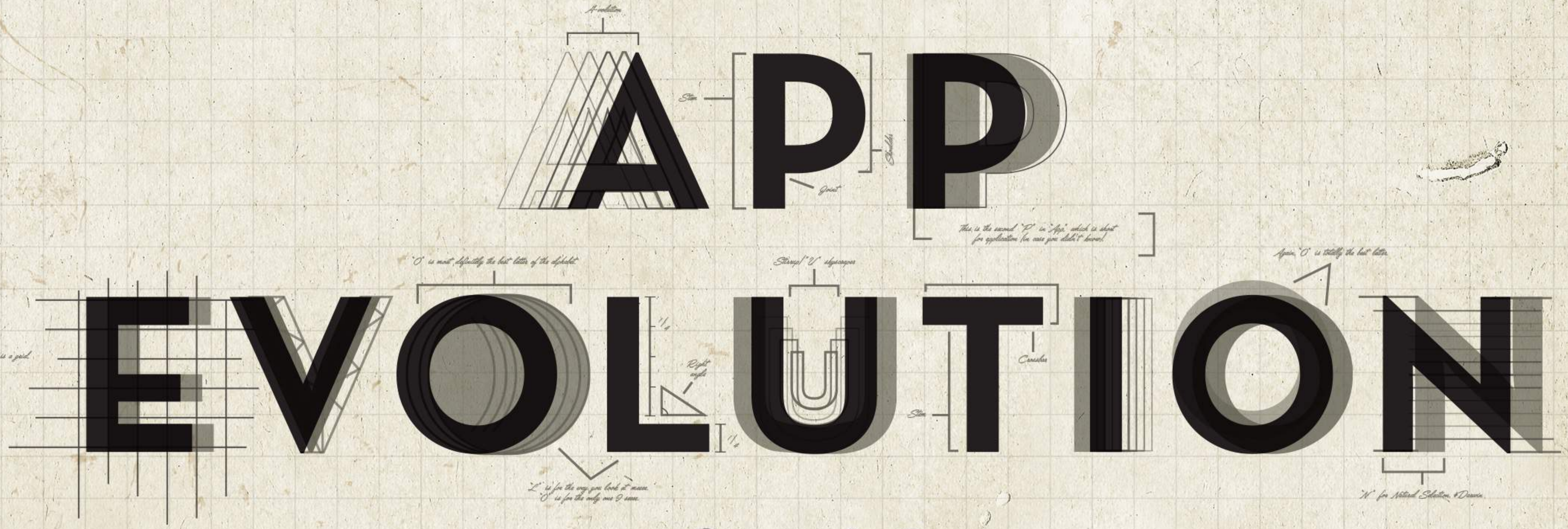

A-volution
Stem
Joint
Shoulder
This is the second "P" in "App," which is short for application (in case you didn't know!)
"O" is most definitely the best letter of the alphabet.
So uhh, "this is a grid.
1/4
Right angle
1/4
"L" is for the way you look at meeee.
"O" is for the only one I seeee.
Stirrup/"U" skyscraper
Crossbar
Stem
Again, "O" is totally the best letter.
"N" for Natural Selection. #Darwin.

