pisa

HTML/CSS to PDF converter

dirk.holtwick@gmail.com
http://www.htmltopdf.org

Introduction

XXX TO BE WRITTEN

pisa is a HTML/CSS to PDF converter written in Python and based on Reportlab Toolkit, pyPDF, TechGame Networks CSS Library and HTML5lib. The primary focus is not on generating perfect printable webpages, but to use HTML and CSS as commonly known tools to generate PDF files within Applications. For example generating documentation (like this one), generating invoices or other office documents and so on.

Command line

If you do not want to integrate **pisa** in your own application, you may use the command line tool that gives you a simple interface to the features of **pisa**.

XXX See README.txt for now

Python module

XXX TO BE WRITTEN

See "test/simply.py" for now

Defaults

Some notes on some default values:

- Usually the position (0,0) in PDF files is found in the lower left corner. For **pisa** it is the upper left corner like it is for HTML.
- The default page size is DIN A4 with portrait orientation.
- The name of the first layout template is "body" (XXX May be changed!)

Cascading Style Sheets

pisa supports a lot of Cascading Style Sheet (CSS). The following styles are supported:

```
background-color
border-bottom-color
border-bottom-style
border-bottom-width
border-left-color
border-left-style
border-left-width
border-right-color
border-right-style
border-right-width
border-top-color
border-top-style
border-top-width
color
display
font-family
font-size
font-style
font-weight
height
line-height
list-style-type
margin-bottom
margin-left
margin-right
margin-top
padding-bottom
padding-left
padding-right
padding-top
page-break-after
page-break-before
size
text-align
text-decoration
text-indent
vertical-align
white-space
width
zoom
```

And it adds some vendor specific styles:

```
-pdf-frame-border
-pdf-frame-break
-pdf-frame-content
-pdf-keep-with-next
-pdf-next-page
-pdf-outline
-pdf-outline-level
-pdf-outline-open
-pdf-page-break
```

Layout Definition

Pages and Frames

Pages can be layouted by using some special CSS at-keywords and properties. All special properties start with `-pdf-` to mark them as vendor specific as defined by CSS 2.1. Layouts may be defined by page using the `@page` keyword. Then text flows in one or more frames which can be defined within the @page block by using `@frame`. Example:

```
@page {
  @frame {
    margin: 1cm;
  }
}
```

In the example we define an unnamed page template - though it will be used as the default template - having one frame with 1cm margin to the page borders. The first frame of the page may also be defined within the `@page` block itself. See the equivalent example:

```
@page {
  margin: 1cm;
}
```

To define more frames just add some more `@frame` blocks. You may use the following properties to define the dimensions of the frame:

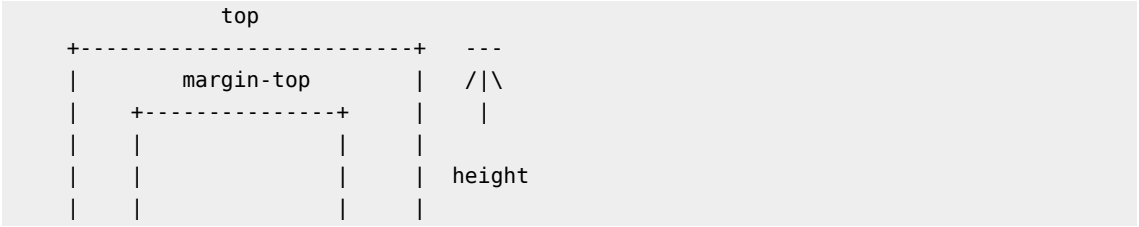
- marign
- margin-top
- margin-left
- margin-right
- margin-bottom
- top
- left
- right
- bottom
- width
- height

Here is a more complex example:

```
@page lastPage {
  top: 1cm;
  left: 2cm;
  right: 2cm;
  height: 2cm;
  @frame middle {
    margin: 3cm;
  }
  @frame footer {
    bottom: 2cm;
    margin-left: 1cm;
    margin-right: 1cm;
```

```
    height: 1cm;
  }
}
```

Layout scheme:



By default the Frame uses the whole page and is defined to begin in the upper left corner and end in the lower right corner. Now you can add the position of the frame using "top", "left", "bottom" and "right". If you now add "height" and you have a value other than zero in "top" the "bottom" will be modified. If you had not defined "top" but "bottom" the "height" will be

Page size and orientation

A page layout may also define the page size and the orientation of the paper using the "size" property as defined in CSS 3. Here is an example defining page size "DIN A5" with "landscape" orientation (default orientation is "portrait"):

```
@page {
  size: a5 landscape;
  margin: 1cm;
}
```

Here is a list of valid page sizes:

- a0 ... a6
- b0 ... b6
- letter
- legal
- elevenseventeen

PDF watermark/ background

For the use of PDF backgrounds specify the source file in the "background-image" property, like this:

```
@page {
  background-image: url(bg.pdf);
}
```

Static frames

Some frames should be static like headers and footers that means they are on every page but do not change content. The only information that may change is the page number. Here is a simple example that show how to make an element named by ID the content of a static frame.

