

mPDF

Example 58. New mPDF v5-4 features

```
<?php

ini_set("memory_limit","64M");

include("../mpdf.php");

$mpdf=new mPDF('');


//=====

$html = '
<style>
.gradient {
    border:0.1mm solid #220044;
    background-color: #f0f2ff;
    background-gradient: linear #c7cdde #f0f2ff 0 1 0 0.5;
    box-shadow: 0.3em 0.3em #888888;
}
.rounded {
    border:0.1mm solid #220044;
    background-color: #f0f2ff;
    background-gradient: linear #c7cdde #f0f2ff 0 1 0 0.5;
    border-radius: 2mm;
    background-clip: border-box;
}
h4 {
    font-weight: bold;
    margin-top: 1em;
    margin-bottom: 0.3em;
    margin-top: 0;
}
div.text {
    padding:0.8em;
    margin-bottom: 0.7em;
}
p { margin: 0.25em 0; }
.code {
    font-family: monospace;
    font-size: 9pt;
    background-color: #d5d5d5;
    margin: 1em 1cm;
    padding: 0 0.3cm;
    border:0.2mm solid #000088;
    box-shadow: 0.3em 0.3em #888888;
}
table {
    overflow: visible;
    empty-cells: hide;
    border:1px solid #000000;
    font-family: sans-serif;
    font-size: 10pt;
    background-gradient: linear #c7cdde #f0f2ff 0 1 0 0.5;
}
td, th {
    border:1px solid #000000;
    text-align: left;
    font-weight: normal;
}
td.markedcell {
    text-decoration: line-through;
```

```

        color: #CC0000;
    }
    td.underlinedcell {
        text-decoration: underline;
        color: #CC0000;
    }
    td.rotatedcell {
        text-decoration: line-through;
        color: #CC0000;
        text-rotate: 45;
    }
    td.cost { text-align: right; }
    caption.tablecaption {
        font-family: sans-serif;
        font-weight: bold;
        border: none;
        caption-side: top;
        margin-bottom: 0;
        text-align: center;
    }
    u.doubleu {
        text-decoration: none;
        border-bottom: 3px double #000088;
    }
    a.reddashed {
        text-decoration: none;
        border: 1px dashed #880000;
    }
    .shadowtitle {
        height: 8mm;
        background-color: #EEDDFF;
        background-gradient: linear #c7cdde #f0f2ff 0 1 0 0.5;
        padding: 0.8em;
        padding-left: 3em;
        font-family:sans;
        font-size: 26pt;
        font-weight: bold;
        border: 0.2mm solid white;
        border-radius: 0.2em;
        box-shadow: 0 0 2em 0.5em rgba(0,0,255,0.9);
        color: #AAAACC;
        text-shadow: 0.03em 0.03em #666, 0.05em 0.05em rgba(127,127,127,0.5), -0.015em -0.015em white;
    }
    h3 {
        margin: 3em 0 2em -15mm;
        background-color: #EEDDFF;
        background-gradient: linear #c7cdde #f0f2ff 0 1 0 0.5;
        padding: 0.5em;
        padding-left: 3em;
        width: 50%;
        font-family:sans;
        font-size: 16pt;
        font-weight: bold;
        border-left: none;
        border-radius: 0 2em 2em 0;
        box-shadow: 0 0 2em 0.5em rgba(255,0,0,1);
        text-shadow: 0.05em 0.04em rgba(127,127,127,0.5);
    }
    .css {
        font-family: arial;
        font-style: italic;
        color: #000088;
    }
    table.zebra tbody tr:nth-child(2n+1) td { background-color: rgba(255,255,127,0.6); }
    table.zebra tbody tr:nth-child(2n+1) th { background-color: rgba(255,255,127,0.6); }
    table.zebra thead tr { background-color: #FFBBFF; }
    table.zebra tfoot tr { background-color: #BBFFFF; }

</style>

```

<body>

<div class="shadowtitle">New Features in mPDF v5.4</div>

<h3>Bookmark styles<bookmark content="Bookmark styles" level="0" /></h3>

<div>

<p>Bookmarks can be styled by adding code as below to your script. You can define a colour (array of RGB) and/or a font-style (B, I, or BI) for each level (starting at 0). Results may depend on the PDF Reader you are using.</p>

<p class="code">

```
$this->bookmarkStyles = array(<br />
    &nbsp; &nbsp; &nbsp; &nbsp; 0 => array('\color\'=> array(0,64,128), '\style\'=>\'B\'),<br />
    &nbsp; &nbsp; &nbsp; &nbsp; 1 => array('\color\'=> array(128,0,0), '\style\'=>\'I\'),<br />
    &nbsp; &nbsp; &nbsp; &nbsp; 2 => array('\color\'=> array(0,128,0), '\style\'=>\'I\'),<br />
);
</p>
</div>
```