APP EVOLUTION

with Swift

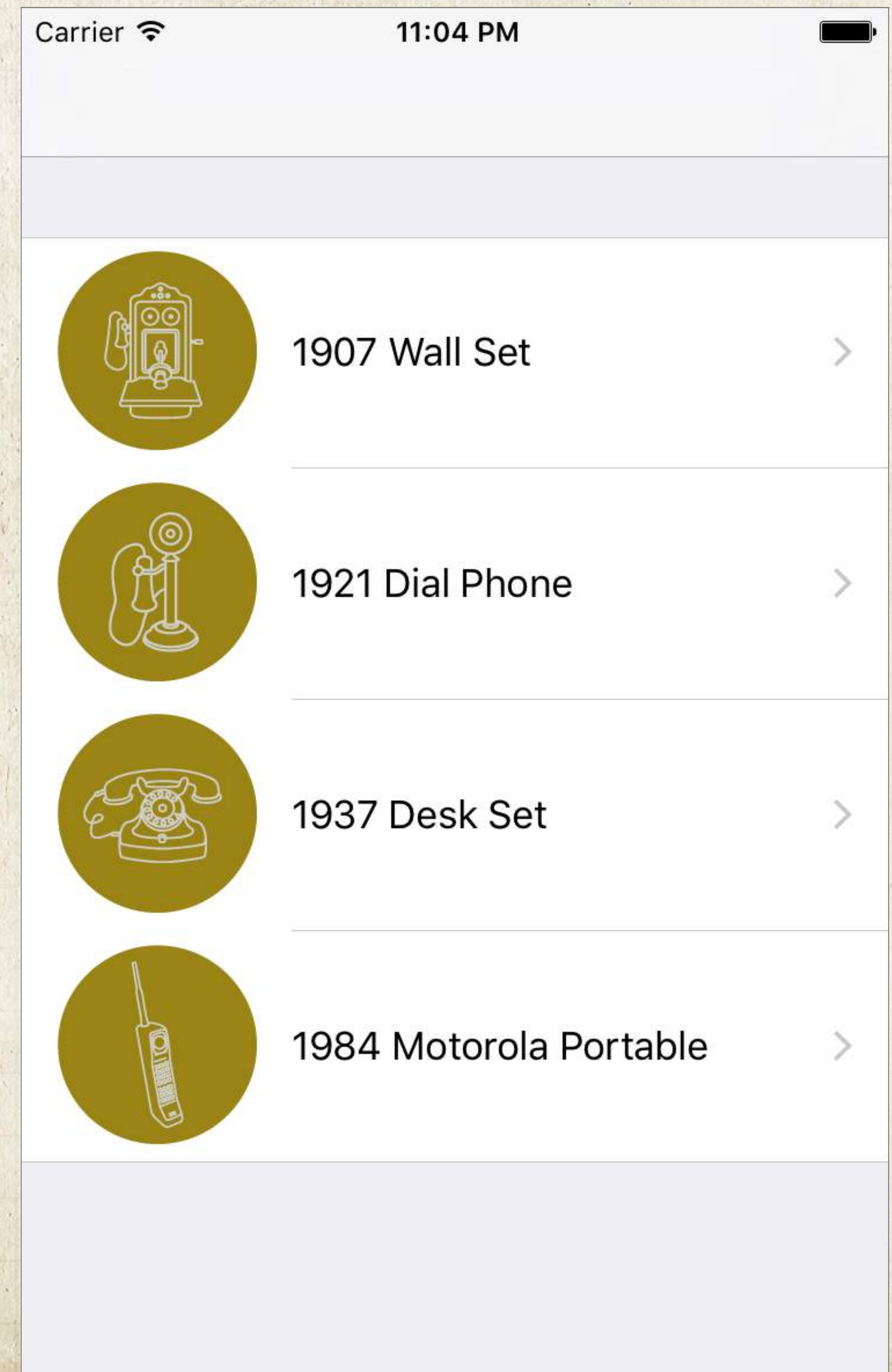
Level 4

Table Views

Section 1 - Creating a Table View

APP EVOLUTION with Swift

Demo: Using a Table View to Display a List of Things



1907 Wall Set

1921 Dial Phone

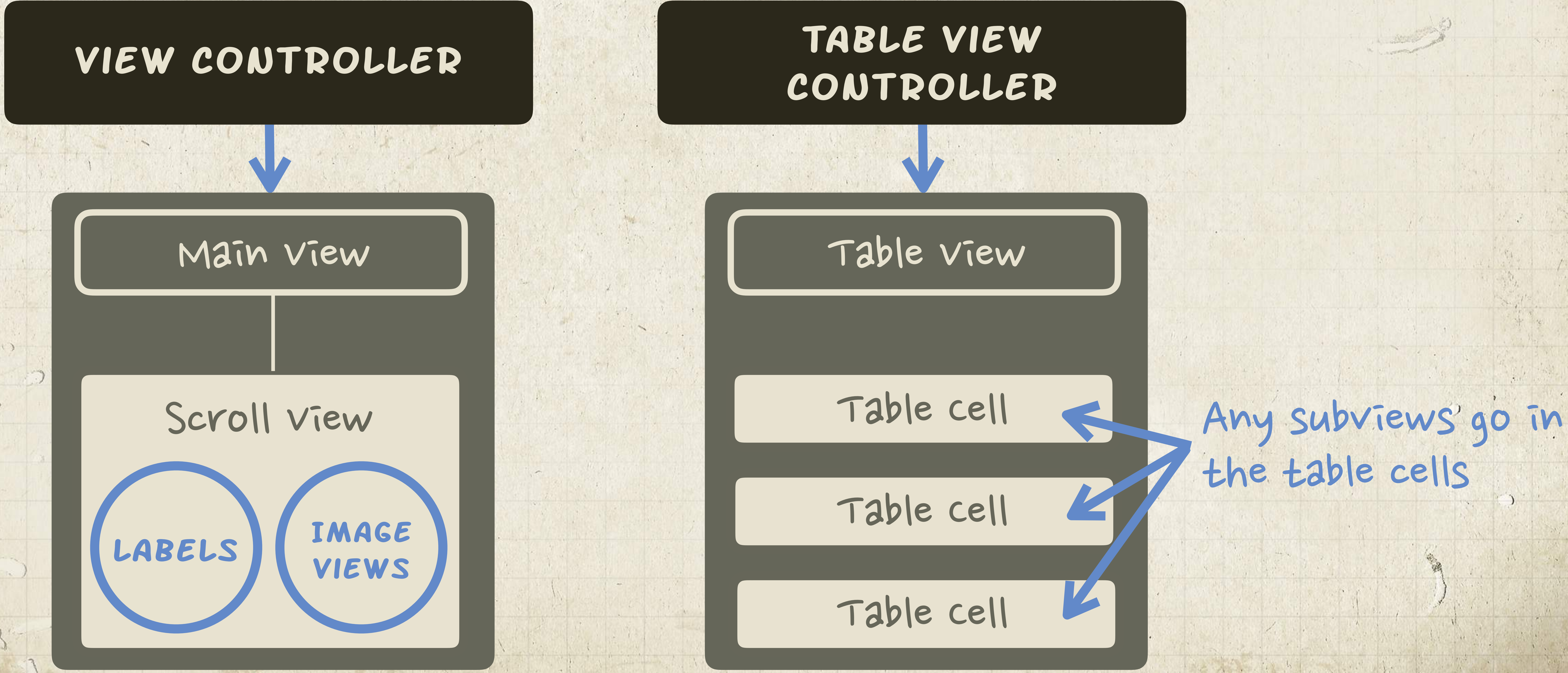
1937 Desk Set

1984 Motorola Portable

APP EVOLUTION with Swift

A Table View Controller’s Hierarchy

A Table View Controller is just a special View Controller that has support for working with Table Views built-in

