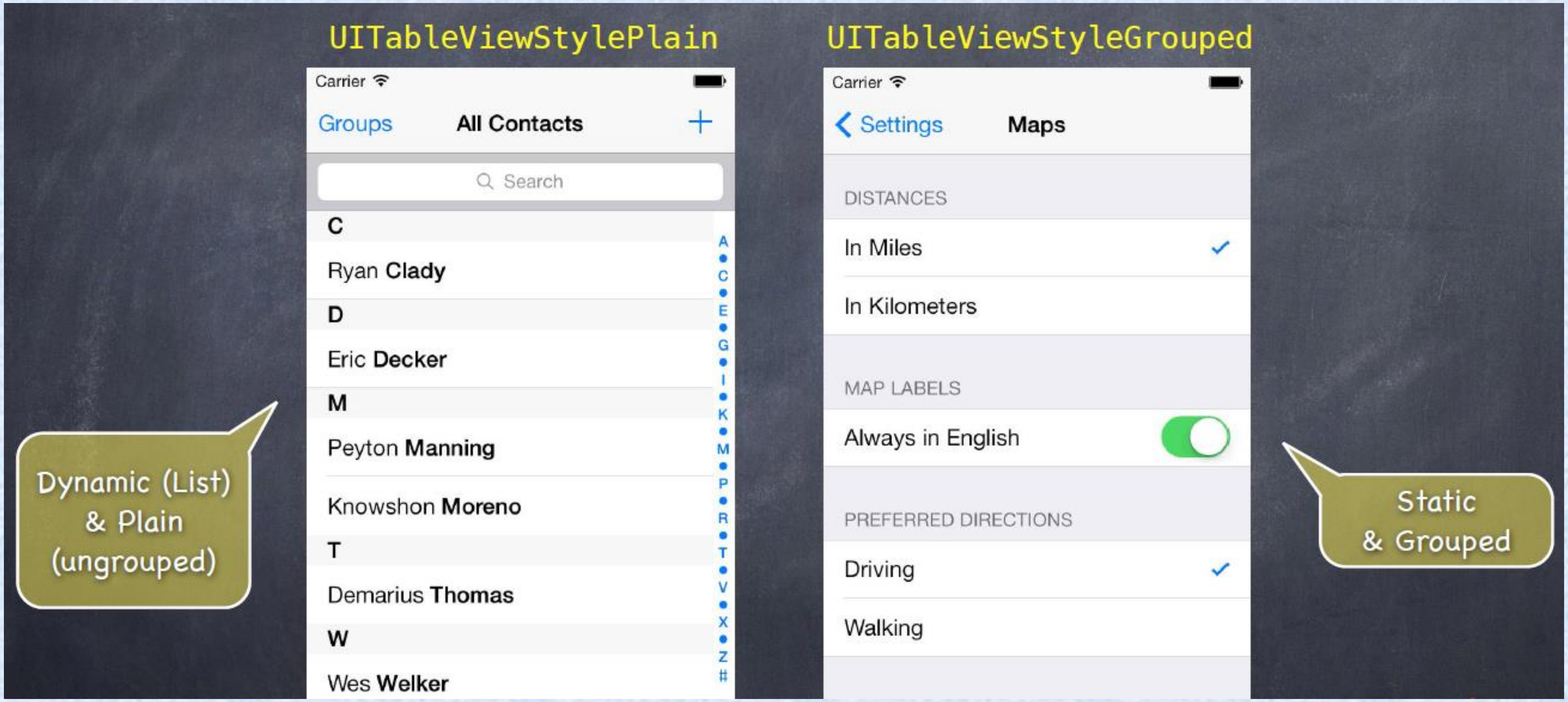
表视图

一、什么是表视图



Carrier

Groups　All Contacts　+

Search

C
Ryan Clady
D
Eric Decker
M
Peyton Manning
Knowshon Moreno
T
Demarius Thomas
W
Wes Welker

A • C • E • G • I • K • M • P • R • T • V • X • Z #

Carrier

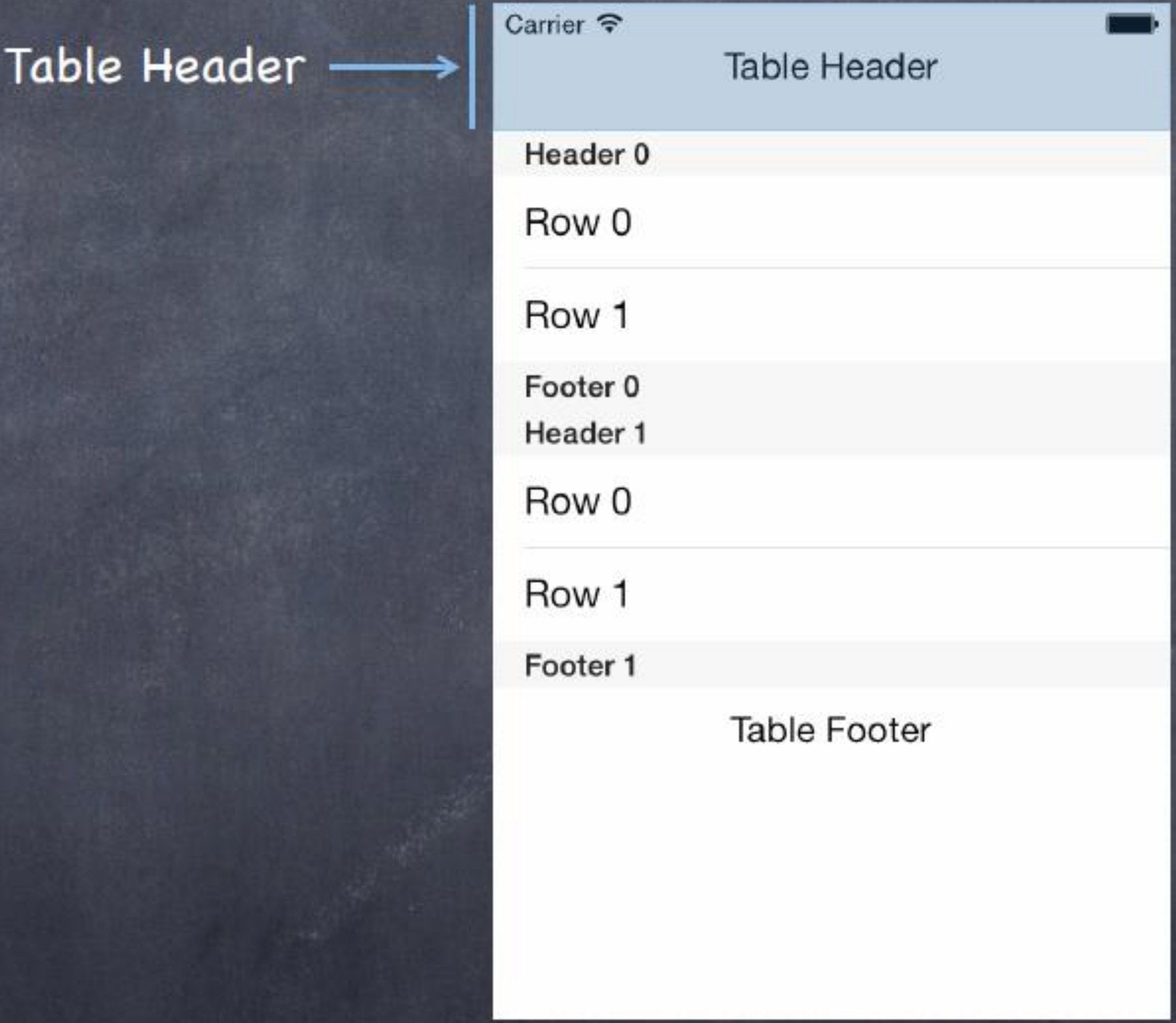
Settings　Maps

DISTANCES
In Miles ✓
In Kilometers

MAP LABELS
Always in English

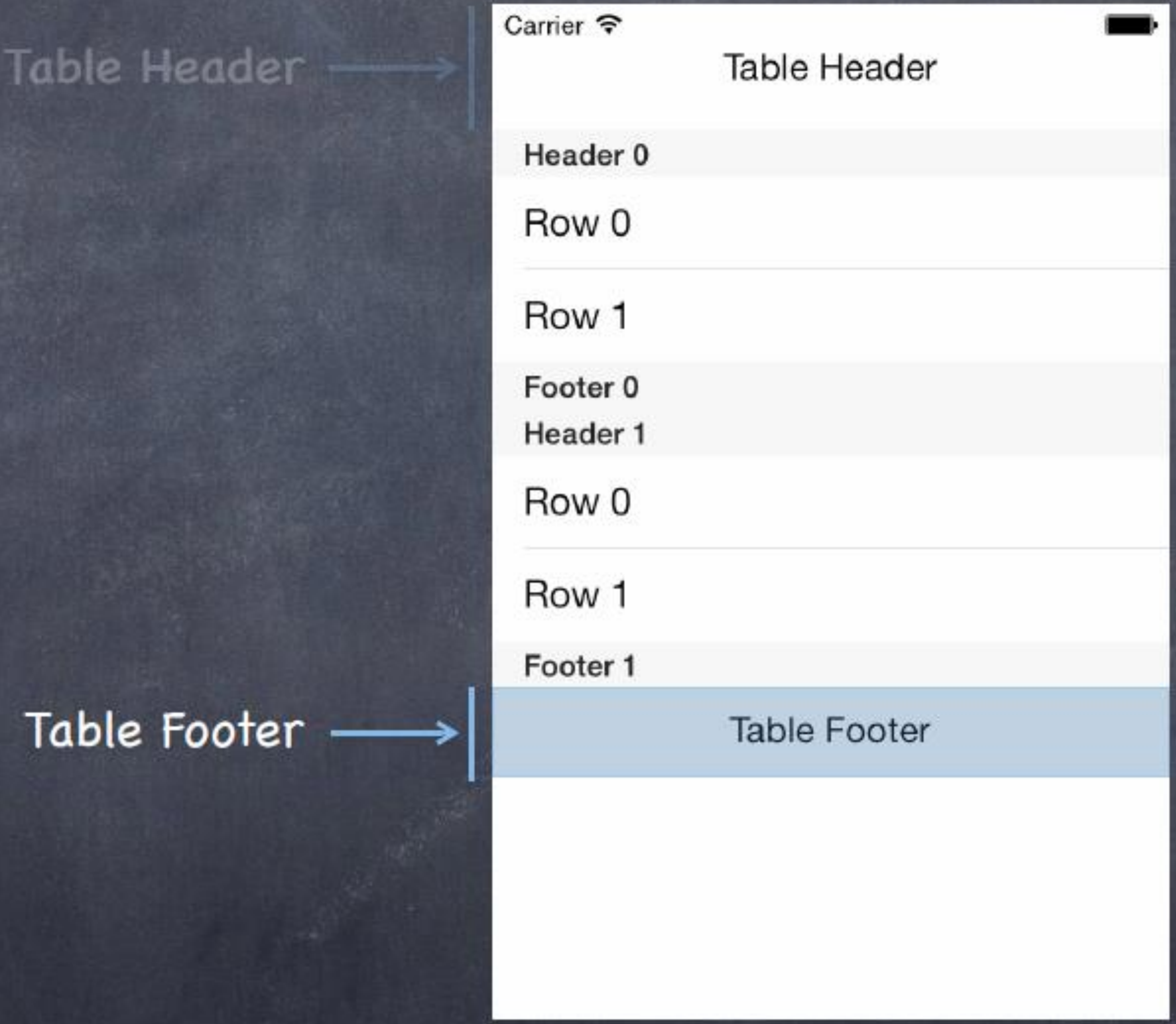
PREFERRED DIRECTIONS
Driving ✓
Walking

Plain Style

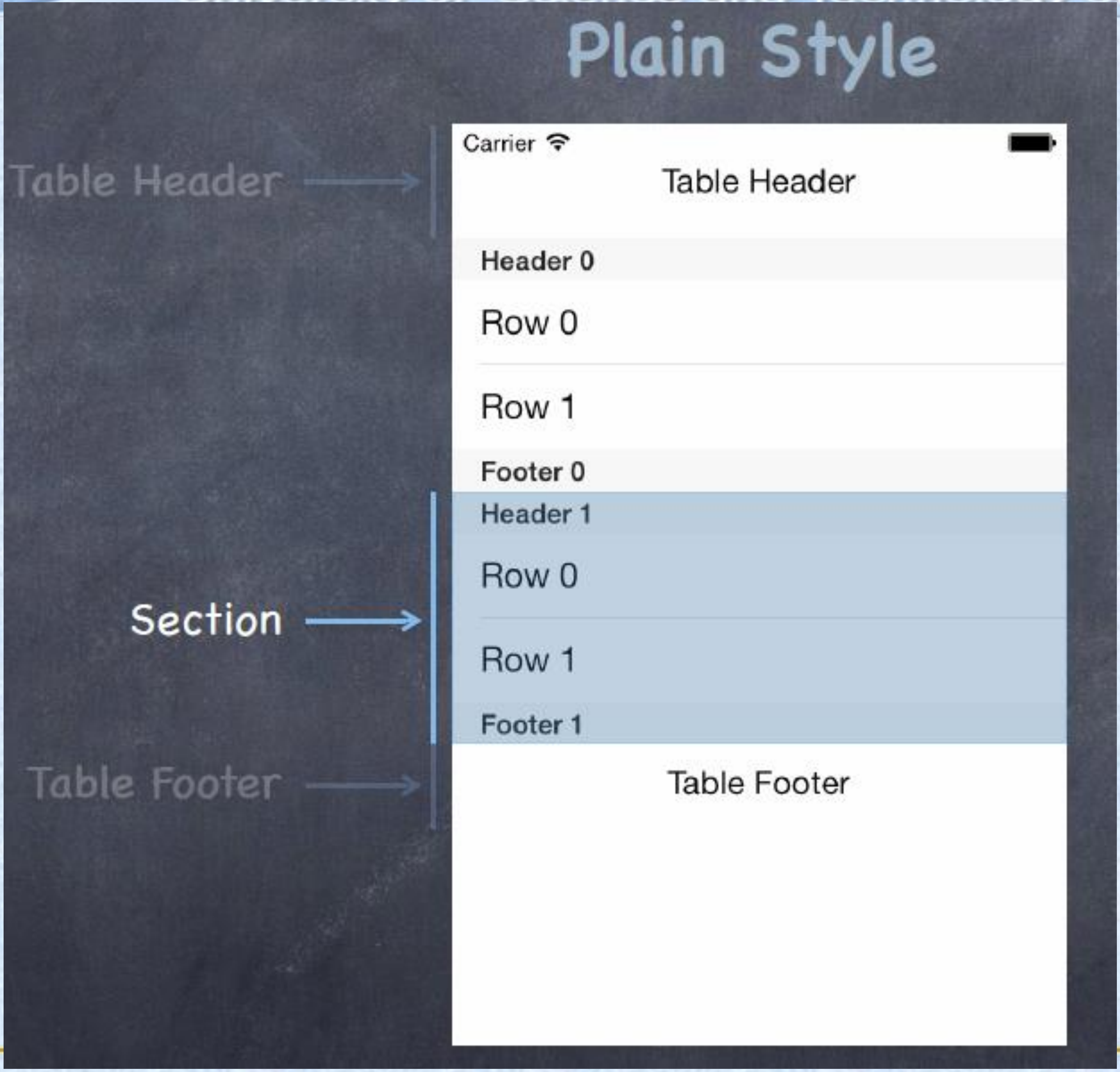


```
@property UIView *tableHeaderView;
```