<h3>Embedded SVG code<bookmark content="Embedded SVG code" level="0" /></h3>

<p>SVG Images can be embedded in your HTML code. This is formally part of the XHTML specification and is supported by IE9+ and most other browsers.</p>

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" viewBox="0 0 400 200" width="400" height="200">
    <circle cx="130" cy="100" r="80" stroke="black" stroke-width="1" fill="red" />
    <circle cx="200" cy="100" r="80" stroke="black" stroke-width="1" fill="blue" />
</svg>
<p class="code">
<code><svg xmlns="http://www.w3.org/2000/svg" version="1.1" viewBox="0 0 200 200" width="200"
height="200"&gt; <br />
&nbsp; &nbsp; <code><circle cx="100" cy="50" r="40" stroke="black" stroke-width="1" fill="red" /&gt;<br />
&nbsp; &nbsp; <code><circle cx="130" cy="50" r="40" stroke="black" stroke-width="1" fill="blue" /&gt;<br />
&nbsp; </code></code>
</p>
```

<h3>Improved CSS support<bookmark content="Improved CSS support" level="0" /></h3>

<div class="gradient text">

<p>border-radius is now supported on fixed/absolute-positioned block elements.</p>

<p>visibility can be set on any block-style element e.g. DIV,P or images IMG as: visible|hidden|printonly|screenonly</p>

<p>background-color now supports rgba() and cmyka() alpha transparency formats in tables.</p>

<p>Color, underline and strike-through are now supported in table cells, including with rotated text (see example below of spread table)</p>

<p>page-break-after: left|right|always is supported on all block-style elements and tables</p>

<p>text-transform: capitalize|uppercase|lowercase is supported in table cells</p>

</div>

<div class="gradient text">

<h4>Zebra stripes in Tables<bookmark content="Zebra stripes" level="1" /></h4>

<div>

<p>:nth-child() selector can be used in tables (on TR, TD or TH) to stripe rows or columns. Both the <i>a</i>n+<i>b</i> and odd/even forms are supported e.g.</p>

<p class="code">

```
tr:nth-child(2n+1) { background-color: rgba(255,255,127,0.6); } <i>or</i><br />
tr:nth-child(odd) { background-color: rgba(255,255,127,0.6); }
</p>
```

<table class="zebra" align="center">

<tbody>

<tr>

<th>Row 1</th>

<td>This is data</td>

<td class="cost">167.00</td>

```

</tr>
<tr>
<th>Row 2</th>
<td>
<p>This is data p</p>
</td>
<td class="cost">
<p>444.53</p>
</td>
</tr>
<tr>
<th>
<p>Row 3</p>
</th>
<td>
<p>This is long data</p>
</td>
<td class="cost">14.00</td>
</tr>
<tr>
<td>
<p>Row 4</p>
</td>
<td>This is data</td>
<td class="cost">
<p>0.88</p>
</td>
</tr>
<tr>
<td>Row 5</td>
<td>Also data</td>
<td class="cost">144.00</td>
</tr>
<tr>
<td>Row 6</td>
<td>Also data</td>
<td class="cost">8089.00</td>
</tr>
</tbody></table>

```

<p>Note: mPDF does NOT correctly apply specificity to all CSS. The following stylesheet:</p>

```

<p class="code">
table.zebra tbody tr:nth-child(2n+1) td { background-color: yellow; }<br />
table.zebra tbody td:nth-child(odd) { background-color: blue; }
</p>

```

<p>should make every odd row yellow, and every odd column blue, but with the row/yellow overriding the column/blue.

In mPDF the td:nth-child(odd) trumps the plain td, so the column colour wins out. You can force the effect you want by using:</p>

```

<p class="code">
table.zebra tbody tr:nth-child(2n+1) td:nth-child(1n+0) { background-color: yellow; }
</p>
<p>The :nth-child(1n+0) selector just selects every td cell.</p>

```

</div>

</div>

<div class="gradient text">

<p>border can now be defined on in-line elements eg SPAN</p>

```

<ul><li style="font-family: arial;">Cum sociis natoque <u class="doubleu">penatibus</u> et <a
class="reddashed" href="#">magnis dis parturient</a> montes</li></ul>

```

<p>Note: Remember that in mPDF, inside table cells, properties set on block elements are set when possible as in-line properties - so a P element inside a table with border set, will appear with a border around the text line as though it had been set on SPAN </p>

</div>