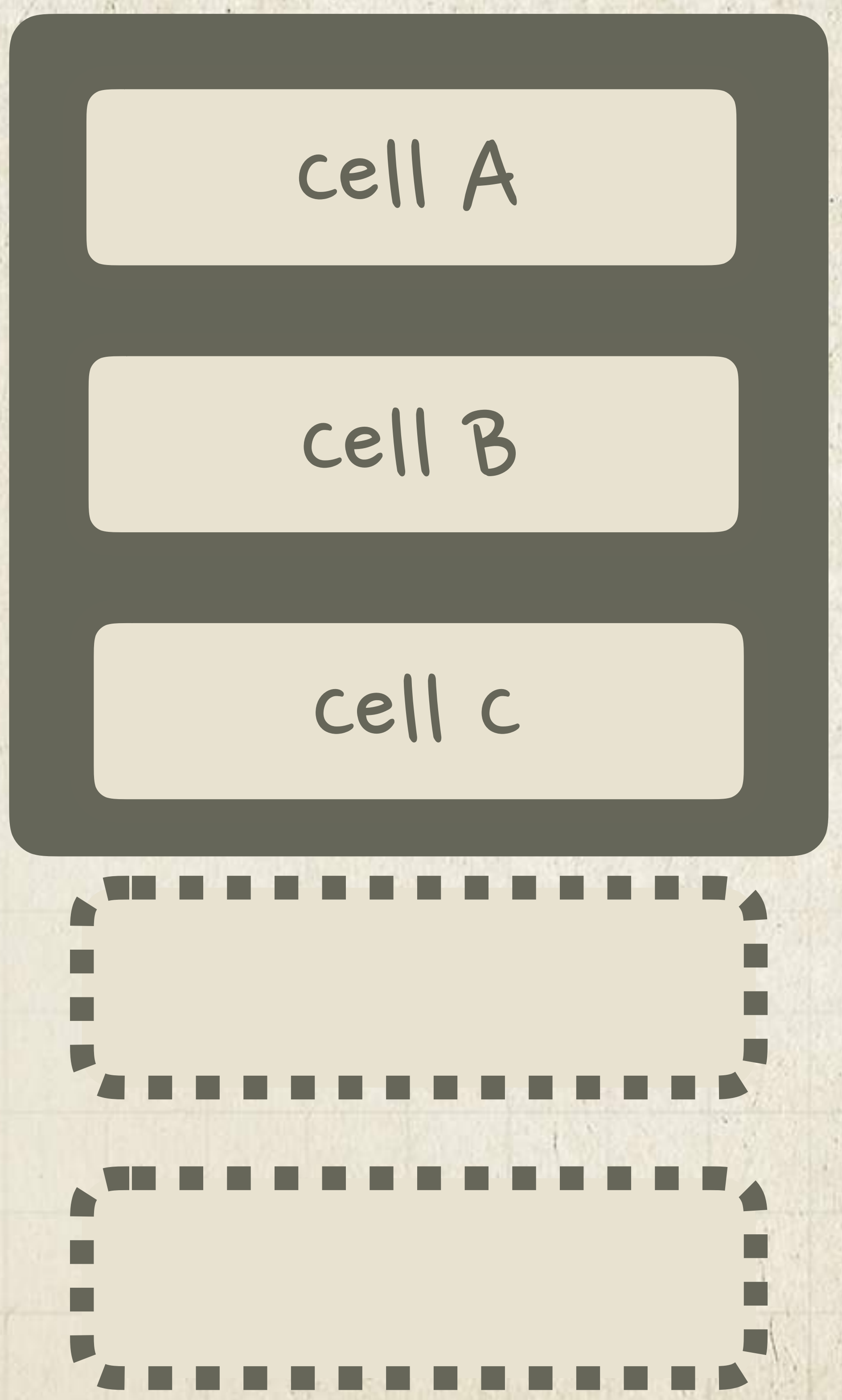


Screencast: Creating a Table View Controller

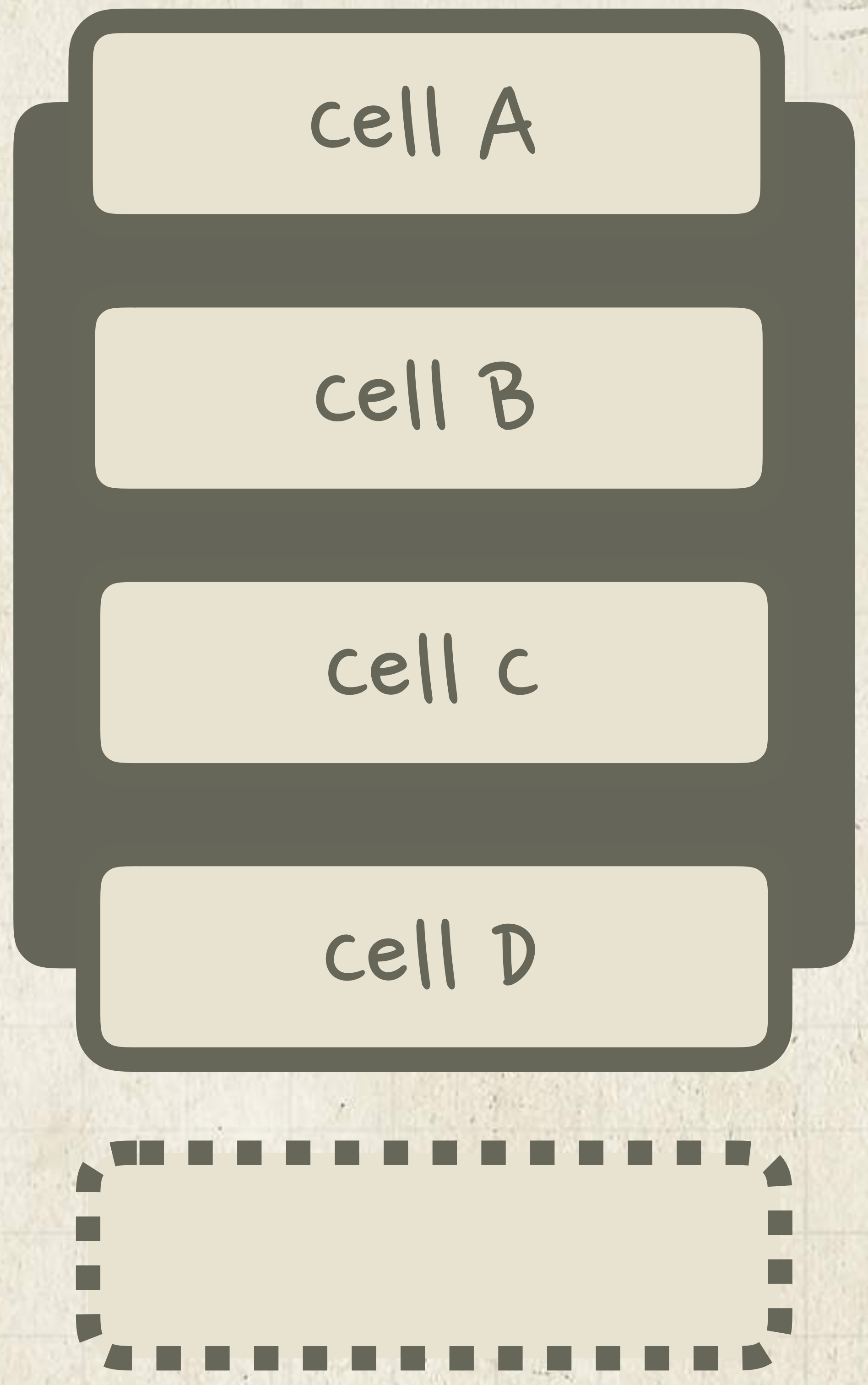
How Table View Cell Recycling Works

Table cells are recycled as they scroll offscreen, and any new cells are just recycled cells filled in with new data.

