

Level 4

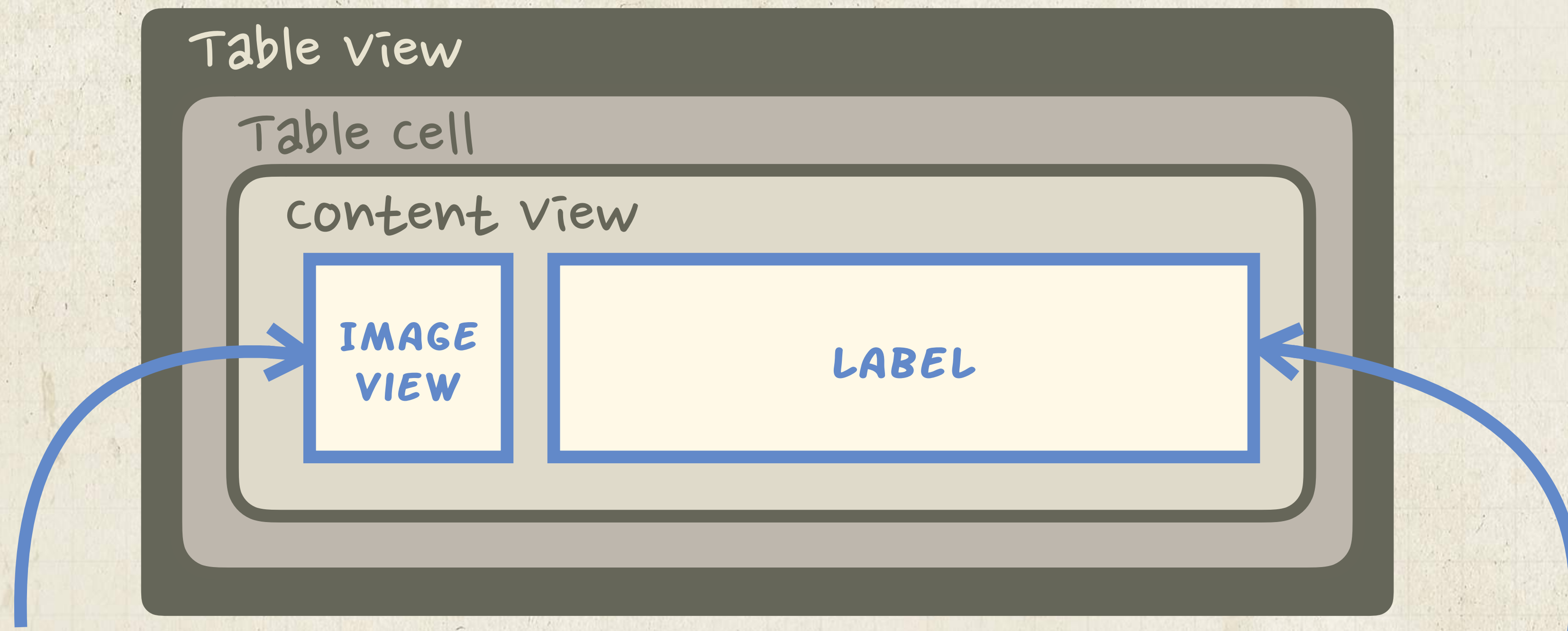
Table Views

Section 2 - Displaying Data in Table Cells



Table View Cells Come Pre-loaded With Subviews

Standard table cells come with an image view and label already wired up



This subview is already linked with the name **imageView**

This subview is already linked with the name **textLabel**

Creating a Cell for Each Row

ProductsTableViewController.swift

```
import UIKit

class ProductsTableViewController: UITableViewController {
    ...

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

        cell.textLabel.text = "Hello friend."

        return cell
    }
}
```



Let's see how we can fix this error

❗ Value of optional type 'UILabel?' not unwrapped; did you mean to use '!' or '?'?

A Quick Look Into the Table View Cell Docs

Apple docs for UITableViewCell

`textLabel` Property

Returns the label used for the main textual content of the cell.

Declaration

SWIFT

```
var textLabel: UILabel? { get }
```

Here's the official documentation for the cell's `textLabel`.

This means it's **optional**.

It may or may not exist

First, Consider This Situation

It's possible to crash your app if you try to access an object or property that doesn't exist.

It's possible to create a custom cell without a `textLabel`

```
cell.textLabel.text = "Hello friend"
```

So setting it will crash the app!

That would mean this property wouldn't exist...

which means this text property wouldn't exist...

Using a conditional to check for an optional value

One way to work with optionals is to use a conditional to first check if a value exists, and if it does, continue running code

```
if cell.textLabel != nil {  
    cell.textLabel.text = "Hello friend"  
}
```

But there's a great shortcut...