In this case it is the ID "footer".

```
<html>
<style>
@page {
  margin: 1cm;
  margin-bottom: 2.5cm;
  @frame footer {
    -pdf-frame-content: footerContent;
    bottom: 2cm;
    margin-left: 1cm;
    margin-right: 1cm;
    height: 1cm;
  }
}
</style>
<body>
  Some text
  <div id="footerContent">
    This is a footer on page #<pdf:pagenumber>
  </div>
</body>
</html>
```

For better debugging you may want to add this property for each frame definition: "-pdf-frame-border: 1;". It will paint a border around the frame.

Fonts

By default there is just a certain set of fonts available for PDF. Here is the complete list - and their repective alias names - **pisa** knows by default (the names are not case sensitive):

- **Times-Roman**: Times New Roman, Times, Georgia, serif
- **Helvetica**: Arial, Verdana, Geneva, sansserif, sans
- **Courier**: Courier New, monospace, monospaced, mono
- **ZapfDingbats**
- **Symbol**

But you may also embed new font faces by using the `@font-face` keyword in CSS. Here is how:

```
@font-face {
  font-family: Example, "Example Font";
  src: url(example.ttf);
}
```

The "`font-family`" property defines the names under which the embedded font will be known. "src" defines the place of the fonts source file. This can be a TrueType font or a Postscript font. The file name of the first has to end with ".ttf" the later with one of ".pfb" or ".afm" (the missing file name will be calculated like this: "test" + ".afm" and "test" + ".pfb").

To define other shapes you may do like this:

```
/* Normal */
@font-face {
   font-family: DejaMono;
   src: url(font/DejaVuSansMono.ttf);
}

/* Bold */
@font-face {
   font-family: DejaMono;
   src: url(font/DejaVuSansMono-Bold.ttf);
   font-weight: bold;
}

/* Italic */
@font-face {
   font-family: DejaMono;
   src: url(font/DejaVuSansMono-Oblique.ttf);
   font-style: italic;
}

/* Bold and italic */
@font-face {
   font-family: DejaMono;
   src: url(font/DejaVuSansMono-BoldOblique.ttf);
   font-weight: bold;
   font-style: italic;
}
```

Outlines/ Bookmarks

PDF supports outlines (Adobe calls them "bookmarks). By default **pisa** defines the <H1> to <H6> tags to be shown in the outline. But you can specify exactly for every tag which outline behaviour it should have. Therefore you may want to use the following vendor specific styles:

- **-pdf-outline**
  set it to "true" if the block element should appear in the outline
- **-pdf-outline-level**
  set the value starting with "0" for the level on which the outline should appear. Missing predecessors are inserted automatically with the same name as the current outline
- **-pdf-outline-open**
  set to "true" if the outline should be shown uncollapsed

Example:

```
h1 {
  -pdf-outline: true;
  -pdf-level: 0;
  -pdf-open: false;
}
```

Table of Content

XXX <pdf:toc>

Barcodes

XXX <pdf:barcode>

Custom Tags

XXX TO BE WRITTEN

pdf:pagenumber

Prints current page number. The argument "example" defines the space the page number will require e.g. "00".

pdf:template

Defines the boxes resp. frames of a page in which the text flows. If a frame is full the next frame is used for the text. And if the last frame of the page is full a new page is opened and there we start again with the first frame. So the order of the frame definition is important! Frames that always contain the same text like e.g. title the of a book repeated on each page, are called "static".

The template of the first page is called "default" and contains in the standard definition one frame with 1cm margins. This template definiton can be overridden to change the layout of the first page.

The attribute "background" defines a PDF file that is set into the background of the current page. If this doesn't work it often helps to re-print the watermark PDF with Ghostscript again. If you use "background" the outlines are lost (q.v. h1).

Q.v. frame and static.

An example:

```
<template name="default">
  <frame name="head"
         box="17,3cm 2.8cm 3.3cm 20cm">
  <frame name="address"
         box="2.7cm 5.0cm 8.57cm 4.00cm">
  <frame name="body"
         box="3.5cm 10.5cm 13.9cm 17.5cm">
</template>

<template name="sub">
   <static box="1cm 1cm 13,9cm 1cm">
       The big book, page <pagenumber>.
   </static>
   <frame name="c"
           box="3.5cm 3.5cm 13.9cm 17.5cm">
</template>
```

pdf:frame

Q.v. template.

pdf:static

Q.v. template.

pdf:nexttemplate

Defines the template to be used on the next page.

pdf:nextpage

Create a new page after this position.

pdf:nextframe

Jump to next unused frame on the same page or to the first on a new page. You may not jump to a named frame.