Any visible cells are created on the initial load of the table view

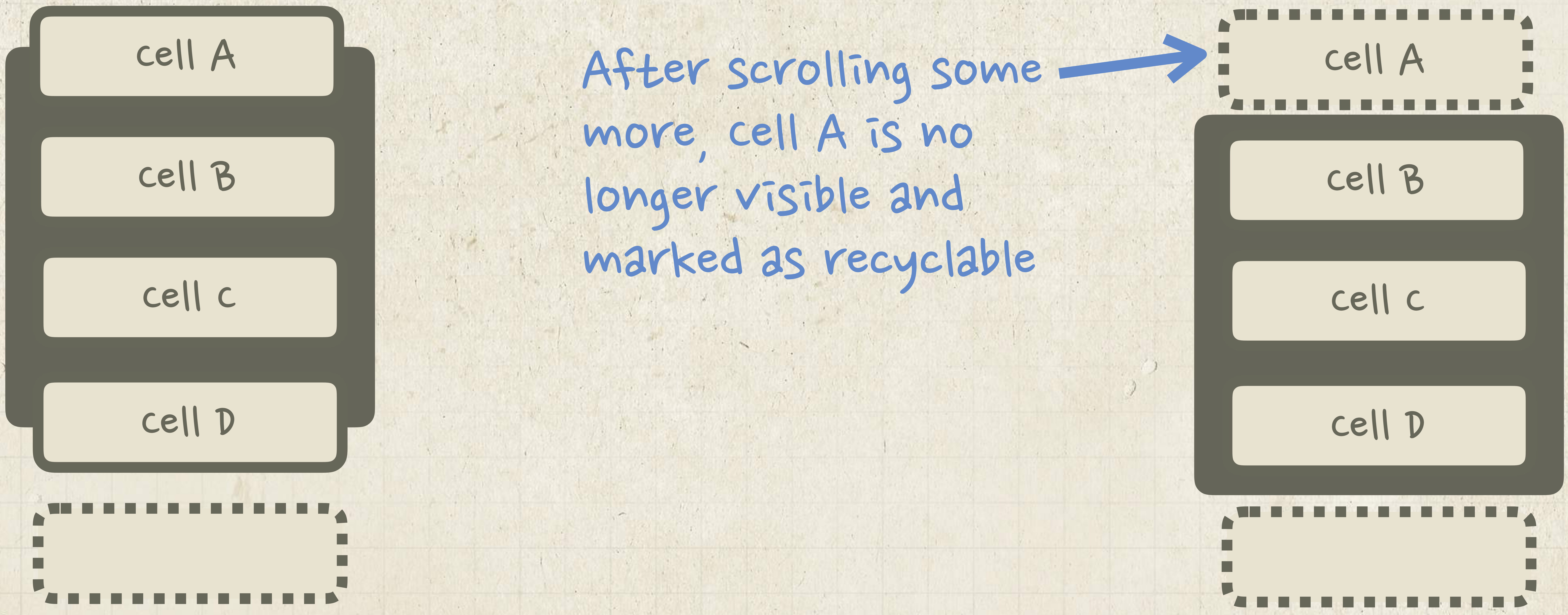


As the table view scrolls, now four unique cells are visible



How Table View Cell Recycling Works

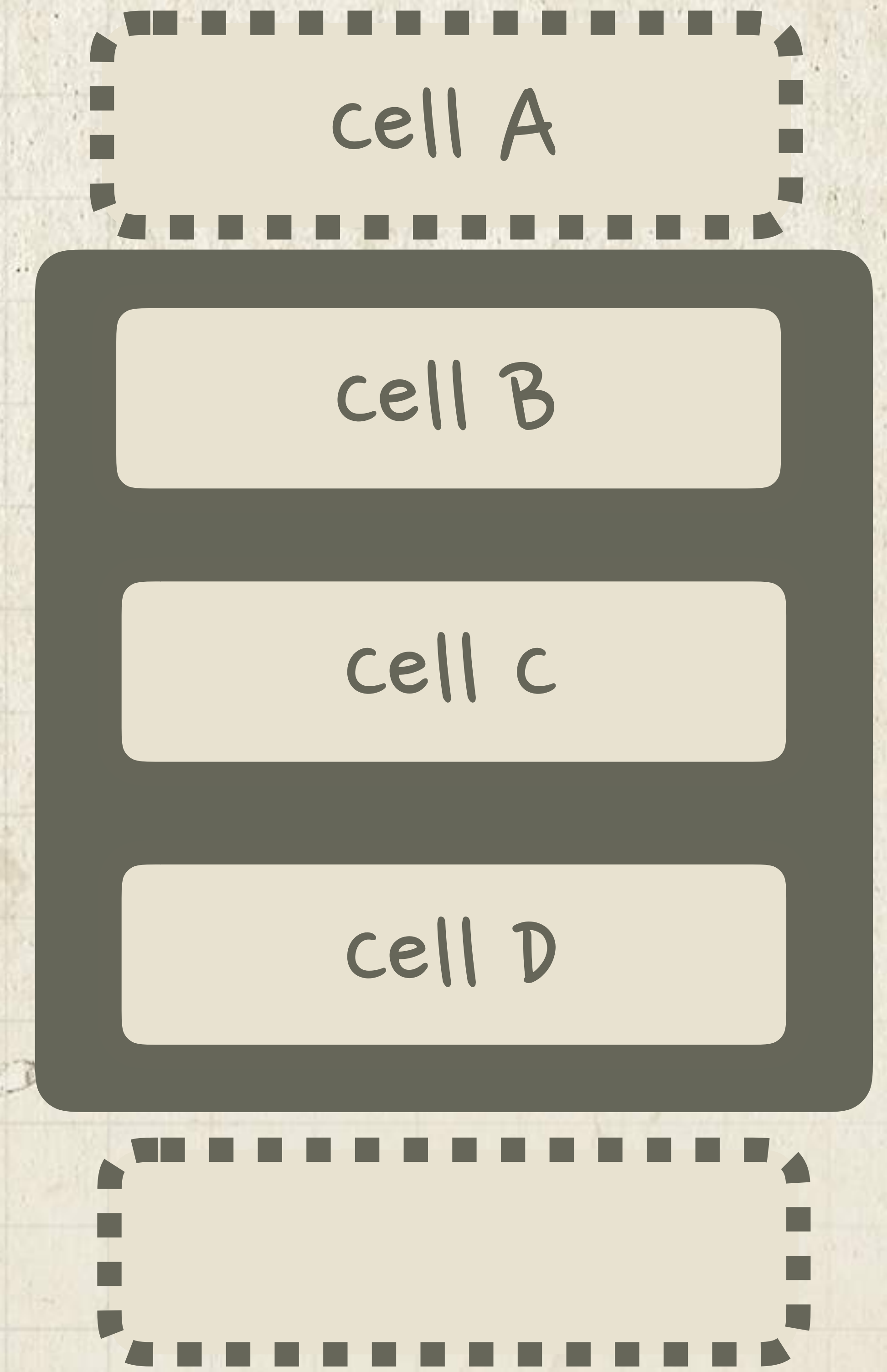
Table cells are recycled as they scroll offscreen, and any new cells are just recycled cells filled in with new data.

