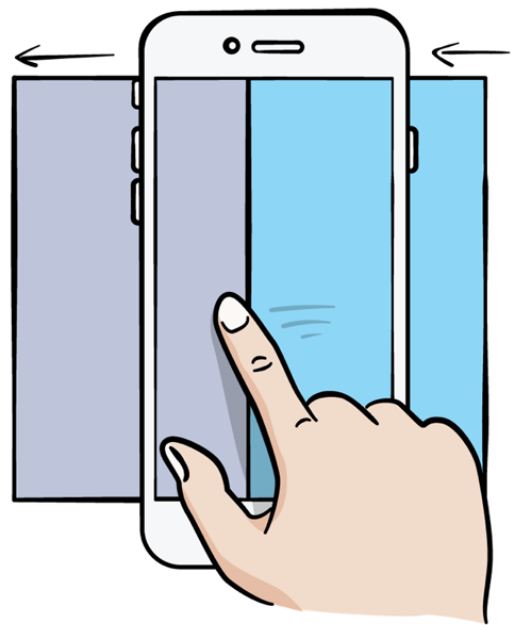


SCROLL VIEW SCHOOL



HANDS-ON CHALLENGES

Scroll View School

Brian Moakley

Copyright ©2017 Razeware LLC.

Notice of Rights

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

Notice of Liability

This challenge and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use of other dealing in the software.

Trademarks

All trademarks and registered trademarks appearing in this book are the property of their own respective owners.

Table of Contents: Overview

Scroll View School for Video 12: Paging Scroll Views 1	5
--	---

Table of Contents: Extended

Scroll View School for Video 12: Paging Scroll Views 1....	5
Challenge.....	5
Need Help.....	6

Scroll View School for Video

12: Paging Scroll Views 1

By Brian Moakley

Creating your own paging scroll views is - conceptually - pretty easy. You lay out your view controllers in a straight line and then allow the user to page through them. In this challenge, you'll do the same, but a lot of the grunt work is handled for you.

Challenge

When you open the starter app, you'll see right away that there's a compile error. Don't worry ... this is intended. When you add the paging functionality, this error will go away.

The sample project contains a bunch of images labeled, 01.jpg, 02.jpg, etc. These represent each month of the year and are meant to be used as paging targets.

The `WallpaperViewController` holds a reference to the images. Each view controller will need to have an image assigned to it.

Here's what you had to do ... in `ViewController.swift`, there's code already in `viewDidLoad()`.

You'll see an empty loop. In this empty loop, create a view controller, assign it a wallpaper image, make sure to add the view controller to the pages array (to keep it in memory), and then add the view to the scroll view.

You'll see the following line:

```
views["page\({i}")] = wallpaperController.view
```

Make sure to uncomment this line. This is associating the wallpaper view with the auto layout metrics. The `wallpaperController` refers to the current `wallpaperViewController` in that loop. If you give it a different name, make sure to update this line of code.

Finally, when all said and done, make sure the scroll view pages through the wallpaper.

Need Help

When in doubt, re-watch the demo in the video. Everything is pretty much the same except don't worry about the auto layout code. That's all built for you.

If you still find yourself stuck, check out the completed project or leave a note on the forums.