Intro to Auto Layout

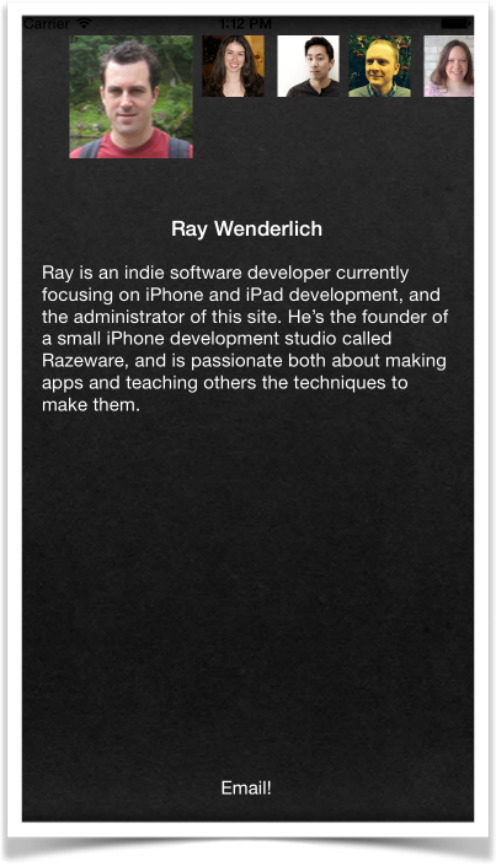
Hands On Challenges

Introduction to Auto Layout Hands-On Challenges

Challenge G: UILayoutGuide and NSLayoutAnchor

Constraints in code using just NSLayoutConstraint can get pretty verbose. Thankfully, Apple added API in iOS 9 to help make constraints in code easier to create and read. They also added UILayoutGuide to help in those cases where you might have created a full-fledged UIView, just for the purpose of making layout easier.

Here's what you will build:



Fortunately, you've built this already, but now you will build it using layout guides and anchors. Good luck!

Challenge Hint

The first time you built this, you used a container view to center the images at the top. This kind of thing is what UILayoutGuide was made for. But rather than containing the images, try using layout guides to evenly space the images across the width of the super view.

Since you already have the view laid out in a storyboard, you can make all the constraints placeholders and they will be removed at build time. Then you don't have to write the code to create the views, just to add the constraints. If you just removed all the constraints, Interface Builder would automatically add some for you and those would conflict with the ones you want to create in code. Creating placeholders lets Interface Builder know that it doesn't have to create automatic constraints.

Challenge Solution

1) Select the view containing all the images and choose **Editor\Unembed** to remove it from the hierarchy.

2) Select the image views and click the **Resolve Auto Layout Issues** button, and choose **Reset to Suggested Constraints**

3) Select all the constraints and in the attributes inspector, check the **Placeholder** check box. Make sure you get all the constraints, some may be added to the image views themselves.

4) Create an outlet collection for the image views and add them all to it from left to right.

5) Compare the code from the solution