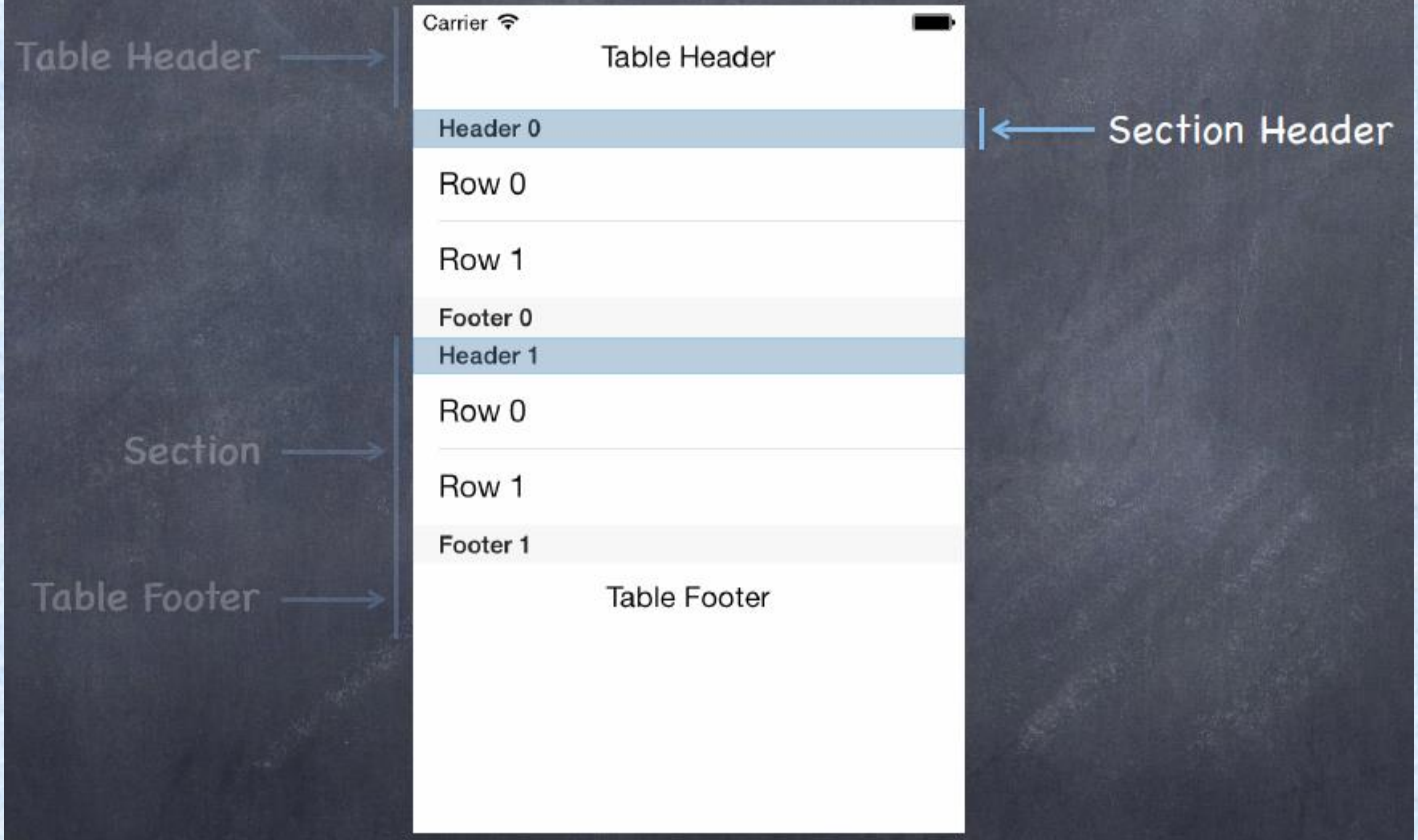
Plain Style



```
@property UIView *tableFooterView;
```

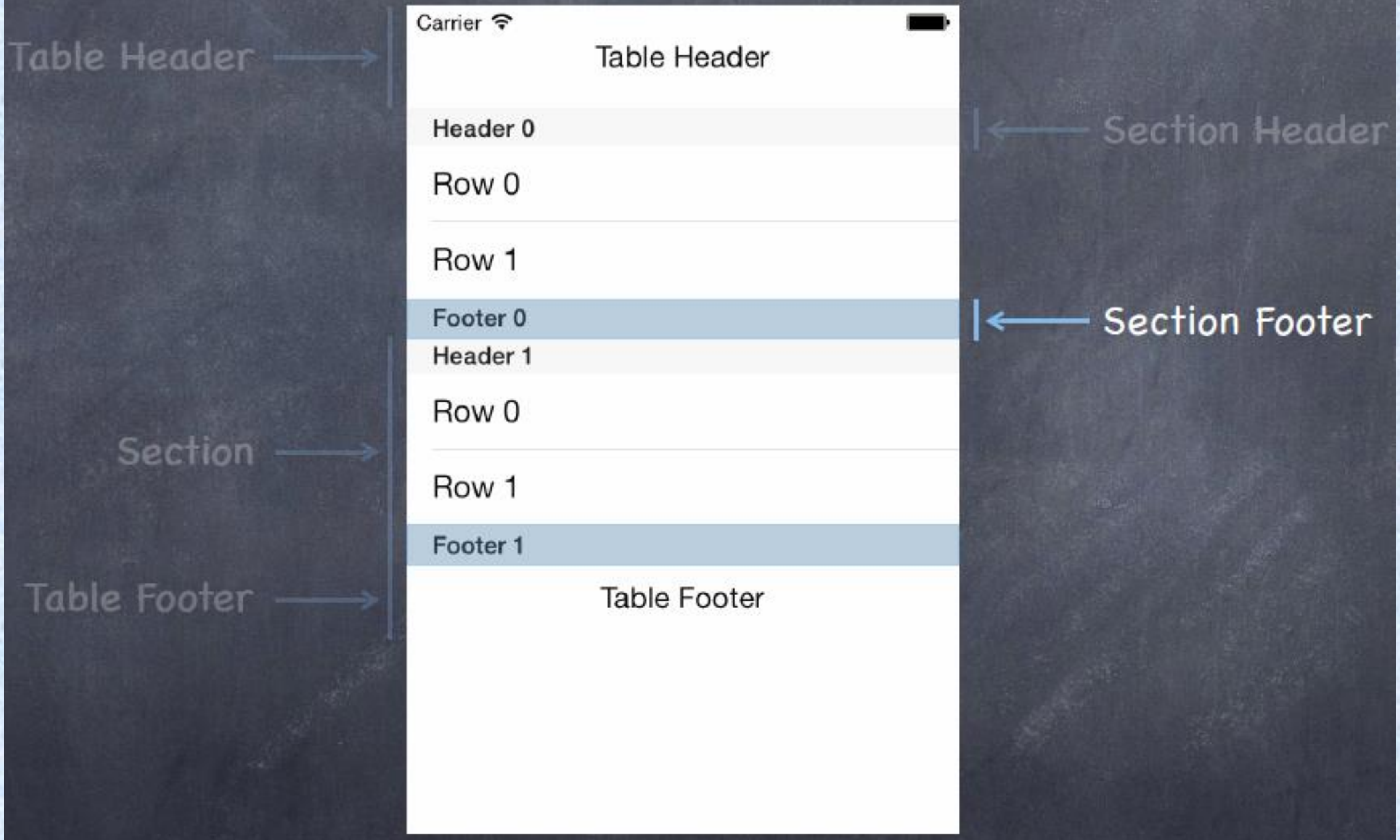

Plain Style
Table Header
Section
Table Footer
Carrier
Table Header
Header 0
Row 0
Row 1
Footer 0
Header 1
Row 0
Row 1
Footer 1
Table Footer

Plain Style



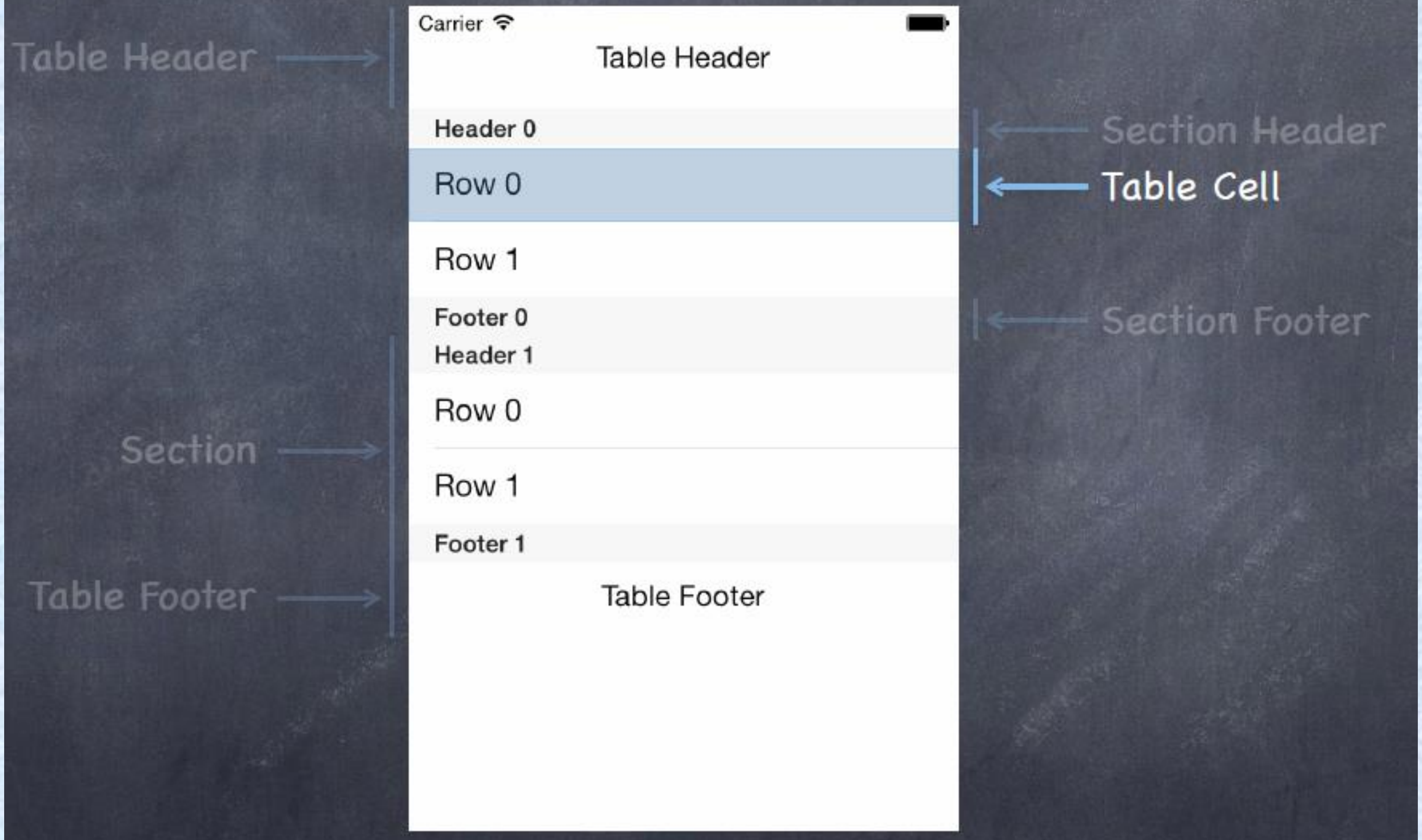
`UITableViewDataSource's tableView:titleForHeaderInSection:`

