

Extracted from:

HTML5 and CSS3, Second Edition

Level Up with Today's Web Technologies

This PDF file contains pages extracted from *HTML5 and CSS3, Second Edition*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2013 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

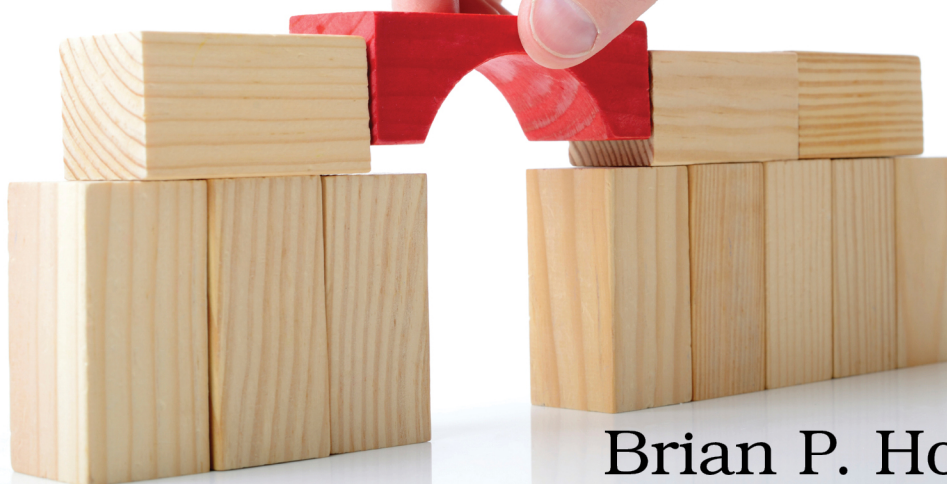
Dallas, Texas • Raleigh, North Carolina

The
Pragmatic
Programmers

HTML5 and CSS3

Second Edition

Level Up with Today's
Web Technologies



Brian P. Hogan

Edited by Susannah Davidson Pfalzer



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at <http://pragprog.com>.

The team that produced this book includes:

Susannah Davidson Pfalzer (editor)
Potomac Indexing, LLC (indexer)
Candace Cunningham (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2013 The Pragmatic Programmers, LLC.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-937785-59-8
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—October 2013

Creating Multicolumn Layouts

The print industry has had columns for years, and web designers have looked at those publications with envy. Narrow columns make it easier for readers to read your content, and with displays getting wider, developers are looking for ways to preserve comfortable column widths. After all, nobody wants to follow multiple lines of text across the monitor any more than they want a line of text to flow across the whole page of a newspaper. There have been some pretty clever solutions in the past ten years, but none of those solutions are as simple and easy as the method the CSS3 specification provides.

Splitting Columns

Each month, AwesomeCo publishes a newsletter for its employees. The company uses a popular web-based email system. Email-based newsletters don't quite look good and are hard to maintain. They've decided to put the newsletter on the intranet site and are planning to send emails to employees, with a link to pull up the newsletter in their browsers. For a mocked-up version of this newsletter, see the following figure.



Figure 12—Our single-column newsletter is hard to read because it's very wide.

The new director of communications, who has a background in print publications, has decided that she would like the newsletter to look more like an actual newsletter, with two columns instead of one.

If you've ever tried to split some text into multiple columns using `div`s and `float`s, you know how hard that can be. The first big hurdle is deciding where to split the text. In publishing software such as InDesign, you can link text boxes together so that when one fills up with text, the text flows smoothly into the linked text area. We don't have anything quite like that on the Web yet, but we have something that works really well and is quite easy to use. We can split an element's contents into multiple columns, each with the same width.

We'll start with the markup for the newsletter. It's fairly basic HTML. Since its content will change once it's written, we'll use placeholder text for the content:

```
css3_columns/condensed_newsletter.html
```

```
<body>
  <div id="container">
    <header id="header">
      <h1>AwesomeCo Newsletter</h1>
      <p>Volume 3, Issue 12</p>
    </header>
    <section id="newsletter">
      <article id="director_news">
        <header>
          <h2>News From The Director</h2>
        </header>
        <div>
          <p>
            Lorem ipsum dolor sit amet...
          </p>
          <aside class="callout">
            <h4>Being Awesome</h4>
            <p>
              &quot;Lorem ipsum dolor sit amet, ...&quot;
            </p>
          </aside>
          <p>
            Duis aute irure dolor in ...
          </p>
        </div>
      </article>
      <article id="awesome_bits">
        <header>
          <h2>Quick Bits of Awesome</h2>
        </header>
```

```

    <div>
      <p>
        Lorem ipsum dolor sit amet...
      </p>
    </div>
  </article>
  <article id="birthdays">
    <header>
      <h2>Birthdays</h2>
    </header>
    <div>
      <p>
        Lorem ipsum dolor sit amet...
      </p>
    </div>
  </article>
</section>
<footer id="footer">
  <h6>
    Send newsworthy things to
    <a href="mailto:news@aweseomco.com">news@awesomeco.com</a>.
  </h6>
</footer>
</div>
</body>

```

To split this into a two-column layout, we need to add a few new properties to our style sheet:

- `column-count` lets us specify how many columns we want to divide our content into.
- `column-gap` defines how much space should be placed in between the columns.
- `column-rule` gives us a border between the columns.