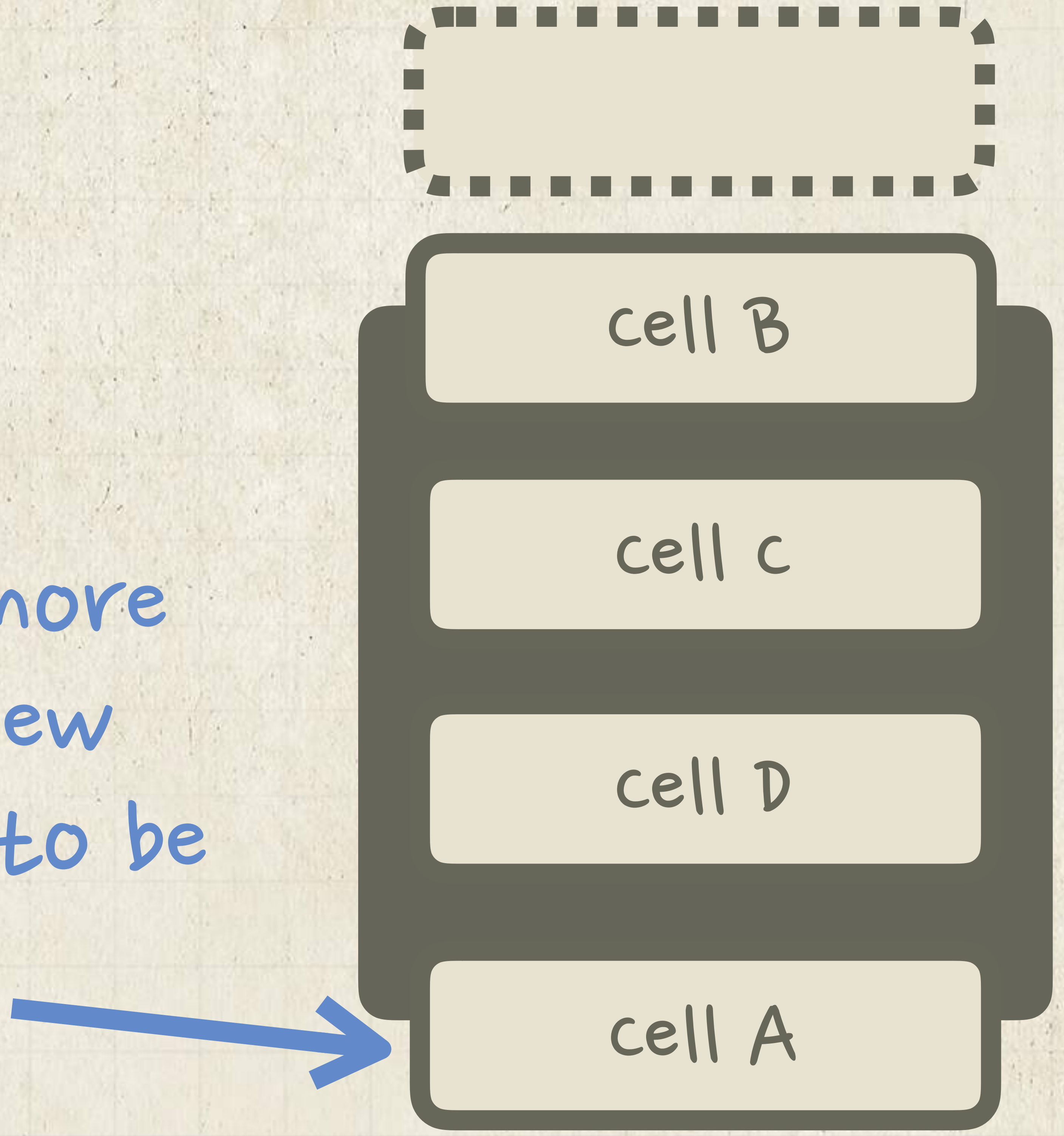


After scrolling some more, cell A is no longer visible and marked as recyclable

How Table View Cell Recycling Works

Table cells are recycled as they scroll offscreen, and any new cells are just recycled cells filled in with new data.





After even more scrolling, a new cell is ready to be displayed

There's no need for a brand new cell because cell A would be recycled when it went offscreen

Exploring the Table View Controller Swift File

ProductsTableViewController.swift

```swift
import UIKit

class ProductsTableViewController: UITableViewController {

}
```

This is a view controller that has access to table-specific functions

The controller runs some of those functions automatically:

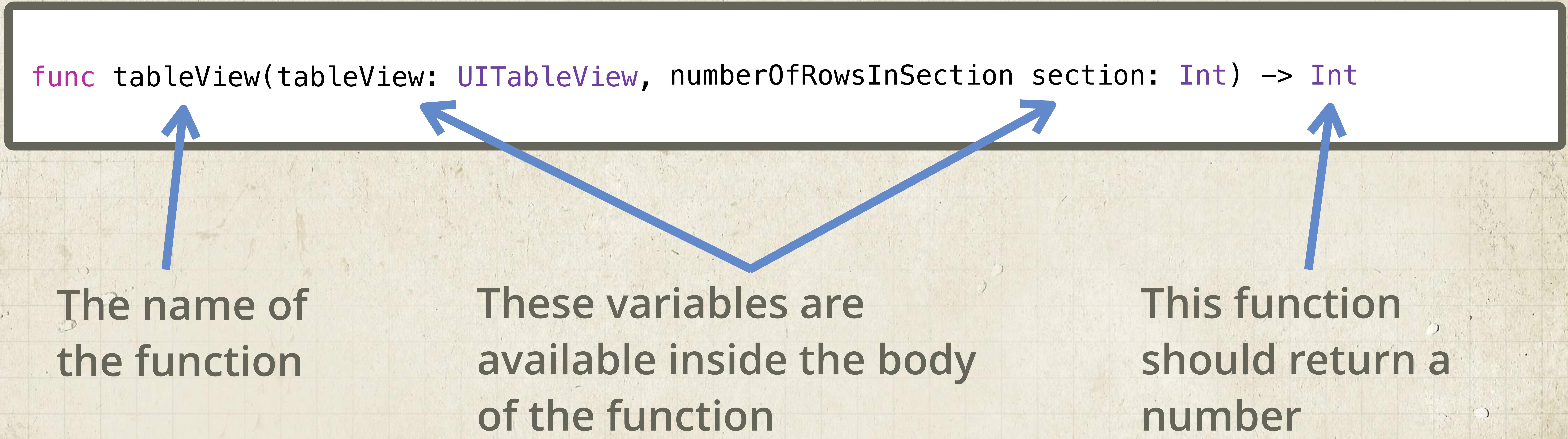
```swift
func tableView(tableView: UITableView,
    numberOfRowsInSection section: Int) -> Int
```

Set the number of rows

create one cell for each row

```swift
func tableView(tableView: UITableView,
    cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell
```

How to Read Function Declarations



```
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int
```