Plain Style



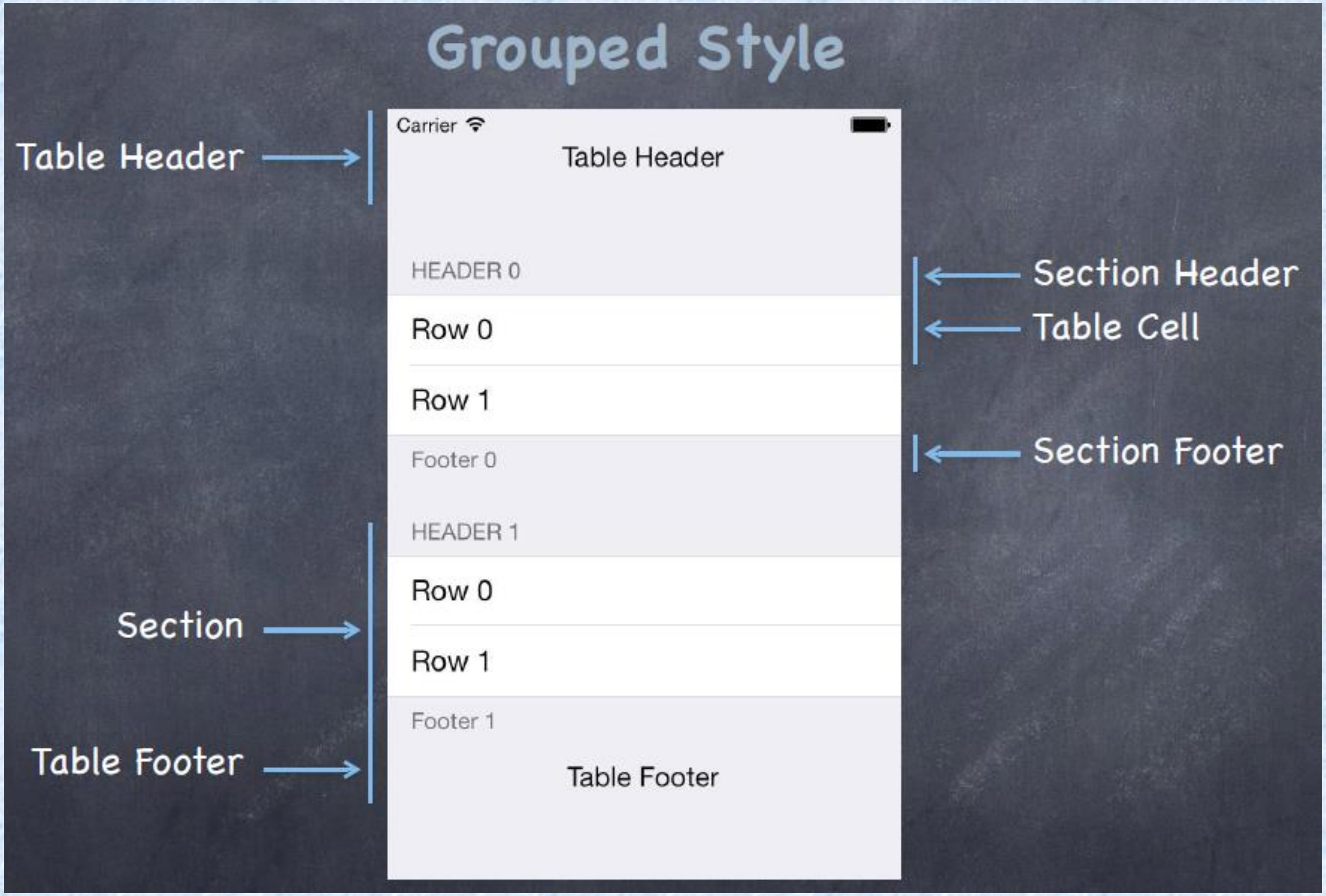
`UITableViewDataSource’s tableView:titleForFooterInSection:`

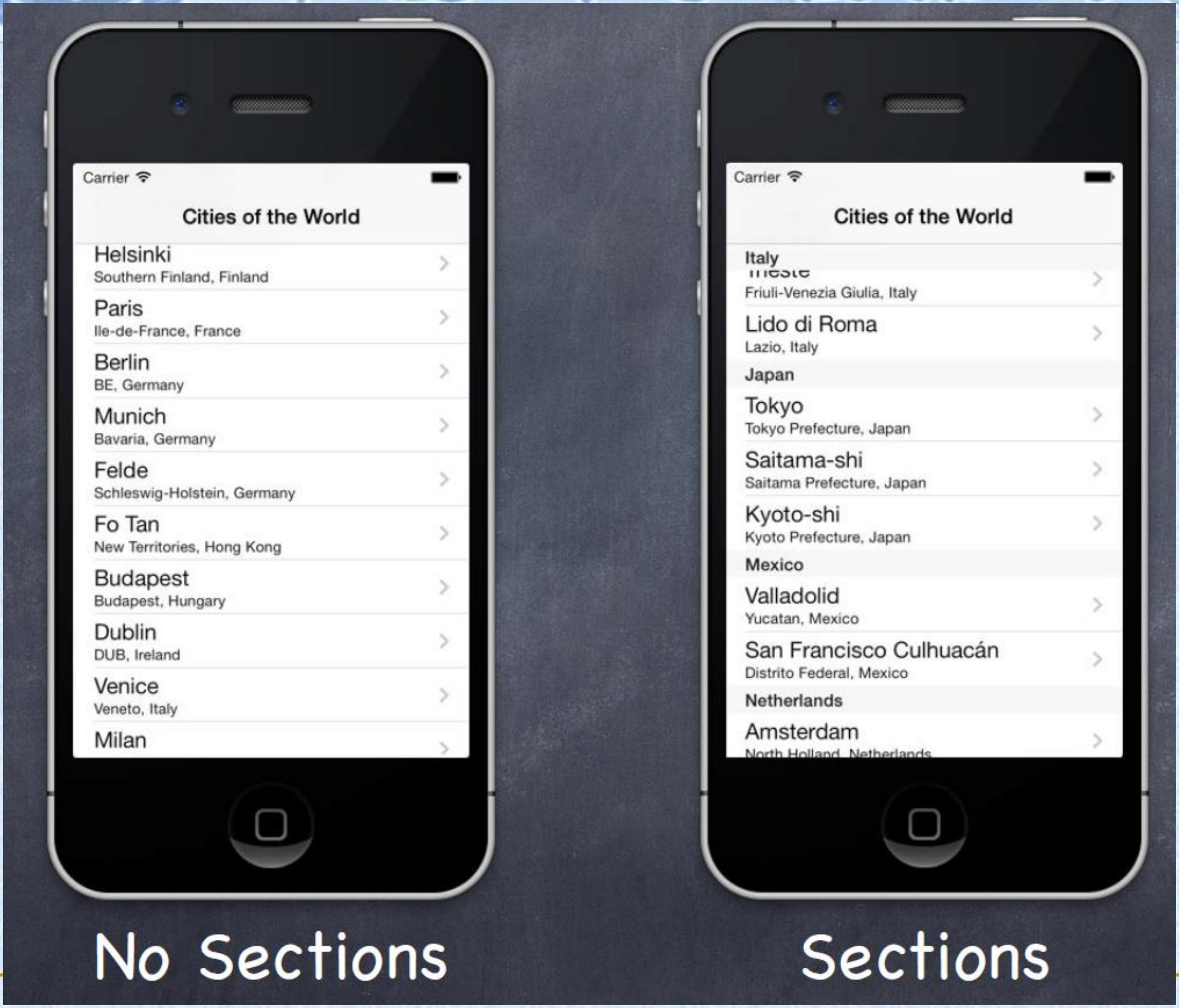
Plain Style



`UITableViewDataSource's tableView:cellForRowAtIndexPath:`

Grouped Style





Carrier

Cities of the World

Helsinki
Southern Finland, Finland

Paris
Ile-de-France, France

Berlin
BE, Germany

Munich
Bavaria, Germany

Felde
Schleswig-Holstein, Germany

Fo Tan
New Territories, Hong Kong

Budapest
Budapest, Hungary

Dublin
DUB, Ireland

Venice
Veneto, Italy

Milan

No Sections

Carrier

Cities of the World

Italy

Friuli-Venezia Giulia, Italy

Lido di Roma
Lazio, Italy

Japan

Tokyo
Tokyo Prefecture, Japan

Saitama-shi
Saitama Prefecture, Japan

Kyoto-shi
Kyoto Prefecture, Japan

Mexico

Valladolid
Yucatan, Mexico

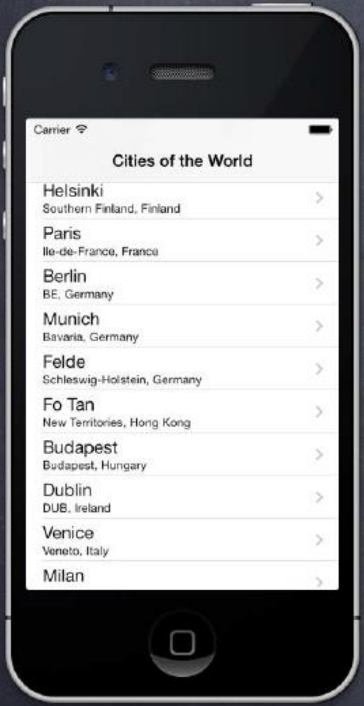
San Francisco Culhuacán
Distrito Federal, Mexico

Netherlands

Amsterdam
North Holland, Netherlands

Sections

Cell Type



Carrier

Cities of the World

Helsinki
Southern Finland, Finland

Paris
Ile-de-France, France

Berlin
BE, Germany

Munich
Bavaria, Germany

Felde
Schleswig-Holstein, Germany

Fo Tan
New Territories, Hong Kong

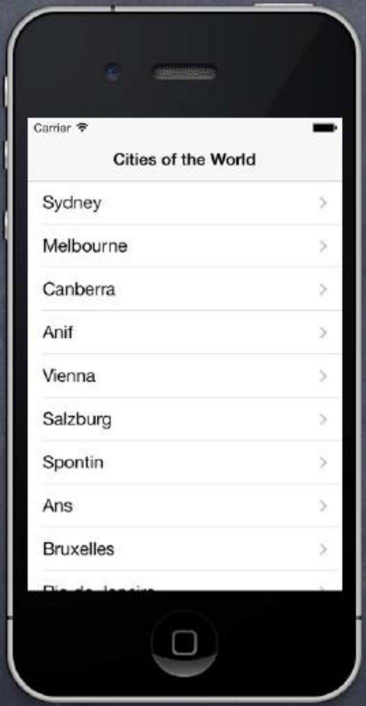
Budapest
Budapest, Hungary

Dublin
DUB, Ireland

Venice
Veneto, Italy

Milan

Subtitle
`UITableViewCellStyleSubtitle`

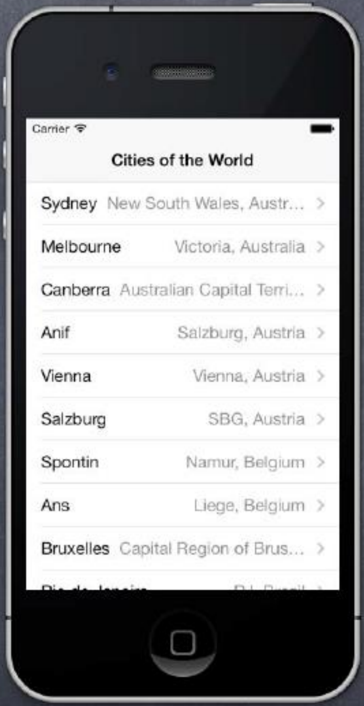


Carrier

Cities of the World

Sydney
Melbourne
Canberra
Anif
Vienna
Salzburg
Spontin
Ans
Bruxelles

Basic
`UITableViewCellStyleDefault`

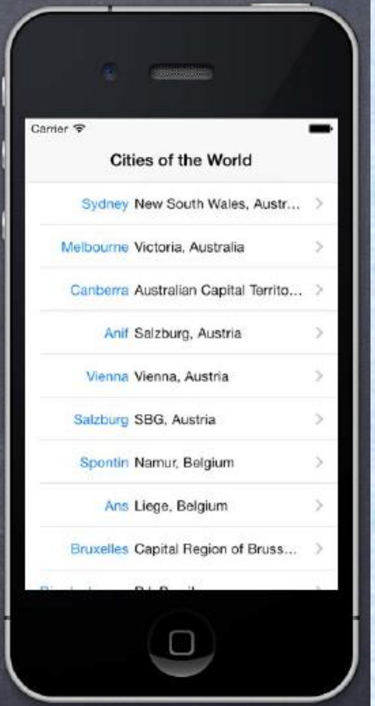


Carrier

Cities of the World

Sydney	New South Wales, Austr...
Melbourne	Victoria, Australia
Canberra	Australian Capital Terri...
Anif	Salzburg, Austria
Vienna	Vienna, Austria
Salzburg	SBG, Austria
Spontin	Namur, Belgium
Ans	Liege, Belgium
Bruxelles	Capital Region of Brus...

Right Detail
`UITableViewCellStyleValue1`



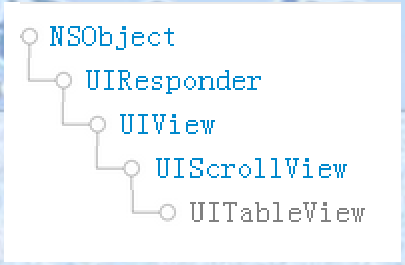
Carrier

Cities of the World

Sydney	New South Wales, Austr...
Melbourne	Victoria, Australia
Canberra	Australian Capital Territo...
Anif	Salzburg, Austria
Vienna	Vienna, Austria
Salzburg	SBG, Austria
Spontin	Namur, Belgium
Ans	Liege, Belgium
Bruxelles	Capital Region of Bruss...

