

# CSS PAGED MEDIA - @PAGE RULE

[http://www.tutorialspoint.com/css/css\\_paged\\_media.htm](http://www.tutorialspoint.com/css/css_paged_media.htm)

Copyright © tutorialspoint.com

Paged media differ from continuous media in that the content of the document is split into one or more discrete pages. Paged media includes paper, transparencies, pages that are displayed on computer screens, etc.

The CSS2 standard introduces some basic pagination control features that let authors help the browser figure out how to best print their documents.

The CSS2 page model specifies how a document is formatted within a rectangular area -- the page box -- that has a finite width and height. These features fall into two groups –

- CSS2 features that define a particular page layout.
- CSS2 features that control the pagination of a document.

## Defining Pages : the @page rule

The CSS2 defines a "page box", a box of finite dimensions in which content is rendered. The page box is a rectangular region that contains two areas –

- **The page area** – The page area includes the boxes laid out on that page. The edges of the page area act as the initial containing block for layout that occurs between page breaks.
- **The margin area** – It surrounds the page area.

You can specify the dimensions, orientation, margins, etc., of a page box within an @page rule. The dimensions of the page box are set with the 'size' property. The dimensions of the page area are the dimensions of the page box minus the margin area.

For example, the following @page rule sets the page box size to 8.5 × 11 inches and creates '2cm' margin on all sides between the page box edge and the page area –

```
<style type="text/css">
  <!--
    @page { size:8.5in 11in; margin: 2cm }
  -->
</style>
```

You can use the *margin*, *margin-top*, *margin-bottom*, *margin-left*, and *margin-right* properties within the @page rule to set margins for your page.

Finally, the *marks* property is used within the @page rule to create crop and registration marks outside the page box on the target sheet. By default, no marks are printed. You may use one or both of the *crop* and *cross* keywords to create crop marks and registration marks, respectively, on the target print page.

## Setting Page Size

The *size* property specifies the size and orientation of a page box. There are four values which can be used for page size –

- **auto** – The page box will be set to the size and orientation of the target sheet.
- **landscape** – Overrides the target's orientation. The page box is the same size as the target, and the longer sides are horizontal.
- **portrait** – Overrides the target's orientation. The page box is the same size as the target, and the shorter sides are horizontal.
- **length** – Length values for the 'size' property create an absolute page box. If only one length value is specified, it sets both the width and height of the page box. Percentage values are not allowed for the 'size' property.

In the following example, the outer edges of the page box will align with the target. The percentage value on the 'margin' property is relative to the target size so if the target sheet dimensions are 21.0cm x 29.7cm *i. e.*, A4, the margins are 2.10cm and 2.97cm.

```
<style type="text/css">
  <!--
  @page {
    size: auto; /* auto is the initial value */
    margin: 10%;
  }
  -->
</style>
```

The following example sets the width of the page box to be 8.5 inches and the height to be 11 inches. The page box in this example requires a target sheet size of 8.5" x 11" or larger.

```
<style type="text/css">
  <!--
  @page {
    size: 8.5in 11in; /* width height */
  }
  -->
</style>
```

Once you create a named page layout, you can use it in your document by adding the page property to a style that is later applied to an element in your document. For example, this style renders all the tables in your document on landscape pages –

```
<style type="text/css">
  <!--
  @page { size : portrait }
  @page rotated { size : landscape }
  table { page : rotated }
  -->
</style>
```

Due to the above rule, while printing, if the browser encounters a <table> element in your document and the current page layout is the default portrait layout, it starts a new page and prints the table on a landscape page.

## Left, Right, and First pages

When printing double-sided documents, the page boxes on left and right pages should be different. It can be expressed through two CSS pseudo-classes as follows –

```
<style type="text/css">
  <!--
  @page :left {
    margin-left: 4cm;
    margin-right: 3cm;
  }

  @page :right {
    margin-left: 3cm;
    margin-right: 4cm;
  }
  -->
</style>
```

You can specify the style for the first page of a document with the :first pseudo-class –

```
<style type="text/css">
  <!--
  @page { margin: 2cm } /* All margins set to 2cm */
```

```
@page :first {
  margin-top: 10cm    /* Top margin on first page 10cm */
}
-->
</style>
```

## Controlling Pagination

Unless you specify otherwise, page breaks occur only when the page format changes or when the content overflows the current page box. To otherwise force or suppress page breaks, use the *page-break-before*, *page-break-after*, and *page-break-inside* properties.

Both the *page-break-before* and *page-break-after* accept the *auto*, *always*, *avoid*, *left*, and *right* keywords.

The keyword *auto* is the default, it lets the browser generate page breaks as needed. The keyword *always* forces a page break before or after the element, while *avoid* suppresses a page break immediately before or after the element. The *left* and *right* keywords force one or two page breaks, so that the element is rendered on a left-hand or right-hand page.

Using pagination properties is quite straightforward. Suppose your document has level-1 headers start new chapters with level-2 headers to denote sections. You'd like each chapter to start on a new, right-hand page, but you don't want section headers to be split across a page break from the subsequent content. You can achieve this using following rule –

```
<style type="text/css">
  <!--
  h1 { page-break-before : right }
  h2 { page-break-after : avoid }
  -->
</style>
```

Use only the *auto* and *avoid* values with the *page-break-inside* property. If you prefer that your tables not be broken across pages if possible, you would write the rule –

```
<style type="text/css">
  <!--
  table { page-break-inside : avoid }
  -->
</style>
```

## Controlling Widows and Orphans

In typographic lingo, orphans are those lines of a paragraph stranded at the bottom of a page due to a page break, while widows are those lines remaining at the top of a page following a page break. Generally, printed pages do not look attractive with single lines of text stranded at the top or bottom. Most printers try to leave at least two or more lines of text at the top or bottom of each page.

- The **orphans** property specifies the minimum number of lines of a paragraph that must be left at the bottom of a page.
- The **widows** property specifies the minimum number of lines of a paragraph that must be left at the top of a page.

Here is the example to create 4 lines at the bottom and 3 lines at the top of each page –

```
<style type="text/css">
  <!--
  @page{orphans:4; widows:2;}
  -->
</style>
```