

Controlling text alignment

Print fiction has historically been displayed with justified text, that is, all the lines of text are adjusted with extra spaces so that they end evenly on both sides. Unfortunately, to do this in a way that looks half decent, you need a robust hyphenation dictionary. Otherwise, particularly long words can cause unsightly rivers of white space that flow through your pages.

Some ereaders let the user control the alignment of an ebook's text. The iBooks app displays justified text by default. Users can go to Settings, click iBooks and choose OFF next to Full Justification to use the settings specified by the book designer. Unfortunately, that setting is a bit hidden. Even worse, iBooks has a bug that keeps it from following the `text-align` property in the CSS.

Luckily, there is a workaround. It turns out that iBooks 1.1 will pay attention to the CSS `text-align` property if and only if there is a second element, within the main `p` element, that contains the text.

1 Contain the body text within every single `p` element in another element. If you're not specifying the font, use `span`. If you do want to specify the font, use `samp`, as described in "iBooks and fonts". You can use the GREGP expression, `<p>(.*</p>` to find all of your `p` elements (add the `class` to that expression if necessary), and then replace it with `<p><samp>\1</samp></p>` or `<p>\1</p>`, depending on whether you're applying font family info or not.

```
<p><span>Here is the text.</span></p>
```

2 Specify the alignment for the `p` elements in the CSS.

```
p {text-align: left}
```

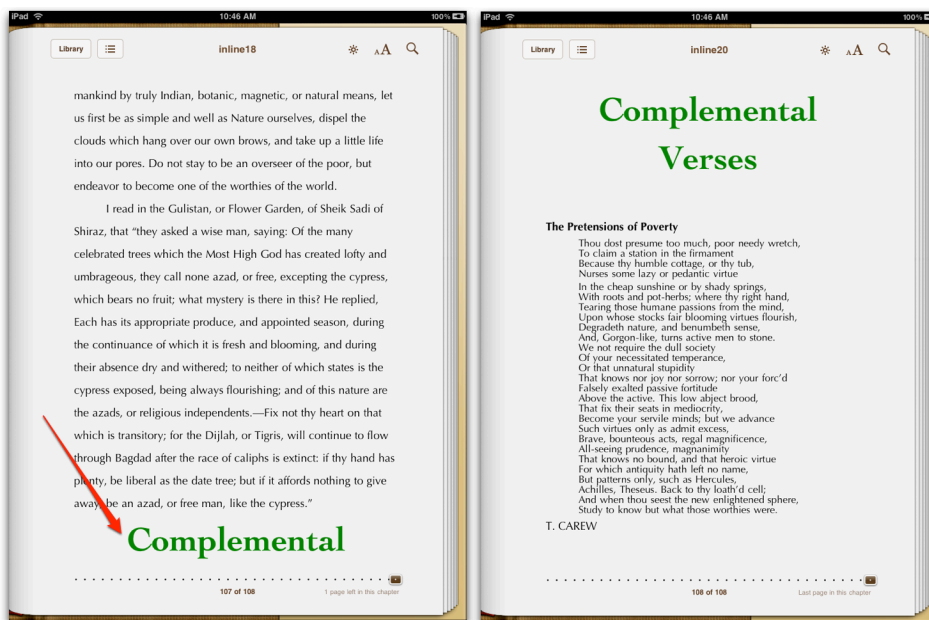
If you do use `samp`, be sure to set the font in the CSS as well. If you use `span`, no extra CSS is required for the `span` selector.

When Apple fixes this bug in iBooks, the workaround will no longer be necessary, but leaving it in shouldn't create problems in future, less-buggy ereaders.

Keeping elements together

The iBooks app doesn't currently support the standard CSS page element controls, like `page-break-before`, `page-break-after`, and `page-break-inside`. Neither does it support the standard CSS `orphans` and `widows` properties that can keep individual lines from appearing at the top and bottom of pages. Some other readers, including Barnes & Noble's Nook, do support the `orphans` and `widows` properties and even have default values set. How then can you best control the flow of text to keep headers from showing up alone at the bottom of pages, captions from being separated from their illustrations, and the last line of a paragraph from starting a new page?

Until iBooks supports the real page-break controls, you can exert a fair amount of force with one special value of the `display` property: `inline-block`. This value forces iBooks to view the affected content much as if it were one discrete item, like an image. So, if you apply `display: inline-block` to a `div`, all of the contents of that `div` are treated as a unit, and are kept together on a page—as much as possible—often starting a new page in the process. It is a roundabout way of creating a page break, but it's definitely better than nothing.



On the left, without `inline-block`, the header starts at the bottom of the page. Ew! On the right, thanks to `inline-block`, the header begins a new page.