Left Detail
`UITableViewCellStyleValue2`

二、UITableView



An instance of UITableView (or simply, a table view) is a means for displaying and editing hierarchical lists of information.

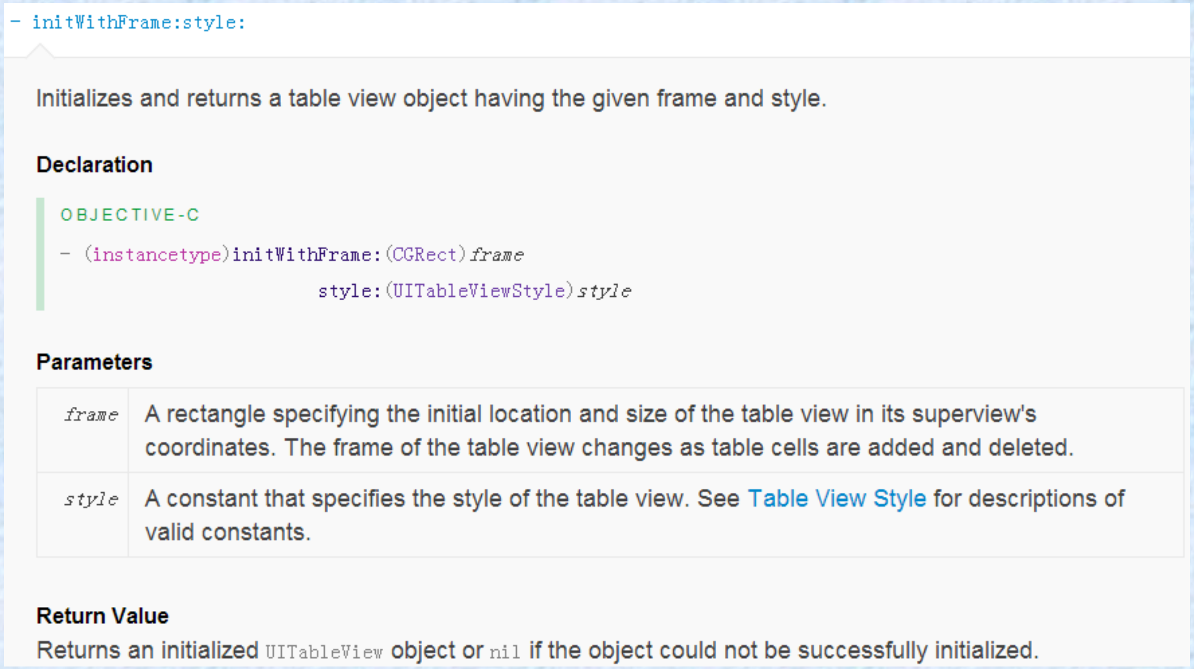
A table view displays a list of items in a single column. UITableView is a subclass of UIScrollView, which allows users to scroll through the table, although UITableView allows vertical scrolling only. The cells comprising the individual items of the table are UITableViewCell objects; UITableView uses these objects to draw the visible rows of the table. Cells have content—titles and images—and can have, near the right edge, accessory views. Standard accessory views are disclosure indicators or detail disclosure buttons; the former leads to the next level in a data hierarchy and the latter leads to a detailed view of a selected item. Accessory views can also be framework controls, such as switches and sliders, or can be custom views. Table views can enter an editing mode where users can insert, delete, and reorder rows of the table.

●创建表视图可以有两种方法:

方法一：在故事板Storyboard中利用IB。

(注意要为IB创建的表视图设置对应的类)

方法二：代码方式



- initWithFrame:style:

Initializes and returns a table view object having the given frame and style.

Declaration

OBJECTIVE-C

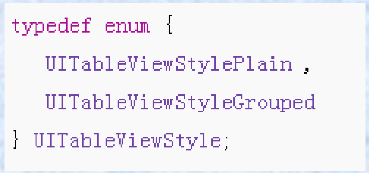
```
- (instancetype)initWithFrame:(CGRect)frame
                        style:(UITableViewStyle)style
```

Parameters

frame	A rectangle specifying the initial location and size of the table view in its superview's coordinates. The frame of the table view changes as table cells are added and deleted.
style	A constant that specifies the style of the table view. See Table View Style for descriptions of valid constants.

Return Value

Returns an initialized `UITableView` object or `nil` if the object could not be successfully initialized.



```
typedef enum {
   UITableViewStylePlain ,
   UITableViewStyleGrouped
} UITableViewStyle;
```

```
UITableView *tv=[[UITableView alloc] initWithFrame:style: CGRectMake(10,10,300,300)
                                    style: UITableViewStylePlain];
[self.view addSubview:tv];
```

在故事板中利用IB设置表视图

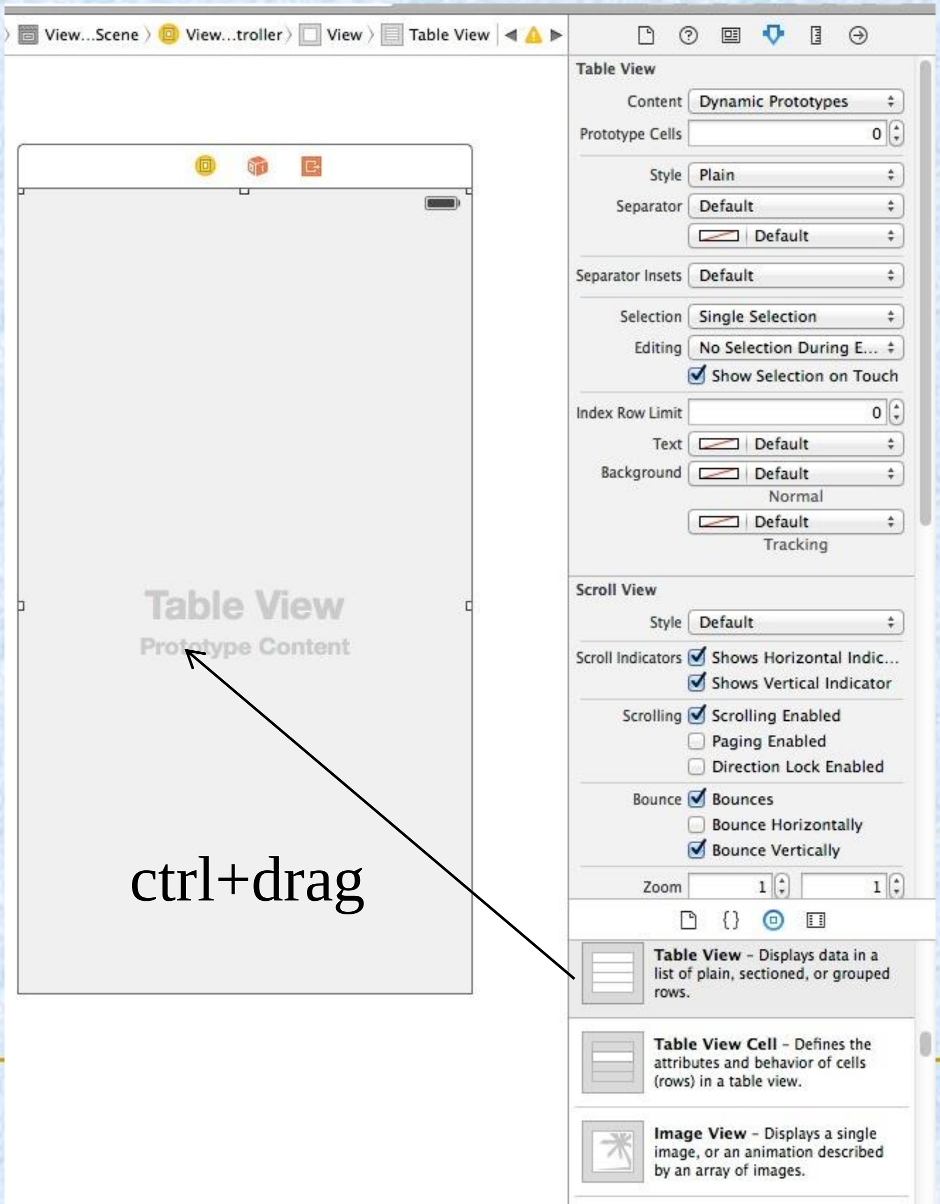


Table View

Content: Dynamic Prototypes
Prototype Cells: 0
Style: Plain
Separator: Default
Default
Separator Insets: Default
Selection: Single Selection
Editing: No Selection During E...
☑ Show Selection on Touch
Index Row Limit: 0
Text: Default
Background: Default
Normal
Default
Tracking

Scroll View

Style: Default
Scroll Indicators: ☑ Shows Horizontal Indic...
☑ Shows Vertical Indicator
Scrolling: ☑ Scrolling Enabled
☐ Paging Enabled
☐ Direction Lock Enabled
Bounce: ☑ Bounces
☐ Bounce Horizontally
☑ Bounce Vertically
Zoom: 1 1

Table View – Displays data in a list of plain, sectioned, or grouped rows.

Table View Cell – Defines the attributes and behavior of cells (rows) in a table view.

Image View – Displays a single image, or an animation described by an array of images.

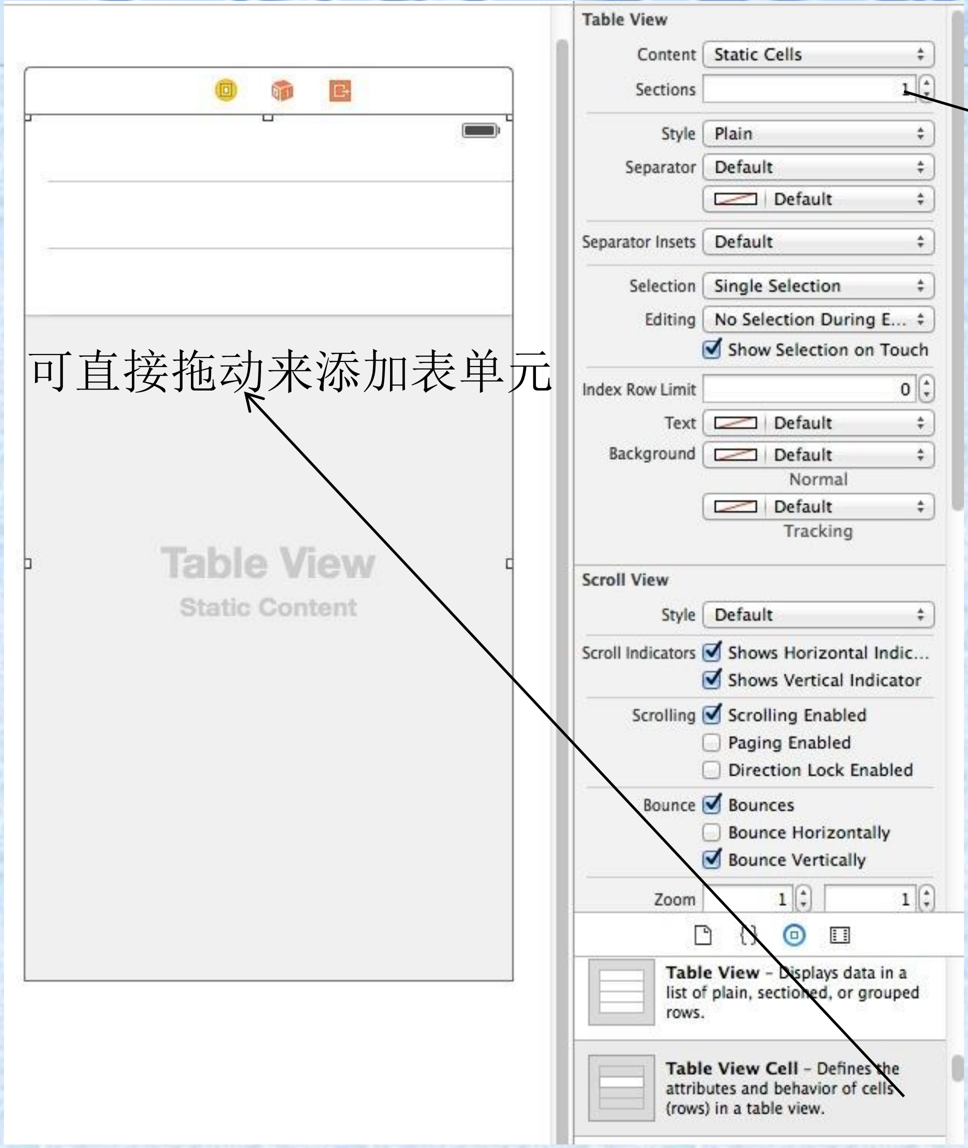


Table View

Content	Static Cells
Sections	1
Style	Plain
Separator	Default
	Default
Separator Insets	Default
Selection	Single Selection
Editing	No Selection During E...
	✓ Show Selection on Touch
Index Row Limit	0
Text	Default
Background	Default
	Normal
	Default
	Tracking

Scroll View

Style	Default
Scroll Indicators	✓ Shows Horizontal Indic...
	✓ Shows Vertical Indicator
Scrolling	✓ Scrolling Enabled
	Paging Enabled
	Direction Lock Enabled
Bounce	✓ Bounces
	Bounce Horizontally
	✓ Bounce Vertically
Zoom	1 1

Table View – Displays data in a list of plain, sectioned, or grouped rows.

Table View Cell – Defines the attributes and behavior of cells (rows) in a table view.

