

Extracted from:

Pragmatic Guide to Sass

This PDF file contains pages extracted from *Pragmatic Guide to Sass*, 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 printer versions; the content is otherwise identical.

Copyright © 2010 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

Pragmatic Guide to Sass

Hampton Catlin and
Michael Lintorn Catlin

Edited by Kay Keppler





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:

Kay Keppler (editor)
Potomac Indexing, LLC (indexer)
Molly McBeath (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2011 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: 978-1-934356-84-5
Printed on acid-free paper.
Book version: P1.0—December 2011

Have you ever been in a situation where you are copying the value of a color over and over again? That very specific blue that your clients want appears in so many places. Then, a couple of weeks later, they want you to change the color. Or—even worse—you have a whole lot of colors to change. Find and replace time! Color handling in CSS is not DRY (there's that *Don't Repeat Yourself* again!) at all.

Sass introduces variables to help us manage problems like this. All variables in Sass are prefixed with a \$ sign. Assigning a variable looks a lot like typing in a CSS property. For instance, we can set the `$primary_color` variable by adding the super-simple line: `$primary_color: #369;`. That's it!

To use the variable, we can just use the variable name where we'd usually use the property value. If we had to change the colors of the whole document, all we'd need to do is change the hex value of the variable and it's sorted for us when the CSS compiles.

We can use variables to represent colors, sizes, percents, and several other things that are less commonly used. Anything that you can put to the right of a CSS property is easily understood by Sass.

Another neat thing about variables is they can be *global* or *scoped*. We've pretty much gone through global variables: They're when a variable is defined on its own line, and they apply to the whole style sheet. Scoped variables, on the other hand, appear within a selector and will only apply to that selector and its children.

We can set default variables with the `!default` tag after assignment. When a variable is used, the default is used if there are no other assignments to that variable.

It's pretty standard in a Sass document to declare the variables at the top of a file and use them throughout. If you're familiar with C, then you'll be familiar with using constants this way. Or if you have a large project, you might want to create a file that defines all of the variables. We'll show you how you can break up your Sass files in [Task 12, *Importing*, on page ?](#).

► Define and use variables.

Download basics/variable_example.scss

```
$primary_color: #369;
$secondary_color: #eee;
$page_width: 300px;

body {
  // Set the background to be #369
  background: $primary_color;
  #wrapper {
    width: $page_width;
    background: white;
    border: $secondary_color;
    h1 {
      color: $primary_color; } } }
```

This compiles to:

```
body {
  background: #336699; }
body #wrapper {
  width: 300px;
  background: white;
  border: #eeeeee; }
body #wrapper h1 {
  color: #336699; }
```