Introducing

Stack Views

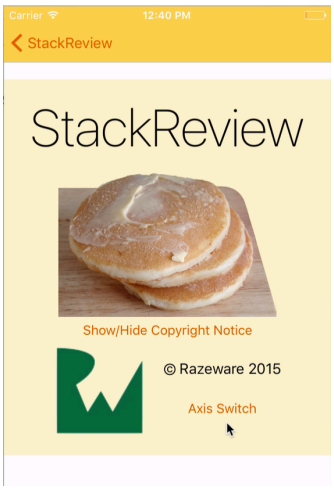
Hands-On Challenges

Introducing Stack Views Hands-On Challenges

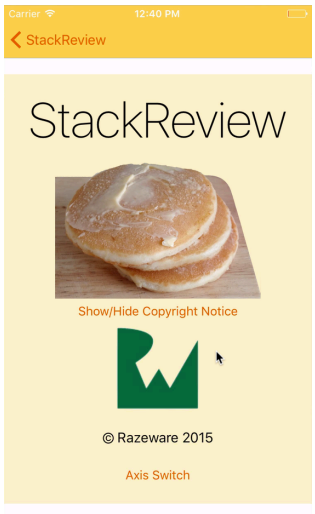
Challenge 5: Stack Views in Code

You saw how to create and configure stack views from code in the tutorial, and now it's your turn.

Your challenge is to update the About view controller to match the following design:



When the user taps the **Axis Switch** button, the layout should animate to the following:



Hints

- In addition to creating another button, you'll need a new stack view.
- You can refer to the previous tutorial to remind yourself how to implement the animation.

Solution

First up, you need to create the required view components – the additional button and the stack view. Notice the way the animation occurs – the copyright label and the button need to stacked in their own stack view, so that flipping the axis of the copyright content stack view will result in the desired effect.

Open AboutViewController.swift and add the following just after you set the text on the copyrightLabel in the `createCopyrightInfo()` method:

```
let axisButton = UIButton(type: .RoundedRect)
axisButton.setTitle("Axis Switch", forState: .Normal)
axisButton.addTarget(self, action: "switchCopyrightAxis",
  forControlEvents: .TouchUpInside)
```

Here you're creating the new button that will switch the axis. You're also adding a target to the button – this is a currently a method without an implementation. You'll fix this later.

You need a new stack view to contain the label and the button. Add the following code just below the code you just added:

```
let textStackView = UIStackView(arrangedSubviews: [copyrightLabel,
  axisButton])
textStackView.axis = .Vertical
textStackView.spacing = 20.0
textStackView.alignment = .Center
```

As you did in the tutorial, here you're creating a new `UIStackView` and passing the views you'd like stacked to the initializer. Then you're setting the axis, spacing and alignment properties to appropriate values.

The last thing you need to do is update the containing stack view to use this new stack view instead of just the label.

Update the line that creates the stack view to match the following:

```
let stackView = UIStackView(arrangedSubviews: [logoImageView,
  textStackView])
```

Notice that you've replaced `copyrightLabel` with the newly created `textStackView` in the `arrangedSubviews` array.

That's all you need to do to update the creation of the copyright content. Now you can turn your attention to providing an implementation for the Axis Switch button.

Create a new method in AboutViewController with the following code:

```
func switchCopyrightAxis() {
  guard let copyrightContentStackView = copyrightContentStackView
    else {
    return
  }

}
```

This method has the same name as the action you associated with the **Axis Switch** button when you created it. The `guard` statement checks that the `copyrightContentStackView` actually exists. If it doesn't you can return early.

Since this method will toggle the axis, you need to work out what to animate it to. Add the following code beneath the `guard` statement:

```
let newAxis : UILayoutConstraintAxis
switch copyrightContentStackView.axis {
case .Horizontal:
  newAxis = .Vertical
case .Vertical:
  newAxis = .Horizontal
}
```

This simple switch statement checks the current value of the `axis` property, and then sets the `newAxis` constant to the other value.

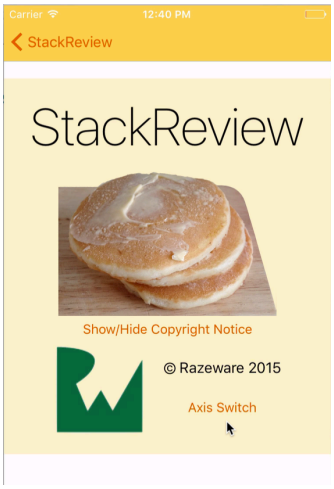
Finally you can animate to this new value. Add the following animation code to the end of the `switchCopyrightAxis` method:

```
UIView.animateWithDuration(0.7) {
  copyrightContentStackView.axis = newAxis
}
```

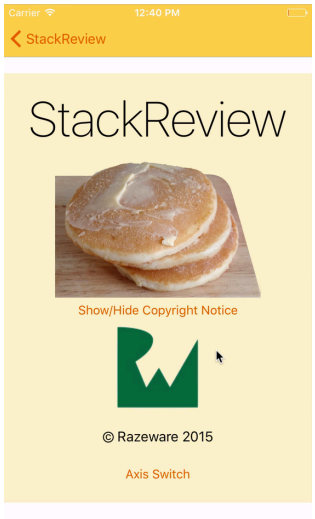
This is a very simple `UIView` animation block that sets the new axis value to the value you calculated. Remember that stack view is fully integrated with `UIView` animations, so this is all that you need to do to achive the desired effect.

Now build and run the app to try out your new functionality.

When you tap the "Show/Hide Copyright Notice" button, you'll see the new Axis Switch button appearing beneath the copyright label:



Now tap the Axis Switch button and observe the cool animation to the new layout:



Tap it again and it'll return to the original layout. That's pretty cool for just a few lines of code!