设置section数量

可直接拖动来添加表单元

静态表视图

表单元的原型

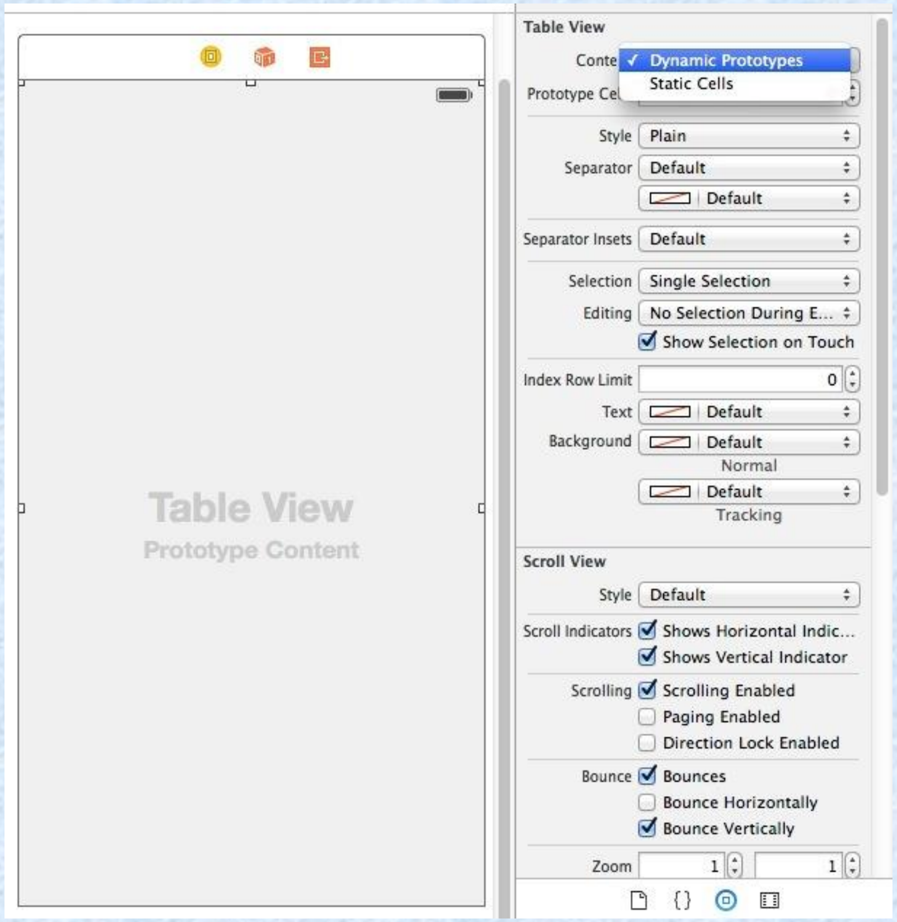


Table View
Content: Dynamic Prototypes
Prototype Cells
Style: Plain
Separator: Default
Default
Separator Insets: Default
Selection: Single Selection
Editing: No Selection During E...
☑ Show Selection on Touch
Index Row Limit: 0
Text: Default
Background: Default
Normal
Default
Tracking

Scroll View
Style: Default
Scroll Indicators: ☑ Shows Horizontal Indic...
☑ Shows Vertical Indicator
Scrolling: ☑ Scrolling Enabled
☐ Paging Enabled
☐ Direction Lock Enabled
Bounce: ☑ Bounces
☐ Bounce Horizontally
☑ Bounce Vertically
Zoom: 1 1

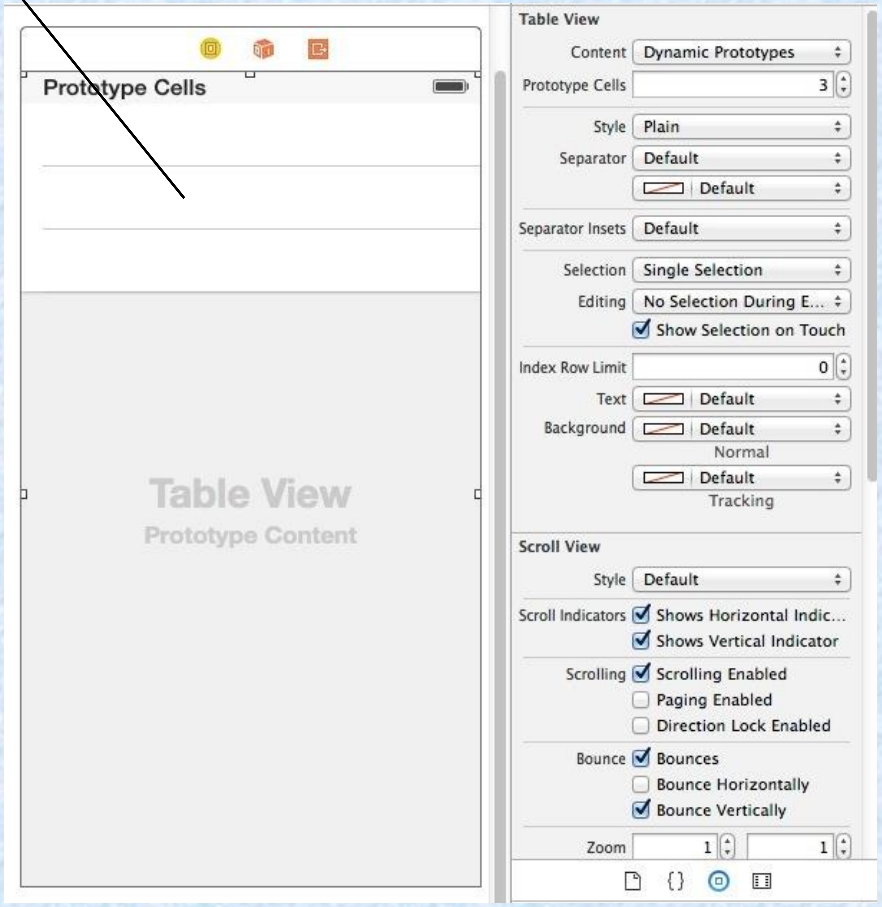
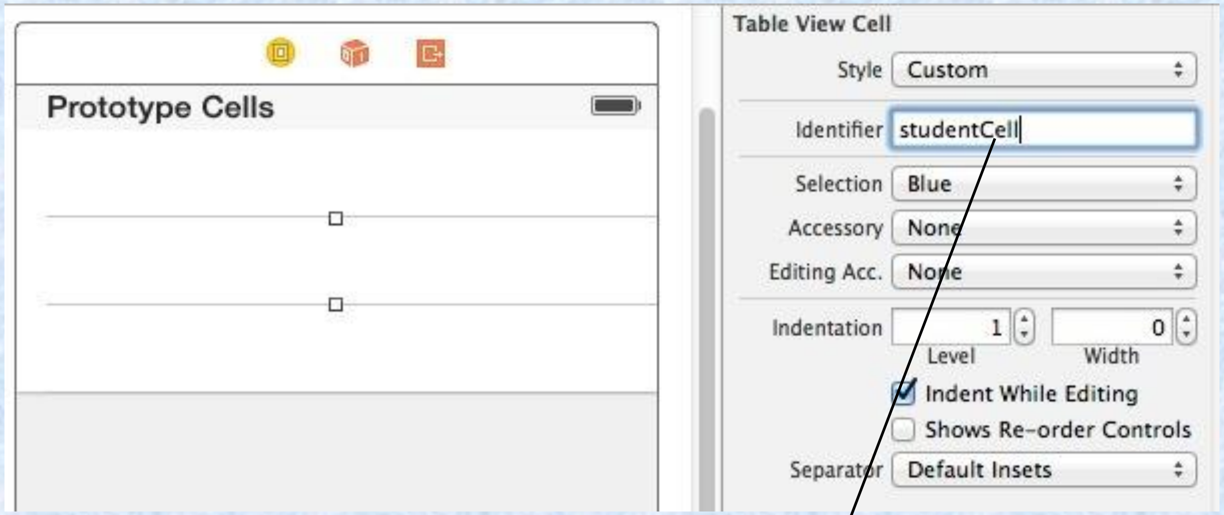


Table View
Content: Dynamic Prototypes
Prototype Cells: 3
Style: Plain
Separator: Default
Default
Separator Insets: Default
Selection: Single Selection
Editing: No Selection During E...
☑ Show Selection on Touch
Index Row Limit: 0
Text: Default
Background: Default
Normal
Default
Tracking

Scroll View
Style: Default
Scroll Indicators: ☑ Shows Horizontal Indic...
☑ Shows Vertical Indicator
Scrolling: ☑ Scrolling Enabled
☐ Paging Enabled
☐ Direction Lock Enabled
Bounce: ☑ Bounces
☐ Bounce Horizontally
☑ Bounce Vertically
Zoom: 1 1

由代码动态生成表单元



Style: Custom
Identifier: studentCell
Selection: Blue
Accessory: None
Editing Acc.: None
Indentation: 1 Level, 0 Width
[✓] Indent While Editing
[ ] Shows Re-order Controls
Separator: Default Insets

用来标识表单元

配置表视图

```
  style
- numberOfRowsInSection:
- numberOfSections
  rowHeight
  separatorStyle
  separatorColor
  separatorEffect
  backgroundView
  separatorInset
```

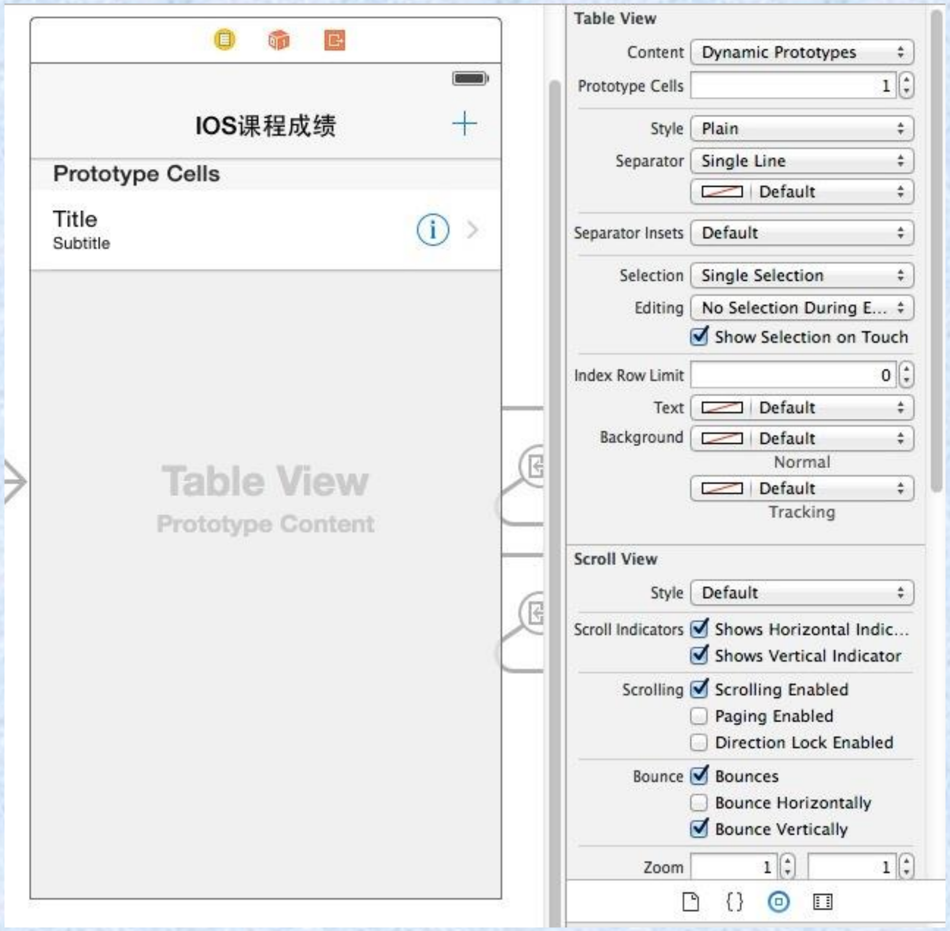

IOS课程成绩
Prototype Cells
Title
Subtitle
Table View
Prototype Content

Table View
Content: Dynamic Prototypes
Prototype Cells: 1
Style: Plain
Separator: Single Line
Default
Separator Insets: Default
Selection: Single Selection
Editing: No Selection During E...
✓ Show Selection on Touch
Index Row Limit: 0
Text: Default
Background: Default
Normal
Default
Tracking

Scroll View
Style: Default
Scroll Indicators: ✓ Shows Horizontal Indic...
✓ Shows Vertical Indicator
Scrolling: ✓ Scrolling Enabled
☐ Paging Enabled
☐ Direction Lock Enabled
Bounce: ✓ Bounces
☐ Bounce Horizontally
✓ Bounce Vertically
Zoom: 1 1