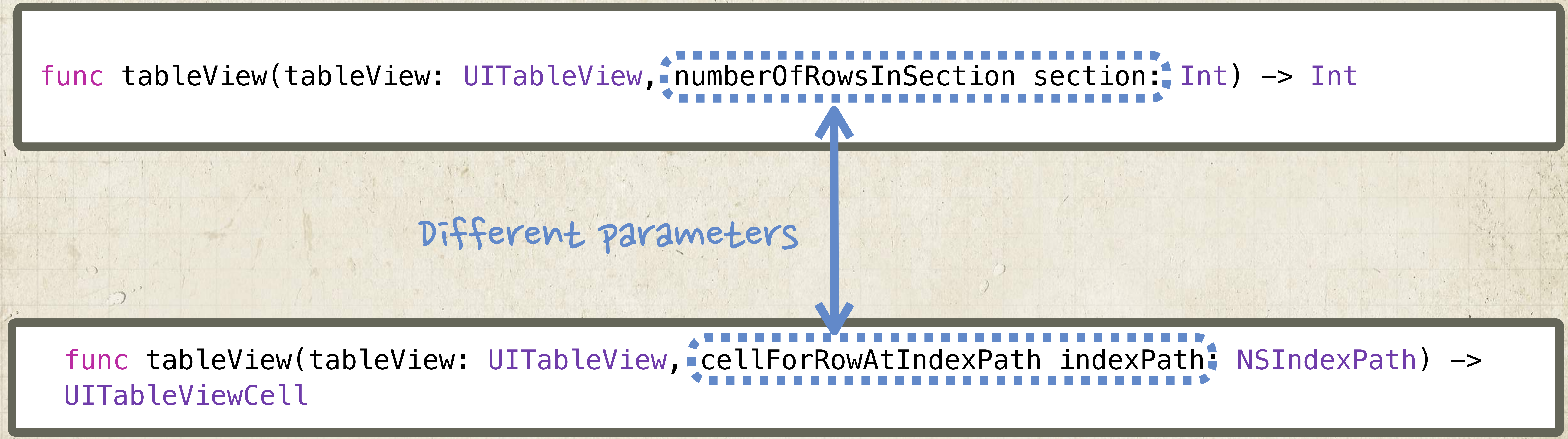
The name of the function

These variables are available inside the body of the function

This function should return a number

Using the Same Function Name Two Different Ways

Though the functions have the same name, since they have different parameters they are two different functions.

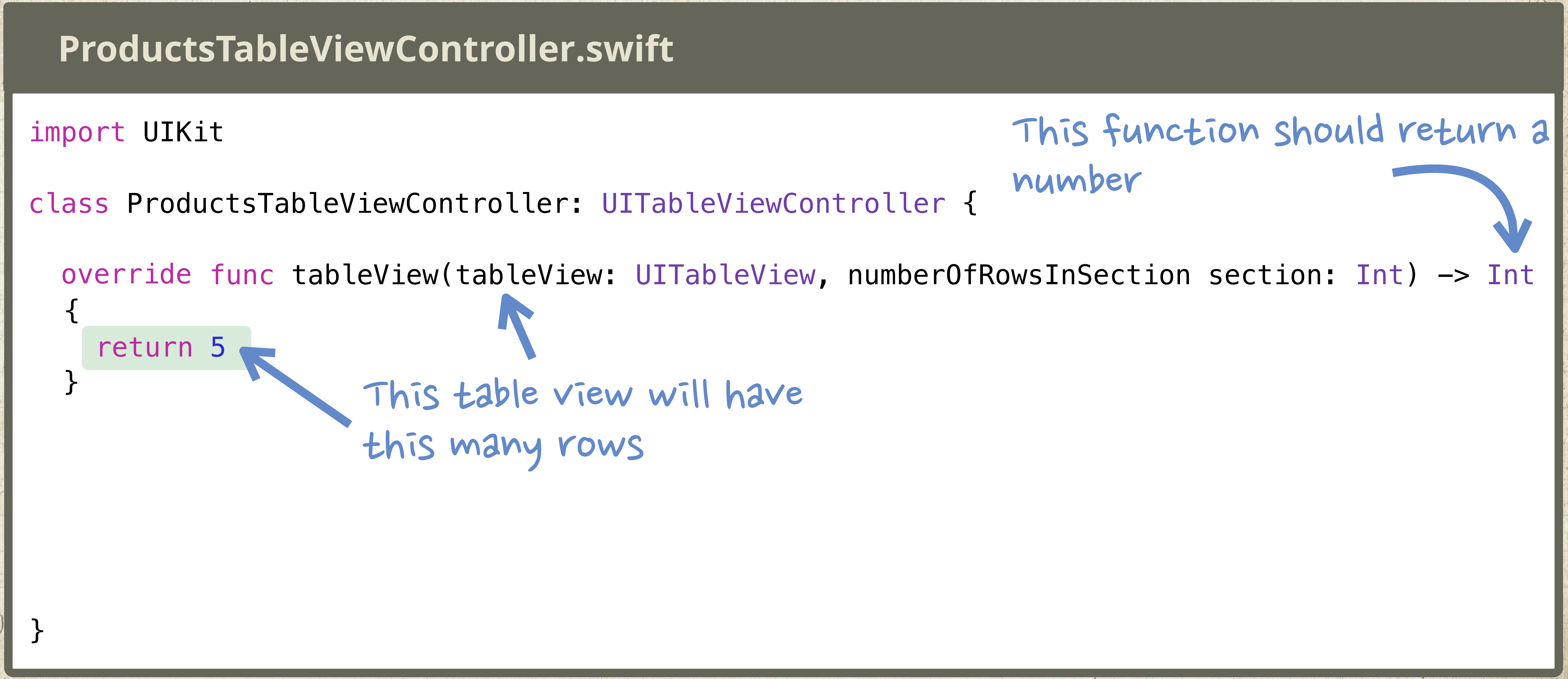


```
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int
```

