Features of `pdfArticle` document class

Robert Ryszard Paciorek <rrp@opcode.eu.org>

2019-05-05

Document class `pdfArticle` is simple document class dedicated for creating pdf documents with LuaLaTeX. This class:

- Use `extarticle` (from `extsizes`) as base document class (for support wide range of base font sizes).
- Require and configure `fontspec` package, that enable support for Unicode Open Type fonts. Configure default document sherif, sans and `mono` fonts via `fontspec` according to class options. By default use *Latin Modern* fonts with support for SMALL CAPS in sherif and mono fonts.
- Require and configure `graphicx` and `graphbox` packages (for including graphics). Set file extensions for search graphics files to `.mp, .pdf, .png, .jpg, .jpeg` (in this order – first METAPOST, last JPEG) and default image resolution to `imgResolution` class atribute (default 150).
- Require `xcolor` (for foreground and background colors) and `fifo-stack` packages and configure colours stack (you can use it with `\FSPush{colors}`, `\FSPop{colors}` and `\FSTop{colors}` commands).
- Require `hyperref` package (for internal and external links in pdfs) and configure color external (url) link as blue and internal (`\ref{}`) links as color from top of colours stack. More setting (eg. set PDF title, author, etc) can be do via `\hypersetup{}` command.
- Require `geometry` package and configure page gemetry (paper size, orientation and marings) according to class options. By default use portrait A4 paper.
- Provide `\forceNewPageGeometry` commad for enforce new page geometry (like `\newgeometry`, but allow change paper size too).
- Require `fancyvrb` and `fvextra` (for good verbatim envirомet with line breaking, line numbering, titles, frames, etc). Redefine standard `verbatim` enviromet and `\verb` commad to using fancyvrb. Allow break long lines on spaces, `/` or `-`, set ↪ as post break marker.
- Require `ulem`, `contour` and `shadowtext` (for text decorations).
- Is incompatible with `sout` package, because define own `\ul[color]{text}`, `\st[color]{text}` and `\hl[color]{text}` commands (based on `ulem` package) for underline, ~~strike out~~ and highlight text with selected color.
- Require `enumitem` (for better enumerate, itemize and description environments).
- Redefine `\alph` and `\Alph` with `alphalph` package (for converting big numbers to letters as `a, b, ..., x, y, z, aa, ab, ac, ..., az, ba, ..., zz, aaa, ...`).
- Require `pbox` and `varwidth` (for vertical box with automatic minimal width).
- Require `overpic` (for putting (La)TeX stuff on images).
- Require `wrapfig` (for wraping text around images).
- Require `array` and `dcolumn` (for useful extentions for tables).
- Require `tabto` (for tabbing to fixed positions).
- Require `ragged2e` (for `justify` environment and configurable `Center`, `FlushLeft`, `FlushRight`).
- Require `changepage` (for changes margins via `adjustwidth` environment).
- Require `setspace` (for setting line stretch – global and via `spacing` environment).
- Require `amsmath` and `unicode-math` (for better math with Unicode Open Type fonts).
- Require `adjustbox` (for scaling, rotating, clipping, etc boxes).
- Only when `extra` option was given require: `minted` (code highlight), `tcolorbox` (nice framed boxes).

Class can be simple used by: `\documentclass{pdfArticle}`. Below is example of class usage with all avalible options, all options in this example are set to default values:

```
\documentclass[
        fontSize=12pt,

        mainFont={Latin Modern Roman},
        mainFontFeat={
                UprightFeatures = { SmallCapsFont={ lmromancaps10regular } },
                ItalicFeatures  = { SmallCapsFont={ lmromancaps10oblique } },
                SlantedFont     = lmromanslant10regular,
                BoldSlantedFont = lmromanslant10bold,
                Ligatures=TeX
        },
        sansFont={Latin Modern Sans},
        sansFontFeat={
                Ligatures=TeX
        },
        monoFont={Latin Modern Mono},
        monoFontFeat={
                UprightFeatures = { SmallCapsFont={ lmmonocaps10regular } },
                ItalicFont      = lmmono10italic,
                ItalicFeatures  = { SmallCapsFont={ lmmonocaps10oblique } },
                SlantedFont     = lmmonoslant10regular
        },

        paperSize=a4paper, paperMode=portrait, twoside=true,
        tmargin=2.2cm, bmargin=2.5cm, lmargin=2.2cm, rmargin=2.2cm,

        imgResolution=150, extra=false
]{pdfArticle}
```

It's also possible load some packages (or do other things) right before load `hyperref` package (after load all other packages) via `\pdfArticlePreHyperRef`:

```
\newcommand\pdfArticlePreHyperRef{
% stuff to do before load hyperref
}
\documentclass{pdfArticle}
```