代码方式

在StoryBoard中利用IB设置

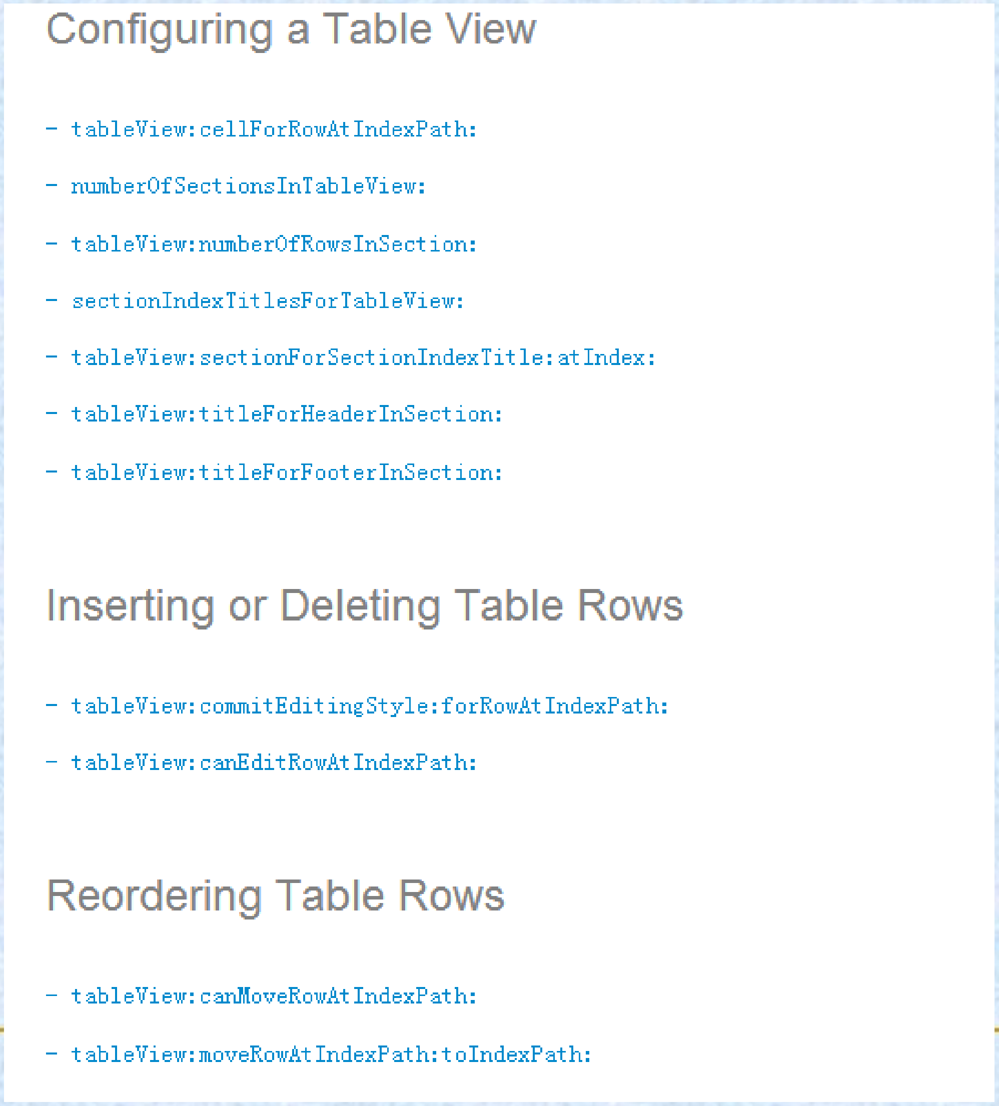
三、表视图与UITableViewDataSource协议

The UITableViewDataSource protocol is adopted by an object that mediates the application’s™ data model for aUITableView object. The data source provides the table-view object with the information it needs to construct and modify a table view.

As a representative of the data model, the data source supplies minimal information about the table view’s appearance. The table-view object’s delegate—an object adopting the UITableViewDelegate protocol—provides that information.

The required methods of the protocol provide the cells to be displayed by the table-view as well as inform the UITableView object about the number of sections and the number of rows in each section. The data source may implement optional methods to configure various aspects of the table view and to insert, delete, and reorder rows.

表视图与UITableViewDataSource协议



Configuring a Table View

- tableView:cellForRowAtIndexPath:
- numberOfSectionsInTableView:
- tableView:numberOfRowsInSection:
- sectionIndexTitlesForTableView:
- tableView:sectionForSectionIndexTitle:atIndex:
- tableView:titleForHeaderInSection:
- tableView:titleForFooterInSection:

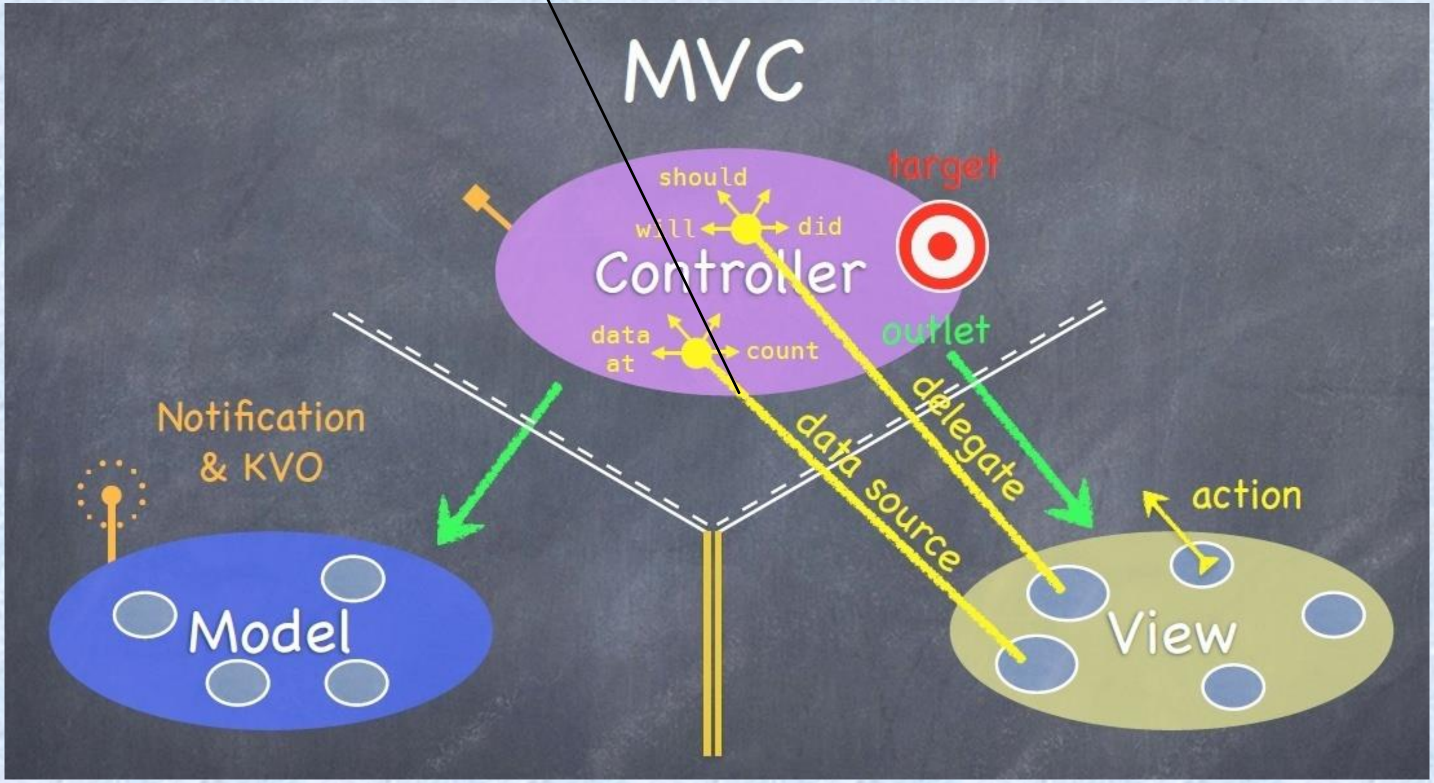
Inserting or Deleting Table Rows

- tableView:commitEditingStyle:forRowAtIndexPath:
- tableView:canEditRowAtIndexPath:

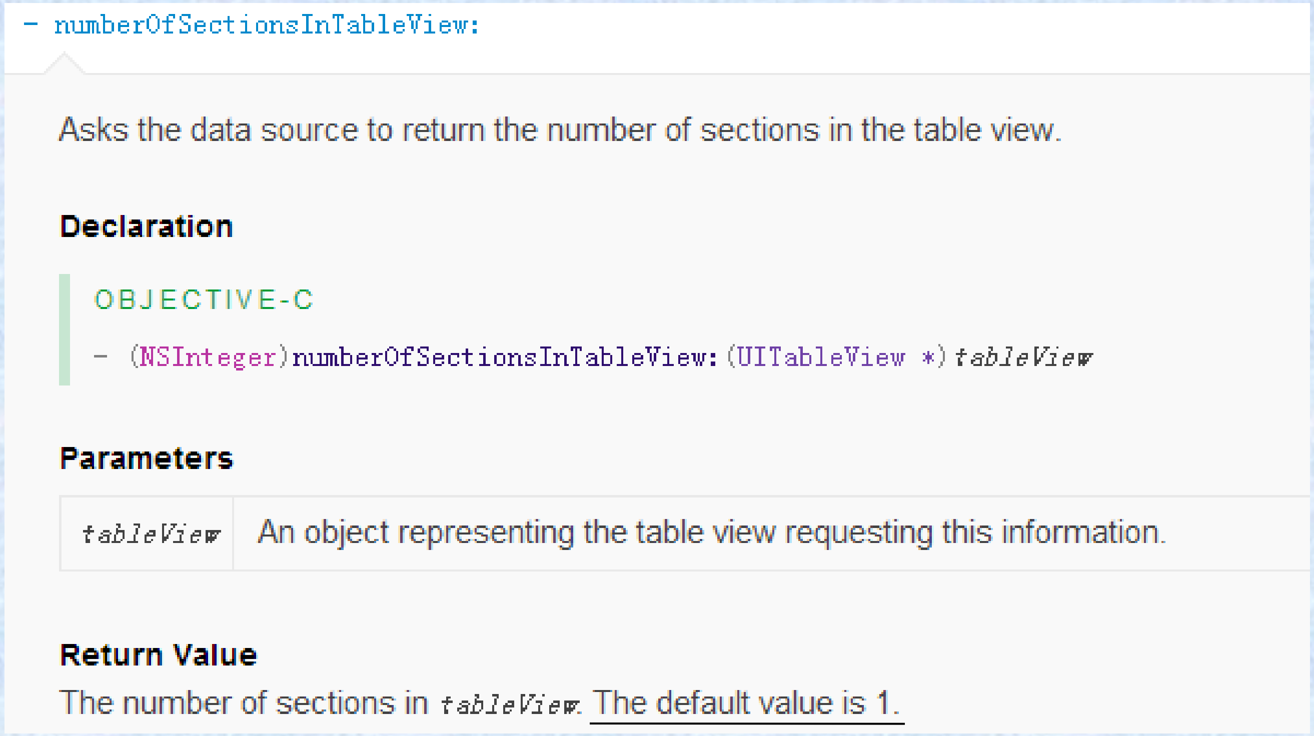
Reordering Table Rows

- tableView:canMoveRowAtIndexPath:
- tableView:moveRowAtIndexPath:toIndexPath:

表视图与UITableViewDataSource协议



设置表视图中小节(section)的数量


- numberOfSectionsInTableView:

Asks the data source to return the number of sections in the table view.

Declaration

OBJECTIVE-C

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
```

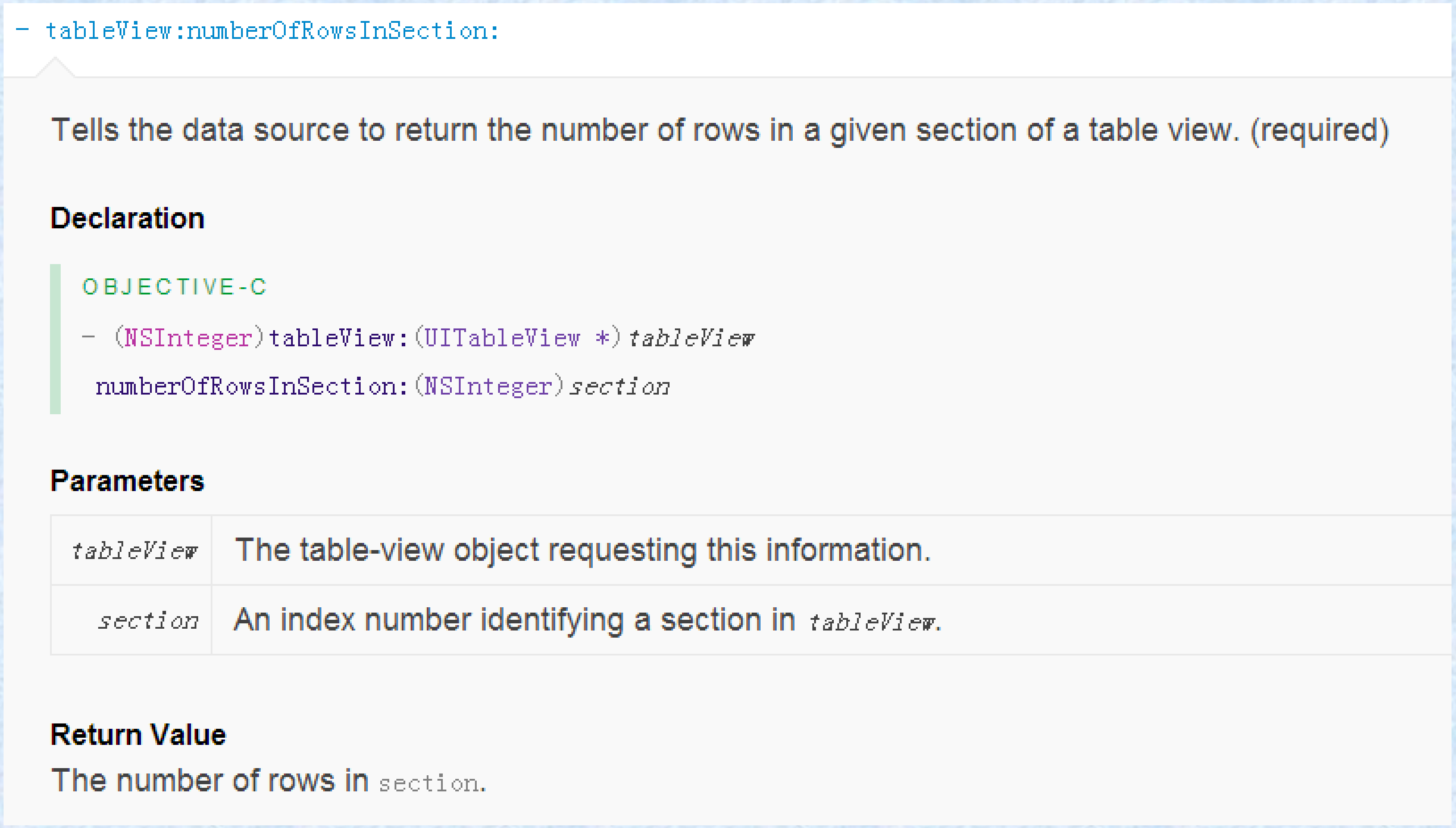
Parameters

tableView	An object representing the table view requesting this information.

Return Value

The number of sections in *tableView*. The default value is 1.

设置表视图中小节(section)中表单元的数量



– tableView:numberOfRowsInSection:

Tells the data source to return the number of rows in a given section of a table view. (required)

Declaration

OBJECTIVE-C

