

Features

Main features

- Accepts UTF-8 encoded HTML
- Supports almost all languages including **RTL** (arabic and hebrew), and **CJK** - (chinese-japanese-korean)
- Bookmarks
- CSS stylesheets
- Word spacing and character spacing for justification
- Nested block-level elements (e.g. P, DIV) including margins, borders, padding, line-height, background colours etc.
- Support (partial) for floating and fixed-position block-elements
- Page layout and orientation
- Text-justification and hyphenation
- Page numbering
- Odd and even paging with mirrored margins
- Page headers & footers
- Columns
- Tables - nested tables, rotated, or autosized to fit on a page
- Table of contents
- Index
- Watermarks
- Images in JPG, GIF, PNG or WMF format
- Password protection
- Annotations
- Barcodes (EAN13, UPC-A/E, Code 11, 39, 93, 128, Codabar, MSI, IMB, Planet, Postnet, RM4SCC etc.)
- Import another PDF file and use as a template
- Joined Arabic text - support for Arabic, Persian, Urdu, Sindhi and Pashto
- Support for Indic languages (incl. consonant conjuncts) - Gujarati, Devanagari, Bengali, Tamil, Malayalam, Oriya, Kannada, Telugu
- Embedded font subsets

More Information

mPDF has a number of enhancements over the original FPDF, HTML2FPDF and UPDF scripts:

UTF-8 encoded HTML is accepted as the standard input.

Almost all languages can be output:

- PDF files which include only characters included in a single codepage: windows-1252, iso-8859-2, iso-8859-4, windows-1251, iso-8859-7 (Greek) and iso-8859-9 (Turkish) can utilise small embedded fonts, producing small file size.
- Unicode fonts will output pages with mixed character sets, and include hebrew, arabic, vietnamese etc. Indic languages work partially. Much larger files are produced.
- CJK (chinese-japanese-korean) characters utilise the Adobe font packs which have to be downloaded by the user from Adobe for free. This keeps the file size small. CJK characters can be mixed with other Unicode characters in one file.

Right-to-left languages are supported, with automatic detection of RTL characters within a document. Transposes: tables, lists, text justification and table cell alignment, as well as full text reversal for RTL characters. Automatically detects non-RTL characters and displays these in original order.

Bookmarks and Meta tag information are supported in all character sets.

A single CSS stylesheet can be used for all pages, with font substitution automatically for CJK characters.

Character substitution will automatically replace any characters that exist in the **Zapfdingbats** or **Symbol**

character sets, and utilise the embedded Adobe fonts. This allows additional characters to be used whilst using small embedded fonts using codepages.

Word spacing and character spacing are both used to justify text; works in unicode mode and CJK characters as well.

Nested block-level elements (e.g. P, DIV) are supported, with comprehensive CSS support e.g. margins, borders, padding, line-height, background colours etc.

CSS style attributes now fully support font, font-size, color, and background color (for highlighting) plus many more.

Table cell padding and borders are supported.

Text-indent for 1st line of paragraph, and hanging indents are supported.

List indenting can be defined.

Custom tags added - PAGEBREAK, COLUMNBREAK, INDEXENTRY

Multiple columns can be started and stopped anywhere on the page with column height adjusted (and optionally aligned to justify).

Tables can be **rotated**, or **autosized** - font-size is reduced if required to fit the page. Background colour for TR rows is supported.

Odd and even paging can be used with inner and outer margins alternated.

A more complex definition of headers and footers allows left/center/right parts to be defined, each with their own font-styles, and including code to allow the date/time to be inserted as well as page numbers.

A table of contents can be generated automatically, which can be inserted at the front of the document, based on custom tags used throughout the HTML code. (Based on Richard Bondi's extension to FPDF.)

An Index can be generated at the end of the document based on custom tags used throughout the HTML code.

Non-breaking space is supported when using fonts that have a character representing chr(160) (not all of them).

A watermark can be used e.g. for 'DRAFT', and will appear as a transparency over other elements. (Based on extra script 'Transparency' by Martin Hall-May at FPDF.)

Intelligent word-wrapping will avoid 'orphan' punctuation or superscript moving to the next line.

Automatic hyphenation is optional.

All **HTML entities** are supported, as well as decimal and hex e.g. ' ≬ or ↤

Password protection can be set for the PDF file.

NB The original commands from FPDF can be used e.g. Write(), but some are altered to allow UTF-8 encoding and RTL text to be processed e.g. use WriteCell() and WriteMultiCell() instead of Cell() and MultiCell().

Printed on Wed 05 Aug 2015 12:36:54 GMT +0100 (DST)