
 reset () (*depdf.base.Base* method), 6
 resolution (*depdf.Config* attribute), 13
 resolution (*depdf.config.Config* attribute), 7

S

same (*depdf.DePDF* attribute), 14
 same (*depdf.pdf.DePDF* attribute), 11
 save_html () (*depdf.components.Paragraph* method), 4
 save_html () (*depdf.components.paragraph.Paragraph* method), 3
 save_html () (*depdf.components.Table* method), 4
 save_html () (*depdf.components.table.Table* method), 4
 save_html () (*depdf.DePage* method), 15
 save_html () (*depdf.DePDF* method), 14
 save_html () (*depdf.page.DePage* method), 9
 save_html () (*depdf.page.MiniDePage* method), 10
 save_html () (*depdf.pdf.DePDF* method), 11
 screenshot (*depdf.DePage* attribute), 15
 screenshot (*depdf.page.DePage* attribute), 9
 set_global () (*depdf.DePage* method), 15
 set_global () (*depdf.page.DePage* method), 9
 skip_empty_table (*depdf.Config* attribute), 13
 skip_empty_table (*depdf.config.Config* attribute), 7
 snap_flag (*depdf.Config* attribute), 13
 snap_flag (*depdf.config.Config* attribute), 7
 soup (*depdf.base.Base* attribute), 6

Span (class in *depdf.components*), 5
 Span (class in *depdf.components.span*), 4
 span_class (*depdf.Config* attribute), 13
 span_class (*depdf.config.Config* attribute), 7

T

Table (class in *depdf.components*), 4
 Table (class in *depdf.components.table*), 4
 table_cell_merge_tolerance (*depdf.Config* attribute), 13
 table_cell_merge_tolerance (*depdf.config.Config* attribute), 8
 table_class (*depdf.Config* attribute), 13
 table_class (*depdf.config.Config* attribute), 8
 table_flag (*depdf.Config* attribute), 13
 table_flag (*depdf.config.Config* attribute), 8
 tables (*depdf.DePage* attribute), 15
 tables (*depdf.page.DePage* attribute), 9
 tables_raw (*depdf.DePage* attribute), 15
 tables_raw (*depdf.page.DePage* attribute), 9
 temp_dir (*depdf.DePage* attribute), 15
 temp_dir (*depdf.page.DePage* attribute), 10
 temp_dir_prefix (*depdf.Config* attribute), 13
 temp_dir_prefix (*depdf.config.Config* attribute), 8
 Text (class in *depdf.components*), 5
 Text (class in *depdf.components.text*), 4
 to_dict (*depdf.base.Base* attribute), 6
 to_dict (*depdf.base.InnerWrapper* attribute), 6
 to_dict (*depdf.components.Cell* attribute), 5
 to_dict (*depdf.components.Table* attribute), 4
 to_dict (*depdf.components.table.Cell* attribute), 4
 to_dict (*depdf.components.table.Table* attribute), 4
 to_dict (*depdf.Config* attribute), 13
 to_dict (*depdf.config.Config* attribute), 8
 to_html (*depdf.components.Table* attribute), 5
 to_html (*depdf.components.table.Table* attribute), 4
 to_html (*depdf.DePage* attribute), 15
 to_html (*depdf.DePDF* attribute), 14
 to_html (*depdf.page.DePage* attribute), 10
 to_html (*depdf.page.MiniDePage* attribute), 10
 to_html (*depdf.pdf.DePDF* attribute), 11
 to_screenshot () (*depdf.DePage* method), 15
 to_screenshot () (*depdf.page.DePage* method), 10
 to_soup () (*depdf.base.Base* method), 6
 toc_flag (*depdf.DePage* attribute), 15
 toc_flag (*depdf.page.DePage* attribute), 10
 top (*depdf.base.Box* attribute), 6

U

unique_prefix (*depdf.Config* attribute), 13
 unique_prefix (*depdf.config.Config* attribute), 8
 update () (*depdf.Config* method), 13
 update () (*depdf.config.Config* method), 8

V

v_edges (*depdf.DePage* attribute), 15
 v_edges (*depdf.page.DePage* attribute), 10
 verbose (*depdf.DePage* attribute), 15
 verbose (*depdf.page.DePage* attribute), 10
 verbose_flag (*depdf.Config* attribute), 13
 verbose_flag (*depdf.config.Config* attribute), 8
 vertical_double_line_tolerance (*depdf.Config* attribute), 13
 vertical_double_line_tolerance (*depdf.config.Config* attribute), 8

W

width (*depdf.base.Box* attribute), 6
 width (*depdf.DePage* attribute), 15
 width (*depdf.page.DePage* attribute), 10
 write_to () (*depdf.base.Base* method), 6

X

x0 (*depdf.base.Box* attribute), 6
 x1 (*depdf.base.Box* attribute), 6
 x_tolerance (*depdf.Config* attribute), 13
 x_tolerance (*depdf.config.Config* attribute), 8
 x_tolerance (*depdf.DePage* attribute), 15
 x_tolerance (*depdf.page.DePage* attribute), 10

Y

y_tolerance (*depdf.Config* attribute), 13
 y_tolerance (*depdf.config.Config* attribute), 8
 y_tolerance (*depdf.DePage* attribute), 15
 y_tolerance (*depdf.page.DePage* attribute), 10