用prince生成PDF时的样式设置

1. **prince官方网址**

http://www.princexml.com/doc/7.0

1. **如果页面在PDF中显示不全，可设置以下几个属性**

@page **{** size**:** A4 landscape **}**—横屏

@page **{** size**:** A4 *portrait* **}**—竖屏

@page **{** size**:** 35cm 40cm **}**

@page **{** margin**:** 40pt 6pt 40pt 6pt **}**

可以合写：

@page **{** size**:** 35cm 40cm; margin**:** 40pt 6pt 40pt 6pt **}**

1. **table-layout与width=’100%’**

如果div内有table，并且div设置了浮动，当为table设置了table-layout:fixed和width=’100%’时，table就会占据整个页面宽度，而不是其父元素div的宽度.

原因不明

1. **网页在PDF中的显示**

只要指定了打印纸张的尺寸和边距：@page { size: A4;margin: 40pt 5pt; }

* 无论网页设置的宽度是多少，PDF中显示的样式是一成不变的，无论是宽度还是其它样式，即使是文字的换行，设置前从哪里换设置后还是该处，没有任何变化；
* 无论PDF中设置的分辨率是多少，PDF中显示的样式也是一成不变的，只不过是宽度加大了而已，文字的换行跟上一点一样，也不会改变；

1. **PDF对样式的支持**

* **样式引用**
* 目前在/lib/pdf\_helper.rb中定义pdf引用的样式文件,在pdf\_layout中定义的不起作用.

|  |
| --- |
| prince.add\_style\_sheets(  "#{RAILS\_ROOT}/public/stylesheets/user/common.css",  "#{RAILS\_ROOT}/public/stylesheets/user/pdf.css"  ) |

* PDF不支持写在HTML页面内的样式，例如：<p style=””>XXX</p>;
* PDF不支持<head><style=”text/css”></style></head>中的样式；
* PDF支持写在CSS文件中的样式；
* @import 在PDF样式文件中必须将其放在最开始,否则将不起作用,即使只放在[@charset "utf-8"; /\* CSS Document \*/]它们后面也不起作用.

@import的两种写法:

写法一:@import url("common.css");

写法二:@import "common.css";

以上原因不明

* **PDF只支持写在PDF内的样式，如：**

<div id=”wrap”><%= yield %></div>

PDF读取的是<%= yield %>中的内容，<div id=”wrap”>并没有被读取，因此为#wrap加的样式在PDF中是不会显示的，同理，如果对PDF某段内容的加样式#wrap .title{},那么该显示也不会显示，因为PDF并没有读取<div id=”wrap”>。

* **图片**

rails框架中,在样式文件中定义的背景图片路径一般写成”/images/--”这种形式,这是因为rails默认路径从public开始,但是在pdf的样式文件中这样写,图片读取不出来,必须使用相对路径,即”../images/--”

**■ Paged Media**

1. **页码的显示Page Numbers**

* **显示当前页**

|  |
| --- |
| @page {  @bottom-right { content: counter(page) }；  font-style: italic  } |

* **显示总页数**

|  |
| --- |
| @page {  @bottom-right { content: counter(pages) }  } |

* **显示当前页和总页数**

|  |
| --- |
| @page {  content: "Page " counter(page) " of " counter(pages)  } |

* **注意点**

@page中只有一个属性时，可以不加分号，如果有一个以上，最后一个属性可以不加分号，但是前面的必须加分号，否则式样不起作用；

1. **页眉和页脚Page Headers and Footers**

页眉和页脚即是放置在打印时头部和尾部的空白区域。

|  |
| --- |
| 【Page headers】   * @top-left-corner * @top-left * @top or @top-center * @top-right * @top-right-corner   【Page footers】   * @bottom-left-corner * @bottom-left * @bottom or @bottom-center * @bottom-right * @bottom-right-corner   【解释】   * @top-left-corner   //左空白区域（即内容左边与纸张左边之间的距离），如果内容超出了这个宽度，则会自动换行；   * @top-left   //与内容区左对齐 |

* **在页面上创建内容【Using generated content】**

在页眉的左侧会显示文字”TOP SECRET”,而且是红色的.

**＜CSS＞**

|  |
| --- |
| @page {  @top-left {  content: "TOP SECRET";  color: red  }  } |

* **COPY文档中的内容【Copying text content from the document】**

可以将文档中的某部分内容COPY到指定位置

**＜CSS＞**

|  |
| --- |
| @page {  @top {  content: string(doctitle)  }  }  h1 { string-set: doctitle content() } |

**■ 表格**

1. **thead tfoot与分页【Running table headers and footers】**

表格跨页显示时，如果表格中包含thead,tfoot,在每个分页上都会自动给该表格的数据添加thead和tfoot.

|  |
| --- |
|  |

如果给表格加了tbody,会以tbody进行分页，如果不加，则会出现上一页只显示表头，页面还很空，数据全跑到下一页显示的效果了。

1. **显示行数【Numbering table rows】**

为表格行加行号，单纯用CSS实现，而不需要用程序实现，因为不是用程序实现的，所以行号列的样式要单独设置。

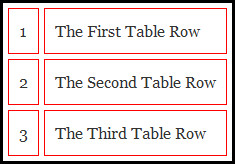
**＜CSS＞**

|  |
| --- |
| table { counter-reset: row }  tr { counter-increment: row }  tr::before {  content: counter(row);  display: table-cell  } |

**＜HTML＞**

|  |
| --- |
| <table>  <tr><td>The First Table Row</td></tr>  <tr><td>The Second Table Row</td></tr>  <tr><td>The Third Table Row</td></tr>  </table> |

**＜效果＞**



1. **table-layout:auto/fixed**

如果给表格设置了table-layout:fixed;width:auto;这两个属性，则等同于table-layout:auto,即如果给表格设置了width:auto属性，无论table-layout的值是什么，都会解析成table-layout:auto.

1. **合并列【Cells that span columns】**

可以用CSS实现

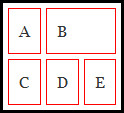
**＜CSS＞**

|  |
| --- |
| td.colspan2 { table-column-span: 2 } |

**＜HTML＞**

|  |
| --- |
| <td class="colspan2"> B </td> |

**＜效果＞**



1. **合并行【Cells that span rows】**

可以用CSS实现

**＜CSS＞**

|  |
| --- |
| td.rowspan2 { table-row-span: 2 } |

**＜HTML＞**

|  |
| --- |
| <td class="rowspan2"> A </td> |

**＜效果＞**

