sitepoint premium

Handout

Introduction to Swift
Lesson 4 - Step 2
Brett Romero

Intro

We’re going to get into our performance view, where we have our second scene. We’ll add our total times to our activities.

ActivityName

We’ll use the `activityName` and append time using string concatenation syntax.

```
    func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITabl
        let cell:UITableViewCell = UITableViewCell(style: UITableViewCellStyle.Default, reuseIdent
        var activity = ActivityManager.activities[indexPath.item]
        cell.textLabel!.text = "\(activity.activityName) \(activity.totalTime)"
        return cell
    }
```

Now we are not refreshing our performance table, that’s why the new times that are getting logged aren’t coming through. So we can make use of `viewDidLoad` just like we did in our first view controller. We’re going to use a similar technique, `viewDidAppear`. It will trigger whenever the view displays. That way, when we leave the performance view and go back to scene one, all of the activities that we loaded in the performance view will then displaying scene one because we reload the picker view, which picks up those new activities.

```
override func viewDidAppear(animated: Bool) {
    super.viewDidAppear(animated)
    tableView.reloadData()
}
```

Now highlight the table view and create an IBOutlet. Call it Table View. This is very similar to what we did before. Now we’ve got a reference to our table view, because remember, we want to load the table view and if we tried to do it earlier, we didn’t have any reference. Now we are just using `reloadData` to do the trick.