Different parameters

```
func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell
```

Setting the Number of Rows in the Table



ProductsTableViewController.swift

```swift
import UIKit

class ProductsTableViewController: UITableViewController {

  override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int
  {
    return 5
  }



}
```

This function should return a number

This table view will have this many rows

Adding the Function That Will Create Cells

The cellForRowAtIndexPath method is used to set up the cell that appears in each row

ProductsTableViewController.swift

```swift
import UIKit

class ProductsTableViewController: UITableViewController {

  override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int
  { ... }


  override func tableView(tableView: UITableView,
    cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell
  {

  }
}
```

Runs one time for each row (5 times in our case)

This function should return a cell

Creating a Cell for Each Row

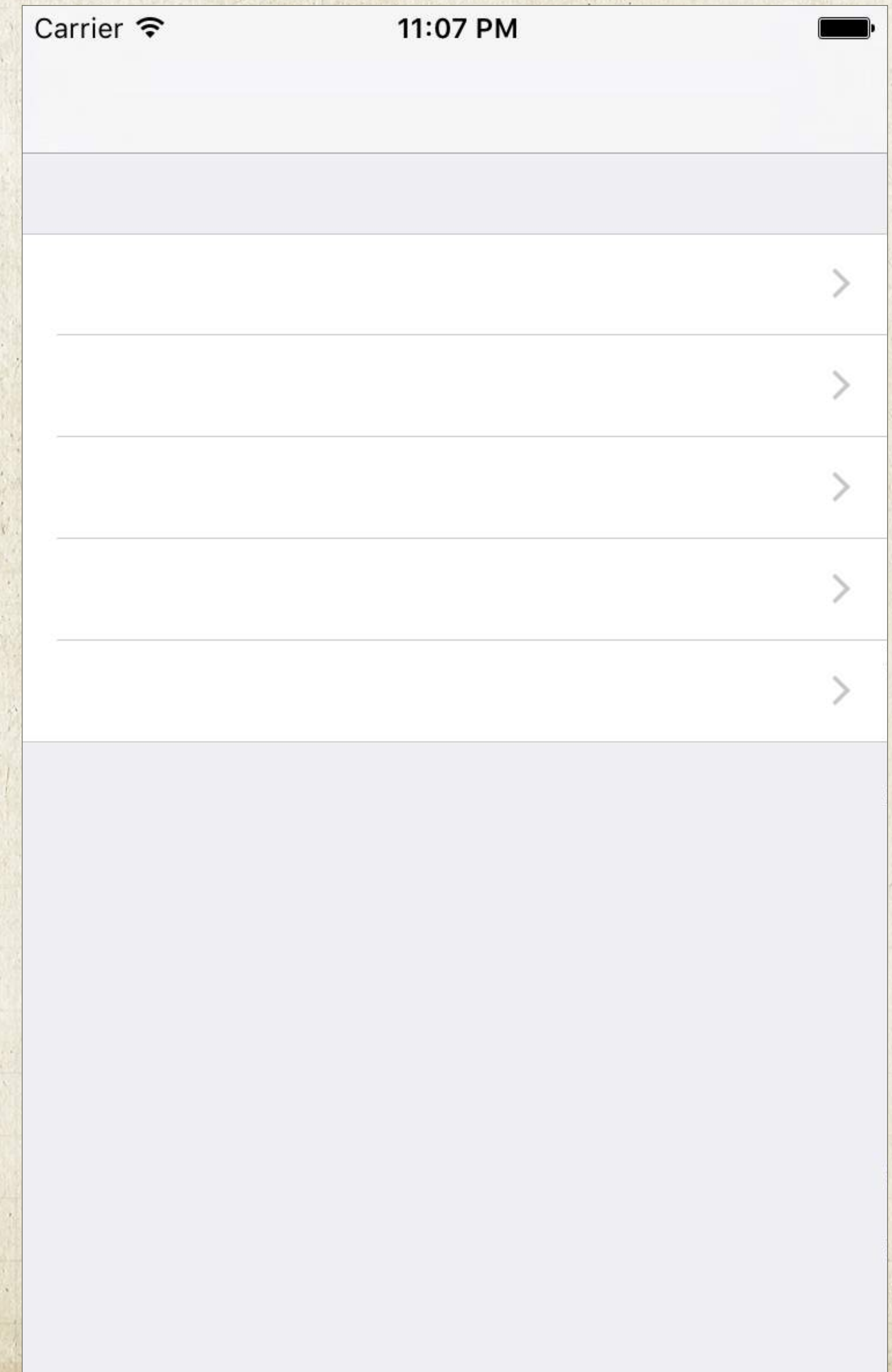
ProductsTableViewController.swift

```
import UIKit

class ProductsTableViewController: UITableViewController {

  override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int
  { ... }

  override func tableView(tableView: UITableView,
    cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell
  {
    let cell = tableView.dequeueReusableCellWithIdentifier("ProductCell",
                                                           forIndexPath: indexPath)

    return cell
  }
}
```

This matches the identifier we added in the Storyboard

Return the created cell

Demo: Table View With Cells



Carrier
11:07 PM