```
<div class="gradient text">
<h4>Shadows<bookmark content="Shadows" level="1" /></h4>
<p><span class="css">box-shadow</span> can be defined on any block-level element (P, DIV etc). It follows the CSS3 recommendation, but <i>inset</i> is not supported.</p>
<p><span class="css">text-shadow</span> can be defined on any element. It follows the CSS3 recommendation, but <i>blur</i> is not supported.</p>
<p class="code">
&lt;span style="text-shadow: 0.03em 0.03em #666, -0.015em -0.015em white;"&gt;<br />
&lt;div style="box-shadow: 0.3em 0.3em #888888;"&gt;
</p>
</div>
```

```
<h3>Other Enhancements<bookmark content="Other Enhancements" level="0" /></h3>
```

```
<h4>Column Totals (Tables)<bookmark content="Column totals" level="1" /></h4>
<p>{colsum} placed in the footer of a table will automatically display the sum of that column. If the table breaks across more than one page, the sum of the values on that page will be displayed. A number following the colsum e.g. {colsum2} will force that number of decimal places to be displayed.</p>
```

```
<table class="zebra" align="center">
<caption class="tablecaption" align="bottom">Table caption goes here</caption>
<thead>
<tr>
<th>Header Row</th>
<td>Header Row</td>
<td>Header Row</td>
</tr>
</thead>
<tfoot>
<tr>
<th></th>
<td>Column total: (using colsum2 in {})</td>
<td class="cost"><b>{colsum2}</b></td>
</tr>
</tfoot>
<tbody>
<tr>
<th>Row 1</th>
<td>This is data</td>
<td class="cost">167.00</td>
</tr>
<tr>
<th>Row 2</th>
<td>
<p>This is data p</p>
</td>
<td class="cost">
<p>444.53</p>
</td>
</tr>
<tr>
<th>
<p>Row 3</p>
</th>
<td>
<p>This is long data</p>
</td>
<td class="cost">14.00</td>
</tr>
<tr>
<th>
<p>Row 4</p>
</th>
<td>
<p>This is data</p>
</td>
<td class="cost">
<p>0.88</p>
</td>
</tr>
</tbody>
```

```

<td>Row 5</td>
<td>Also data</td>
<td class="cost">144.00</td>
</tr>
<tr>
<td>Row 6</td>
<td>Also data</td>
<td class="cost">8089.00</td>
</tr>
<tr>
<td>Row 7</td>
<td>Also data</td>
<td class="cost">3.00</td>
</tr>
<tr>
<td>Row 8</td>
<td>Also data</td>
<td class="cost">23.00</td>
</tr>
</tbody></table>
<br />

```

```

<h4>Table <span style="font-variant: small-caps">caption</span><bookmark content="Table caption"
level="1" /></h4>

```

The caption element for tables is partially supported (see example above).</p>

```

<p class="code">

```

```

&lt;caption align="top|bottom" style="caption-side: top|bottom"&gt;

```

```

</p>

```

```

<ul>

```

```

<li>The caption must come immediately after &lt;table&gt;.</li>

```

```

<li>The CSS <span class="css">caption-side</span> or HTML <span class="css">align</span> attribute of
top|bottom supported</li>

```

```

<li>Left or right placement are not supported.</li>

```

```

<li>The caption is handled as a separate block element brought outside the table, so:

```

```

<ul>

```

```

    <li>CSS will not cascade correctly from the table</li>

```

```

    <li>the width of the caption block is that of page or of the block element containing the
table</li>

```

```

    <li>text alignment will be to the page-width not the table width</li>

```

```

    <li>if table page-break-after: always, the caption will follow the pagebreak</li>

```

```

</ul></li>

```

```

</ul>

```

```

<h4>Core fonts in non-core font document<bookmark content="Core fonts" level="1" /></h4>

```

Core fonts, which do not need to be embedded in a PDF, can now be included in a document which uses non-core fonts. The pseudo font-family names: chelvetica, ctimes and ccourier should be used.</p>

```

<p class="code">

```

```

&lt;div style="font-family: chelvetica"&gt;

```

```

</p>

```

NB You could force mPDF to always use core fonts when Arial/Helvetica/Courier are specified, by editing \$this->fonttrans in config_fonts.php:</p>