```
- (NSInteger)tableView:(UITableView *)tableView
 numberOfRowsInSection:(NSInteger)section
```

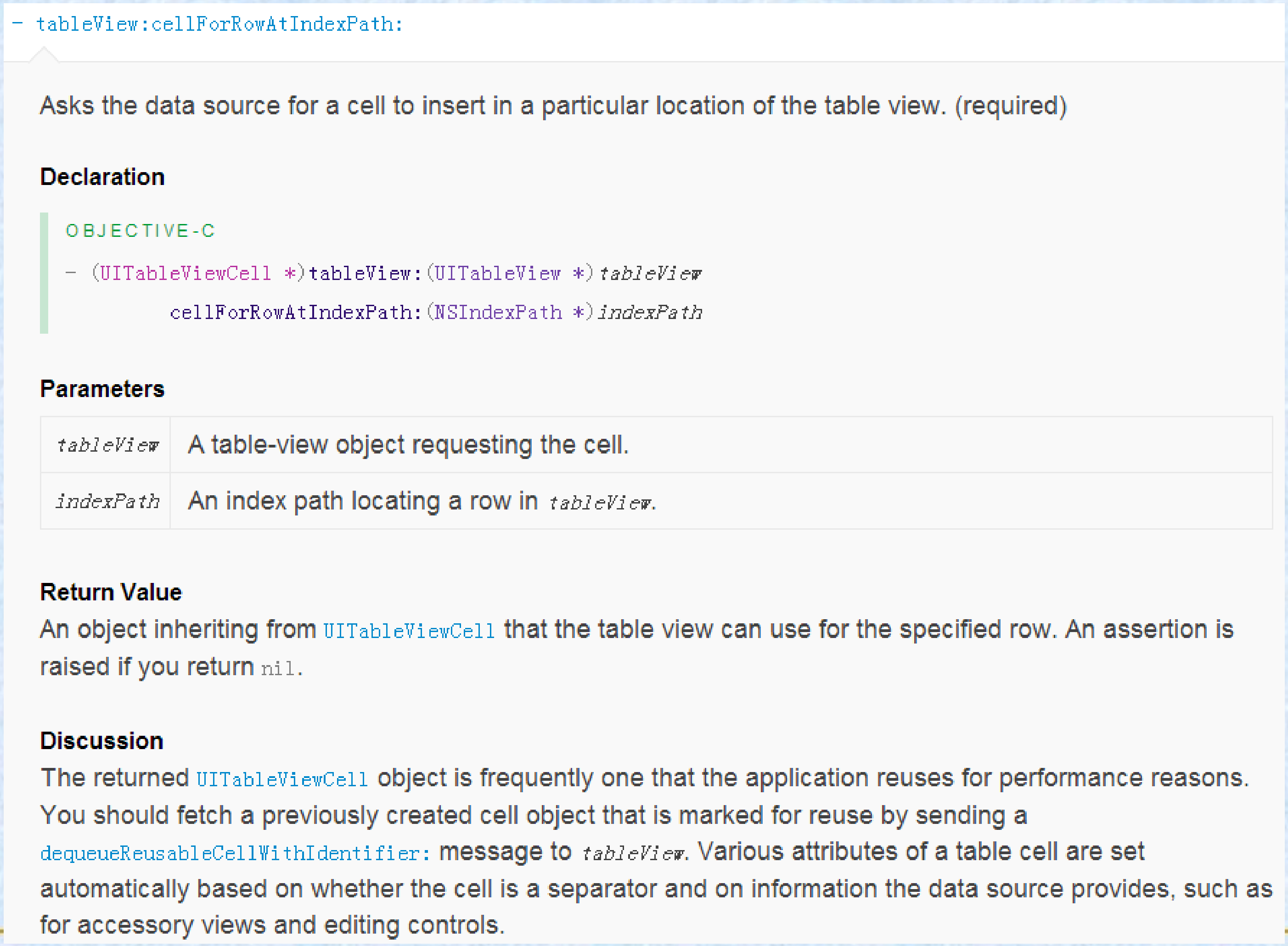
Parameters

tableView	The table-view object requesting this information.
section	An index number identifying a section in *tableView*.

Return Value

The number of rows in `section`.

插入表单元



- tableView:cellForRowAtIndexPath:

Asks the data source for a cell to insert in a particular location of the table view. (required)

Declaration

OBJECTIVE-C

```
- (UITableViewCell *)tableView:(UITableView *)tableView
          cellForRowAtIndexPath:(NSIndexPath *)indexPath
```

Parameters

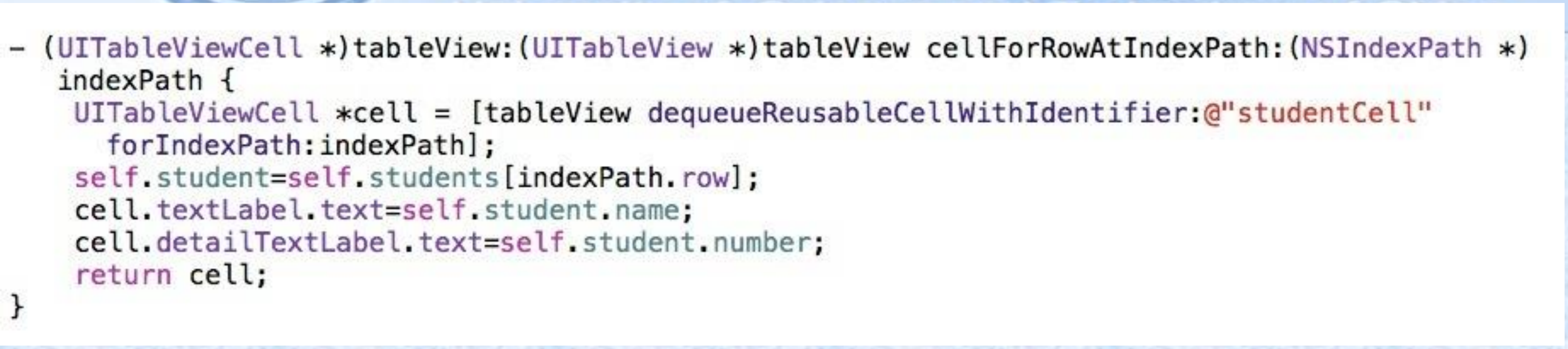
tableView	A table-view object requesting the cell.
indexPath	An index path locating a row in *tableView*.

Return Value

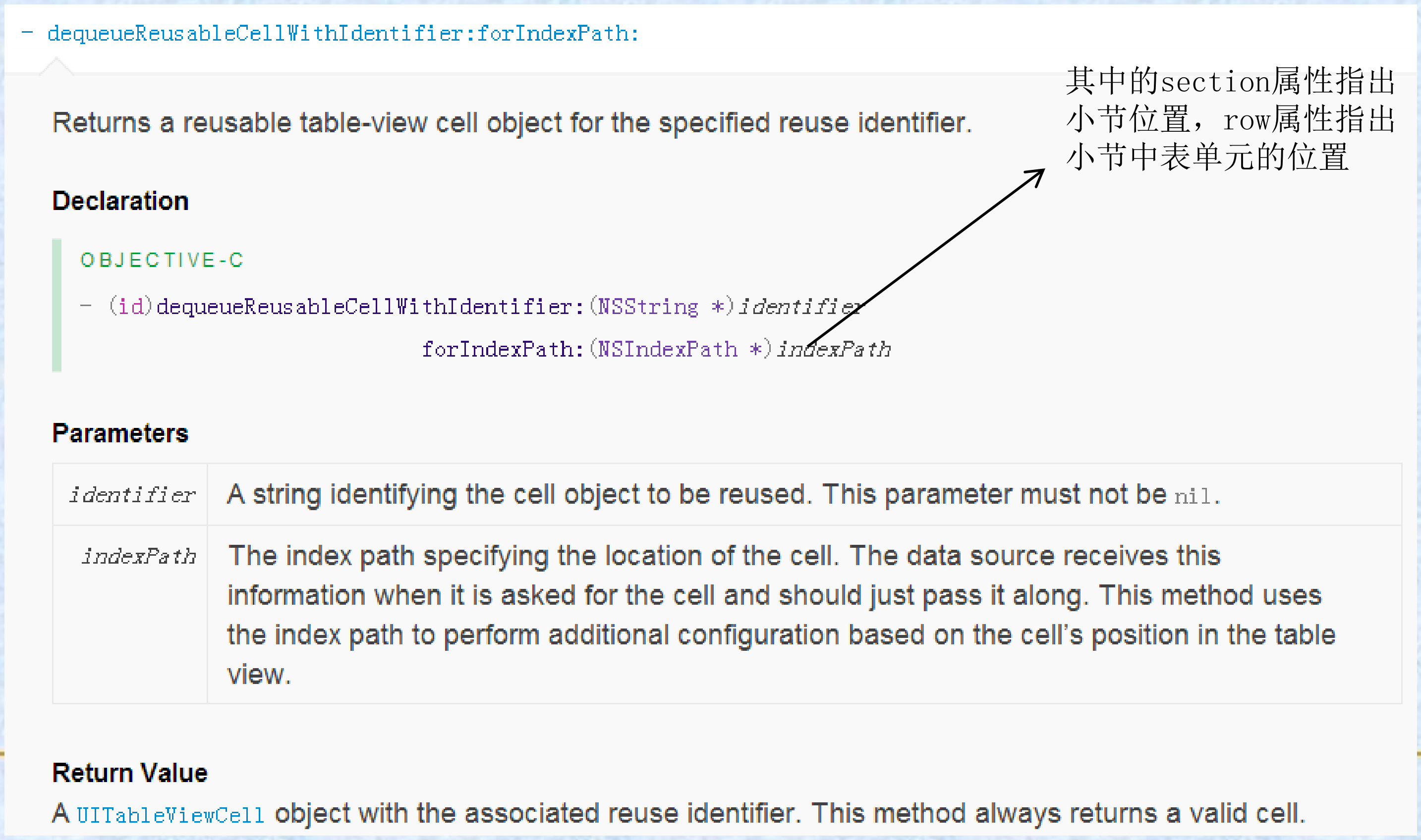
An object inheriting from `UITableViewCell` that the table view can use for the specified row. An assertion is raised if you return `nil`.

Discussion

The returned `UITableViewCell` object is frequently one that the application reuses for performance reasons. You should fetch a previously created cell object that is marked for reuse by sending a `dequeueReusableCellWithIdentifier:` message to *tableView*. Various attributes of a table cell are set automatically based on whether the cell is a separator and on information the data source provides, such as for accessory views and editing controls.



