

# Page breaks

You can force a page break anywhere in the document either by using HTML code or PHP:

```
$mpdf->AddPage();  
<pagebreak />
```

You can define or change all page characteristics when you add the new page:

- orientation
- margins
- numbering (on/off, style or suppress)
- header/footer

Note the special TOCpagebreak() or <tocpagebreak> which are like AddPage() and <pagebreak> whilst at the same time marking the position for a Table of Contents to be later inserted. CSS @page can also be used to define page breaks.

## Forcing Page Breaks

You can set the CSS value for *page-break-before* to "always | left | right" for any block element (p, div etc). This will force a page-break, but take care that any enclosing (outer) block elements will be 'closed down' and their characteristics lost.

## Avoiding Page Breaks

mPDF has limited scope to control when automatic page-breaks occur, and does not have 'widows' or 'orphans' protection.

### page-break-inside

You can set the CSS value for *page-break-inside* to "avoid" for any block element. mPDF will try to avoid a page-break within the block, but this only works across a maximum of 2 pages, and is not compatible with table autosize or table rotate

### "Keep-with-table"

```
$mpdf->use_kwt = true; // DEFAULT=false
```

If set to true, mPDF will automatically set *page-break-inside*=avoid for any H1-H6 header that *immediately* precedes a table, thus keeping the heading together with the table.

- automatically sets the table to fit on one page (i.e. table:*page-break-inside*=avoid) if it is a rotated table
- ignored when: columns on, *page-break-inside*=avoid for surrounding element, active Forms

### Tables

If a table has the property *page-break-inside*:avoid and will not fit on the page, mPDF tries to shrink it to fit - up to a maximum "shrink-factor" set by the variable \$mpdf->shrink\_tables\_to\_fit - default is 1.4 (i.e. about 70% original size). If this still won't fit, it moves it to the next page.

A shrunk table may not be what you want. You can prevent this resizing either by setting the maximum shrink-factor for a particular table e.g. <table autosize="1"> or by setting the variable for the whole document i.e.

```
$mpdf->shrink_tables_to_fit=1;
```

(Note that mPDF will always resize tables if it is the only way to fit a row or whole table onto a full page.)

## Headers and Footers

See the documentation for Headers and Footers

## Types of page break

The handling of borders and padding at page breaks was updated in mPDF 6.0. mPDF has three types of page

breaks:

- 1) "slice" - no border and no padding are inserted at a break. The effect is as though the element were rendered with no breaks present, and then sliced by the breaks afterward
- 2) "cloneall" - each page fragment is independently wrapped with the borders and padding of all open elements.
- 3) "clonebycss" - open elements which have the (custom) CSS property "box-decoration-break" set to "clone" are independently wrapped with their border and padding.

The difference between 2) and 3) is illustrated by this example:

```
<style>
div { border: 1px solid black; padding: 1em; }
.level1 { box-decoration-break: slice; }
.level2 { box-decoration-break: clone; }
.level3 { box-decoration-break: clone; }
</style>

<div class="level1">
<div class="level2">
<div class="level3">
<p style="page-break-after:always">...</p>
<p>...</p>
</div>
</div>
</div>
```

At the forced pagebreak which occurs after the P element:

If the page break type is "cloneall" - the three DIV elements will all be closed, by drawing the border and padding for each at the end of the page; the three DIV elements will be re-opened, drawing the borders and padding, at the top of the next page.

If the page break type is "clonebycss" - starting from the innermost element (div.level3) the DIV elements will have a border and padding at the end of the page if "box-decoration-break" is clone. In this case level2 and level 3 will be closed/cloned and level 1 will be sliced; the opposite will occur at the top of the next page.

### Control of page breaks

Automatic page breaks (in flow of text)	Always "slice"
<toctoppagebreak>	Always "cloneall"
<formfeed>	Always "slice"
If using columns	Always "cloneall"
Page break forced by change of @page selector	Always "cloneall"
<pagebreak>	Always "cloneall" if a change in page size or margins is specified. Otherwise page break type is determined by value of configurable variable: \$this->defaultPagebreakType. Default is "cloneall". Default can be overridden by attribute "page-break-type" e.g. <pagebreak page-break-type="clonebycss" />
Page breaks forced by: page-break-before or page-break-after	Page break type determined by value of configurable variable: \$this->defaultPagebreakType. Default is "cloneall".

### Notes on page breaking

"box-decoration-break: slice | clone" was proposed for CSS3 in

<http://www.w3.org/TR/2012/CR-css3-background-20120417/#the-box-decoration-break> but it appears that it may be withdrawn. Default is "slice"; it is not inherited.

"page-break-before" is not supported on <table>.

"page-break-before|after" is ignored if set on block elements inside a table.

Background images and gradients are not sliced (always cloned).

\$this->restoreBlockPagebreaks in config.php is now redundant.

Printed on Wed 05 Aug 2015 12:06:39 GMT +0100 (DST)