```

<p class="code">

```

```

$this->fonttrans = array(<br />

```

```

    \'arial\' => \'chelvetica\',<br />

```

```

    \'helvetica\' => \'chelvetica\',<br />

```

```

    \'timesnewroman\' => \'ctimes\',<br />

```

```

    \'times\' => \'ctimes\',<br />

```

```

    \'couriernew\' => \'ccourier\',<br />

```

```

    \'courier\' => \'ccourier\',<br />

```

```

    ...

```

```

</p>

```

```

<br />

```

```

<h4>Javascript in Forms<bookmark content="Javascript in Forms" level="1" /></h4>

```

Javascript used in (active) forms has been altered to reflect the Adobe Acrobat specification for

Javascript in PDF documents.</p>

<p>textarea and input (text-types) now accept javascript as: onKeystroke, onValidate, onCalculate and onFormat. onChange is deprecated but is not ignored; it works as though for onCalculate. (PS Select still accepts onChange)</p>

<h4>Overlapping Rows in Tables<bookmark content="Overlapping Table Rows" level="1" /></h4>

<p>Support for overlapping rowspans in tables has been improved (although probably not foolproof!)</p>

<table style="border-collapse: separate; border-spacing: 3.5mm;">

<tr>

<td style="width: 30mm; height: 30mm; background-color: rgb(213,226,253)"> </td>

<td style="width: 30mm; height: 30mm; background-color: rgb(75,155,215)"> </td>

<td rowspan="2" style="width: 30mm; height: 63.5mm; background-color: rgb(183,225,253)"> </td>

</tr>

<tr>

<td colspan="2" rowspan="2" style="width: 63.5mm; height: 63.5mm; background-color: rgb(183,225,253)"> </td>

</tr>

<tr>

<td style="width: 30mm; height: 30mm; background-color: rgb(75,155,215)"> </td>

</tr>

</table>

<h3>Circular Text<bookmark content="Circular Text" level="0" /></h3>

<p>Circular Text can be included in a PDF document as a custom HTML tag (or a function)</p>

top-text and/or bottom-text can be specified

Radius (r) and font-size (using CSS) are user-defined

Width and height are calculated from radius and font-size

Other CSS styles supported on Circular Text: border, margin, padding, color, background-color, font-family, font-size, font-weight, font-style, display, visibility, and opacity

space-width should be specified as an integer defining the letter-spacing as a percentage of normal (default 120)

char-width should be specified as an integer defining the width of each character as a percentage of normal (default 100)

Circular Text is displayed as though an in-line element

<p>NB If \$mpdf->useKerning is true then automatic kerning will be used on Circular Text.</p>

<p class="code">

<textcircle r="30mm" top-text="Circular Text Circular Text" style="color: blue; font-size: 34pt; font-style: italic" />

<textcircle r="30mm" space-width="120" char-width="150" top-text="• Circular Text •" bottom-text="Circular Text" style="background-color: #FFAAAA; border:1px solid red; padding: 0.3em; margin: 0.3em; color: #000000; font-size: 21pt; font-weight:bold; font-family: Arial" />

</p>

<textcircle r="30mm" top-text="Circular Text Circular Text" style="color: blue; font-size: 34pt; font-style: italic" />

<textcircle r="30mm" space-width="120" char-width="150" top-text="• Circular Text •" bottom-text="Circular Text" style="background-color: #FFAAAA; border:1px solid red; padding: 0.3em; margin: 0.3em; color: #000000; font-size: 21pt; font-weight:bold; font-family: Arial" />

<h3 style="page-break-before: left;">Spread tables<bookmark content="Spread Tables" level="0" /></h3>

<div class="gradient text">


```
;
//=====
if ($_REQUEST['html']) { echo $html; exit; }
if ($_REQUEST['source']) {
    $file = __FILE__;
    header("Content-Type: text/plain");
    header("Content-Length: ". filesize($file));
    header("Content-Disposition: attachment; filename='".$file."'");
    readfile($file);
    exit;
}

//=====
$pdf->useActiveForms = true;

$pdf->bookmarkStyles = array(
    0 => array('color'=> array(0,64,128), 'style'=>'B'),
    1 => array('color'=> array(128,0,0), 'style'=>'),
    2 => array('color'=> array(0,128,0), 'style'=>'I'),
);

$pdf->useKerning=true;    // set this to improve appearance of Circular text
                        // must be set before the font is first loaded

$pdf->WriteHTML($html);

// JAVASCRIPT FOR WHOLE DOCUMENT
$pdf->SetJS('
function TwoPages() {
    this.layout="TwoColumnRight";
    this.zoomType = zoomtype.fitW;
}
function OnePage() {
    this.layout="SinglePage";
    this.zoom = 100;
}
');

// OUTPUT
$pdf->Output(); exit;

//=====
//=====
//=====
//=====

?>
```