Let's add this to our style sheet to split the content into two columns with a little gutter between them:

```

css3_columns/stylesheets/style.css
#newsletter{
  -webkit-column-count: 2;
  -webkit-column-gap: 20px;
  -webkit-column-rule: 1px solid #ddccb5;
  -moz-column-count: 2;
  -moz-column-gap: 20px;
  -moz-column-rule: 1px solid #ddccb5;
  column-count: 2;
  column-gap: 20px;
  column-rule: 1px solid #ddccb5;
}

```

The next figure shows that now we have something much more readable.



Figure 13—Our new two-column newsletter

We can add more content, and the browser will automatically determine how to split the content evenly. Also, notice that the floated elements float to the columns that contain them.

To make these columns work across multiple browsers, we've had to define the properties multiple times, prefixing each rule for a specific type of browser.

Vendor-Specific Prefixes

While the World Wide Web Consortium was busy figuring out what features needed to go into the CSS specification, browser-makers added new features themselves and decided to prefix their own implementations. Some of those implementations ended up becoming the standards, solidifying prefixing as a viable practice that continues today. These prefixes let browser-makers introduce features early before they become part of a final specification, and since these features may not follow the specification, the browser-makers can implement the actual specification while keeping their own implementation as well. Most of the time, the vendor-prefixed version matches the CSS specification, but occasionally you'll encounter differences. Unfortunately for you, that means you'll need to declare some properties more than once for each type of browser. Here are the most common vendor prefixes:

- Firefox uses the `-moz-` prefix.
- Chrome and Safari, as well as many mobile browsers and recent versions of Opera, use the `-webkit-` prefix.
- Older versions of Opera use the `-o-` prefix.

Don't blindly use these prefixes, though. As browsers implement more of the standards, these prefixes are less necessary and create dead weight in your CSS. Keep an eye on the browsers your visitors use, and prune these selectors out of your style sheets if you don't need them anymore. You can employ Can I Use... to determine if you need prefixes.⁶



Joe asks:

Can I Specify Different Widths for Each Column?

Nope. Your columns must each be the same size. I was a little surprised, too, at first, so I double-checked the specification, and at the time of writing there is no provision for specifying multiple column widths.

However, when you think about how columns are traditionally used, it makes sense. Columns are not intended to be a hack to easily make a sidebar for your website any more than tables are. Columns are meant to make reading long sections of text easier, and equal-width columns are perfect for that.

Falling Back

CSS3 columns don't work in Internet Explorer 9 and older, and it's probably fine to not have a fallback solution since the content is still readable. But if you've got your heart set on consistency across browsers, you can use `CSS3MultiColumn`, which adds support for basic multicolumn features.⁷

Simply load it *after* your style sheets, and it does the rest of the work for you. Since it only has to target Internet Explorer 9 and below, we can get away with wrapping it in a conditional comment, along with the JavaScript to make Internet Explorer 8 recognize our HTML5 elements:

```
css3_columns/newsletter.html
<!--[if lte IE 9]>
<script>
  // support for styling HTML5 elements
  document.createElement("section");
```

6. <http://caniuse.com/>

7. <https://github.com/BetleyWhitehorne/CSS3MultiColumn>


```

document.createElement("header");
document.createElement("footer");
document.createElement("article");
document.createElement("aside");
</script>

<script src="javascrip/css3-multi-column.min.js"></script>
<![endif]->

```

Refresh the page in Internet Explorer, and you'll see your two-column newsletter, as in the following figure.

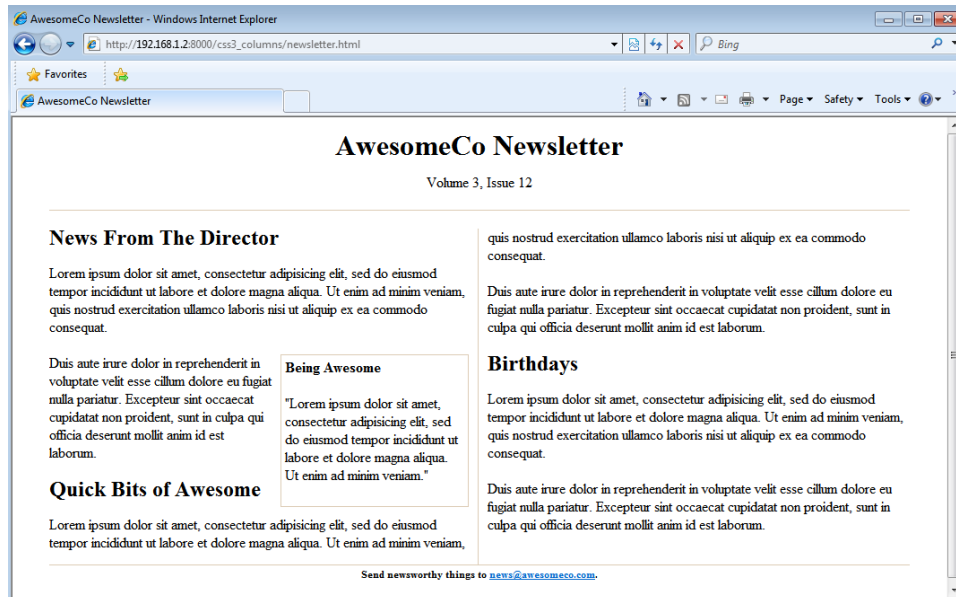


Figure 14—Our Internet Explorer version works, but needs some minor adjustments.

Visitors without JavaScript will still be able to read the content as before, so everybody wins.

Separating your content into multiple columns can make your content easier to read. However, if your page is long, your users might find it annoying to have to scroll back to the top to read the next column. Use this with care.