~~pdf:effect~~

XXX

pdf:font

Depercated: please use CSS @font-face

pdf:spacer

Creates an object of a specific size.

~~pdf:version~~

Prints the version number of current **pisa**.

~~pdf:keeptogether~~

Tries to keep the block in the same frame. Example:

```
<keeptogether>
  <h1>Überschrift</h1>
  <p>Text</p>
</keeptogether>
```

~~drawline~~

Only useable within <static>. Draws a line. Example:

```
<drawline from="1cm 1cm" to="-1cm -1cm">
```

~~drawlines~~

Only useable within <static>. Draws multiple lines. Example:

```
<drawlines coords="1cm 1cm 1cm -1cm -1cm -1cm -1cm 1cm 1cm 1cm">
```

~~drawimg~~

Only useable within <static>. Draws an image at a position. May be used for simple backgrounds and watermarks.

~~drawpoint~~

Only useable within <static>. Draws a point.

~~drawstring~~

Only useable within <static>.Draws text at a certain position.

~~pdf:keepinframe~~

Tries to keep block in one frame. This is very usefull for usage in tables, because big table cells may cause errors in ReportLab toolkit. Then you better put a <keepinframe> around data to avoid these exceptions and get all informations of the table shown. Mode can be one of: "error", raise an error in the normal way; "overflow", ignore ie just draw it and report maxWidth, maxHeight; "shrink", shrinkToFit (is the default); "truncate", fit as much as possible. Example:

```
<table><tr><td>
  <keepinframe maxwidth="10cm" maxheight="10cm">
    A lot of ... data
  </keepinframe>
</td></tr></table>
```

Legal notice

Q Public License v1.0

The intent of this license is to establish freedom to share and change the software regulated by this license under the open source model.

This license applies to any software containing a notice placed by the copyright holder saying that it may be distributed under the terms of the Q Public License version 1.0. Such software is herein referred to as the Software. This license covers modification and distribution of the Software, use of third-party application programs based on the Software, and development of free software which uses the Software.

Granted Rights

1. You are granted the non-exclusive rights set forth in this license provided you agree to and comply with any and all conditions in this license. Whole or partial distribution of the Software, or software items that link with the Software, in any form signifies acceptance of this license.

2. You may copy and distribute the Software in unmodified form provided that the entire package, including - but not restricted to - copyright, trademark notices and disclaimers, as released by the initial developer of the Software, is distributed.

3. You may make modifications to the Software and distribute your modifications, in a form that is separate from the Software, such as patches. The following restrictions apply to modifications:

a. Modifications must not alter or remove any copyright notices in the Software.

b. When modifications to the Software are released under this license, a non-exclusive royalty-free right is granted to the initial developer of the Software to distribute your modification in future versions of the Software provided such versions remain available under these terms in addition to any other license(s) of the initial developer.

4. You may distribute machine-executable forms of the Software or machine-executable forms of modified versions of the Software, provided that you meet these restrictions:

a. You must include this license document in the distribution.

b. You must ensure that all recipients of the machine-executable forms are also able to receive the complete machine-readable source code to the distributed Software, including all modifications, without any charge beyond the costs of data transfer, and place prominent notices in the distribution explaining this.

c. You must ensure that all modifications included in the machine-executable forms are available under the terms of this license.

5. You may use the original or modified versions of the Software to compile, link and run application programs legally developed by you or by others.

6. You may develop application programs, reusable components and other software items that link with the original or modified versions of the Software. These items, when distributed, are subject to the following requirements:

a. You must ensure that all recipients of machine-executable forms of these items are also able to receive and use the complete machine-readable source code to the items without any charge beyond the costs of data transfer.

b. You must explicitly license all recipients of your items to use and re-distribute original and modified versions of the items in both machine-executable and source code forms. The recipients must be able to do so without any charges whatsoever, and they must be able to re-distribute to anyone they choose.

c. If the items are not available to the general public, and the initial developer of the Software requests a copy of the items, then you must supply one.

Limitations of Liability

In no event shall the initial developers or copyright holders be liable for any damages whatsoever, including - but not restricted to - lost revenue or profits or other direct, indirect, special, incidental or consequential damages, even if they have been advised of the possibility of such damages, except to the extent invariable law, if any, provides otherwise.

No Warranty

The Software and this license document are provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Choice of Law

This license is governed by the Laws of Norway. Disputes shall be settled by Oslo City Court.