Using Optional Chaining

Optional chaining is a way to check optionals without writing conditionals

```
cell.textLabel?.text = "Hello friend"
```

Adding this question mark after optional properties means "First check if this exists"

textLabel exists

→ continue running the line of code

textLabel doesn't exist

→ stop running the line of code



Setting the textLabel of a Cell

ProductsTableViewController.swift

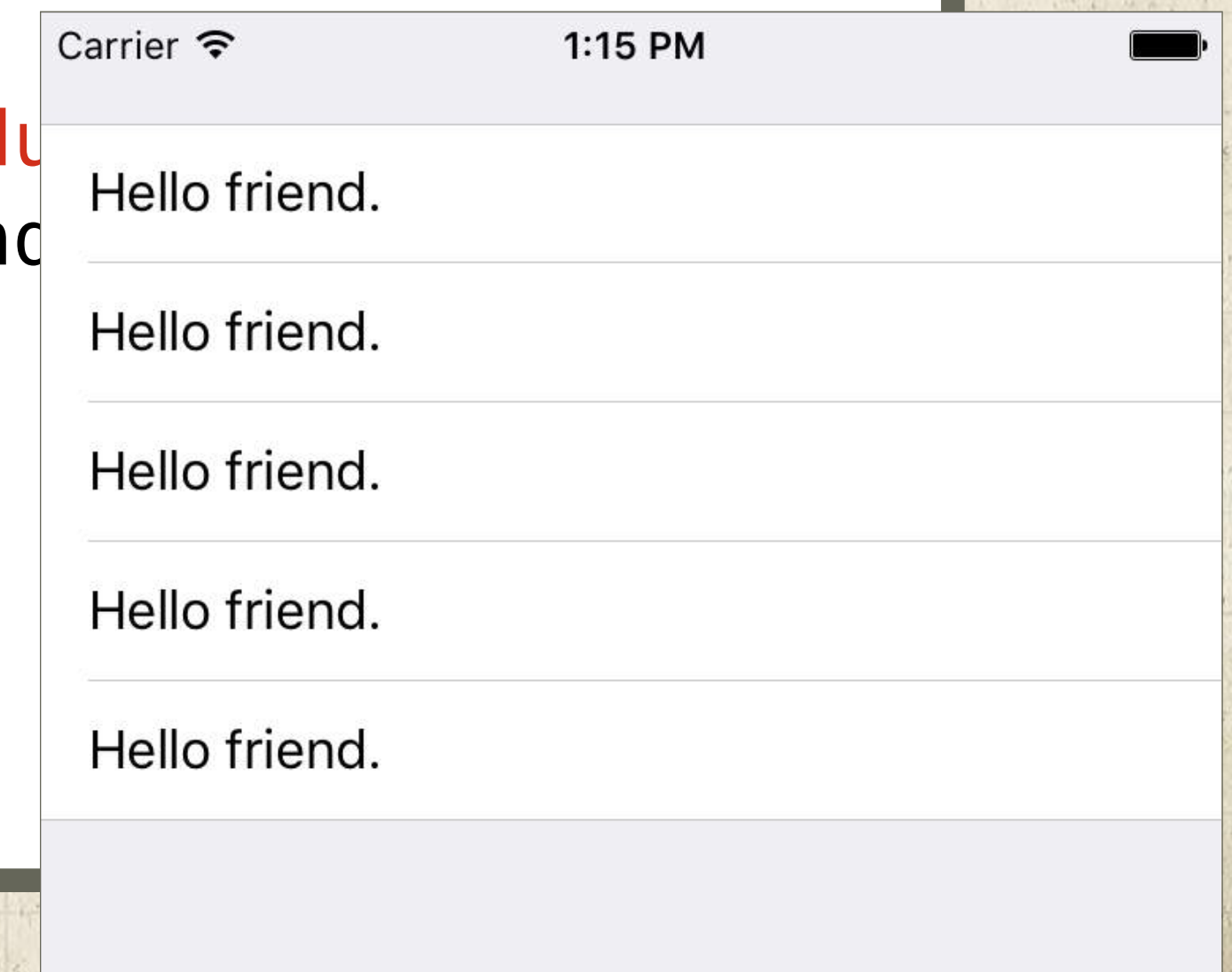
```
import UIKit

class ProductsTableViewController: UITableViewController {
    ...

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

        cell.textLabel?.text = "Hello friend"

        return cell
    }
}
```



Setting the imageView of a Cell

ProductsTableViewController.swift

```
import UIKit

class ProductsTableViewController: UITableViewController {
    ...

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

        cell.textLabel?.text = "Hello friend"
        cell.imageView?.image = UIImage(named: "image-cell1")

        return cell
    }
}
```

We've imported a few more
images into the asset catalog

Carrier 1:18 PM



Hello friend.



Hello friend.



Hello friend.



Hello friend.



Hello friend.

Screencast: Adjusting the Table Cell Height

