BEGINNING

CORE DATA

Hands-On Challenges

Beginning Core Data

Luke Parham

Notice of Rights

Notice of Liability

Trademarks

Challenge #4: Adding Collection View Sections

By Luke Parham

Adding Sections by Key Path

As you saw in the demo, a fetched results controller can take an optional `keyPath` argument by which you expect sections to be divvied up. Your challenge this time is to split this collection view up into sections based on people's eye colors.

To get started, open up the challenge starter and navigate to **ViewController.swift**. Go down to the `reloadData(_:)` and find the `fetchedResultsController` definition.

All you need to do is pass in a `keyPath` so replace the current definition with

```
    fetchedResultsController =
        NSFetchedResultsController(
            fetchRequest: fetchRequest,

managedObjectContext:appDelegate.persistentContainer.viewContext,
            sectionNameKeyPath: #keyPath(Person.eyeColor),
            cacheName: nil)
```

Build and run to see

Ok...not quite what you were expecting? The thing about using a fetched results controller to add your sections is that it expects you to be sorting the results by that same `keyPath`.

Go the the definition of `nameSortDescriptor` and add a second sort descriptor definition.

```
let eyeColorSortDescriptor =
    NSSortDescriptor(key: "eyeColor", ascending: true)
```

Next, replace the line that used to assign just the name sort descriptor with the following.

```
fetchRequest.sortDescriptors = [eyeColorSortDescriptor, sortDescriptor]
```

The important thing to note here is that the eye color sort descriptor is the first one in the array. Using it as a secondary sort would give you the same results as before! Go ahead and build and run to see your collection successfully sorted.

This is technically what you wanted, but it would be pretty nice to have some section headers.

Adding Section Headers

To add a header for each section, you'll need to set up a `UICollectionReusableView`.

First, add a new Swift file called **HeaderCollectionReusableView.swfit** of type `UICollectionReusableView`. Then replace the contents with the following.

```
import UIKit

class HeaderCollectionReusableView: UICollectionReusableView {
  @IBOutlet weak var headerLabel: UILabel!

}
```

Next, go **Main.storyboard** and open the document outline

Next, use the object library to drag and drop a collection reusable view onto the collection. Then add a `UILabel` to the header the same way. Finally, set the reusable view's class to `HeaderCollectionReusableView`.

Now that the storyboard has been set up, the last thing you need to do to connect it and the class you created is to add an outlet for the header title.

To do so, open **HeaderCollectionReusableView.swift** in the assistant editor. Cmd+click on the label you added and drag it over to the class to create the outlet.

The very last thing to do is to return the appropriate header for each section.

First, go to `viewDidLoad()` and add the following size and section inset to the flow layout that's defined there.

```
let width = UIScreen.main.bounds.size.width
layout.sectionInset = UIEdgeInsets(top: 16, left: 0, bottom: 16, right:
0)
layout.headerReferenceSize = CGSize(width:width, height: 44)
```

Then, go down to the collection view data source extension and add the following method.

```
  func collectionView(_ collectionView: UICollectionView,
      viewForSupplementaryElementOfKind kind: String,
      at indexPath: IndexPath) -> UICollectionReusableView {
    let header = collectionView.dequeueReusableSupplementaryView(ofKind:
```

```
  kind, withReuseIdentifier: "Header", for: indexPath) as!
  HeaderCollectionReusableView

      if let peopleInSection = fetchedResultsController.sections?
  [indexPath.section].objects as? [Person],
         let firstPerson = peopleInSection.first {
        header.headerLabel.text = "\
  (stringFromEyeColor(firstPerson.eyeColor)) Eyed People"
      }

      return header
    }
```

Here you're just using the first result in that section to decide what the color of that section is.