```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)
   indexPath {
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"studentCell"
      forIndexPath:indexPath];
    self.student=self.students[indexPath.row];
    cell.textLabel.text=self.student.name;
    cell.detailTextLabel.text=self.student.number;
    return cell;
}
```



`- dequeueReusableCellWithIdentifier:forIndexPath:`

Returns a reusable table-view cell object for the specified reuse identifier.

Declaration

OBJECTIVE-C

```
- (id)dequeueReusableCellWithIdentifier:(NSString *)identifier
                           forIndexPath:(NSIndexPath *)indexPath
```

其中的section属性指出小节位置，row属性指出小节中表单元的位置

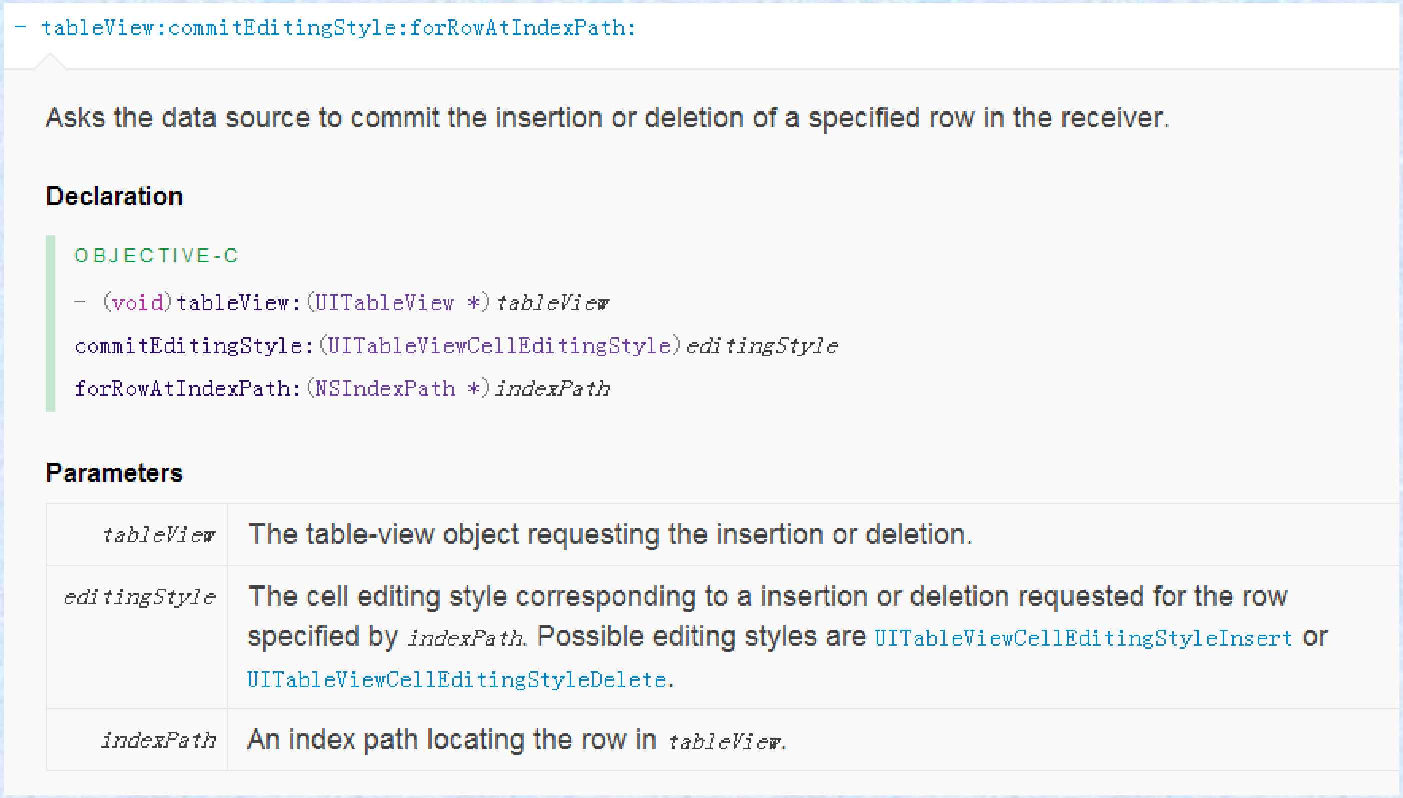
Parameters

identifier	A string identifying the cell object to be reused. This parameter must not be `nil`.
indexPath	The index path specifying the location of the cell. The data source receives this information when it is asked for the cell and should just pass it along. This method uses the index path to perform additional configuration based on the cell's position in the table view.

Return Value

A `UITableViewCell` object with the associated reuse identifier. This method always returns a valid cell.

删除表视图中的某个表单元



- tableView:commitEditingStyle:forRowAtIndexPath:

Asks the data source to commit the insertion or deletion of a specified row in the receiver.

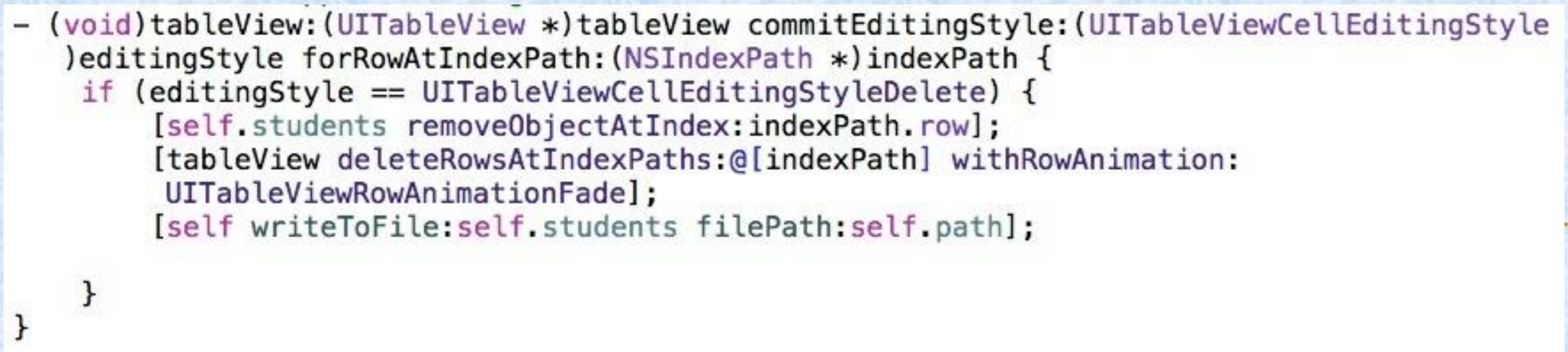
Declaration

OBJECTIVE-C

```
- (void)tableView:(UITableView *)tableView
commitEditingStyle:(UITableViewCellEditingStyle)editingStyle
forRowAtIndexPath:(NSIndexPath *)indexPath
```

Parameters

tableView	The table-view object requesting the insertion or deletion.
editingStyle	The cell editing style corresponding to a insertion or deletion requested for the row specified by *indexPath*. Possible editing styles are `UITableViewCellEditingStyleInsert` or `UITableViewCellEditingStyleDelete`.
indexPath	An index path locating the row in *tableView*.

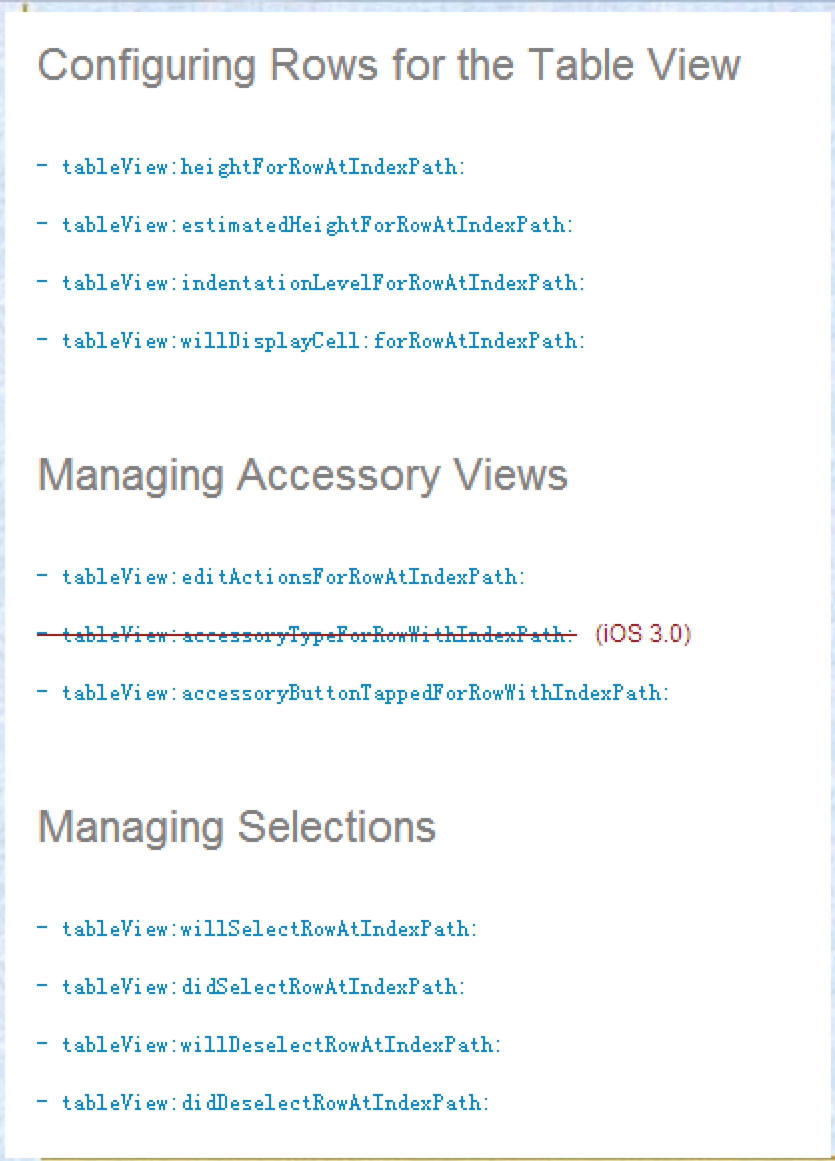


```
- (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle
   )editingStyle forRowAtIndexPath:(NSIndexPath *)indexPath {
    if (editingStyle == UITableViewCellEditingStyleDelete) {
        [self.students removeObjectAtIndex:indexPath.row];
        [tableView deleteRowsAtIndexPaths:@[indexPath] withRowAnimation:
         UITableViewRowAnimationFade];
        [self writeToFile:self.students filePath:self.path];

    }
}
```

四、表视图与UITableViewDelegate协议

The delegate of a UITableView object must adopt the UITableView Delegate protocol. Optional methods of the protocol allow the delegate to **manage selections, configure section headings and footers, help to delete and reorder cells, and perform other actions.**

Many methods of the UITableViewDelegate protocol take NSIndexPath objects as parameters and return values. UIKit declares a category on NSIndexPath that enables you to get the represented row index (row property) and section index (section property), and to construct an index path from a given row index and section index (indexPathForRow:inSection:method). Because rows are located within their sections, you usually must evaluate the section index number before you can identify the row by its index number.



Configuring Rows for the Table View

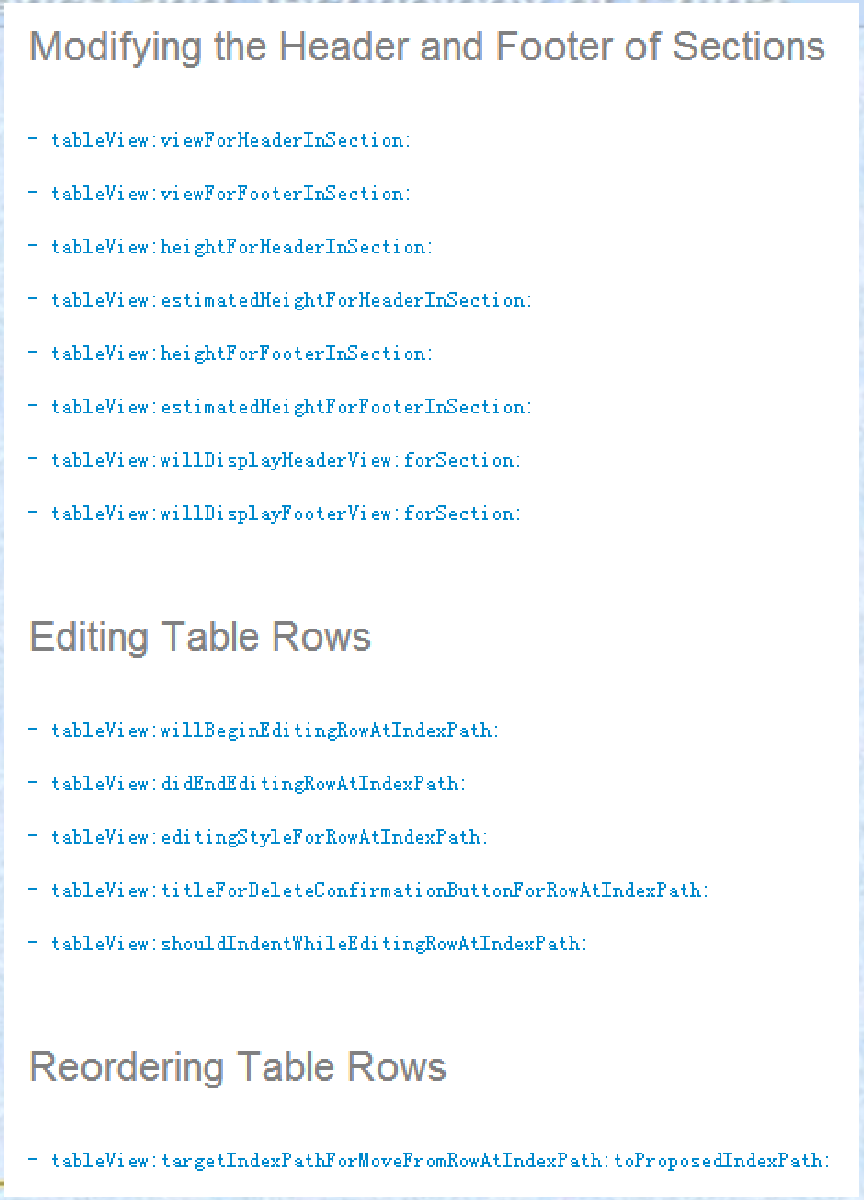
- tableView:heightForRowAtIndexPath:
- tableView:estimatedHeightForRowAtIndexPath:
- tableView:indentationLevelForRowAtIndexPath:
- tableView:willDisplayCell:forRowAtIndexPath:

Managing Accessory Views

- tableView:editActionsForRowAtIndexPath:
- ~~tableView:accessoryTypeForRowWithIndexPath:~~ (iOS 3.0)
- tableView:accessoryButtonTappedForRowWithIndexPath:

Managing Selections

- tableView:willSelectRowAtIndexPath:
- tableView:didSelectRowAtIndexPath:
- tableView:willDeselectRowAtIndexPath:
- tableView:didDeselectRowAtIndexPath:



Modifying the Header and Footer of Sections

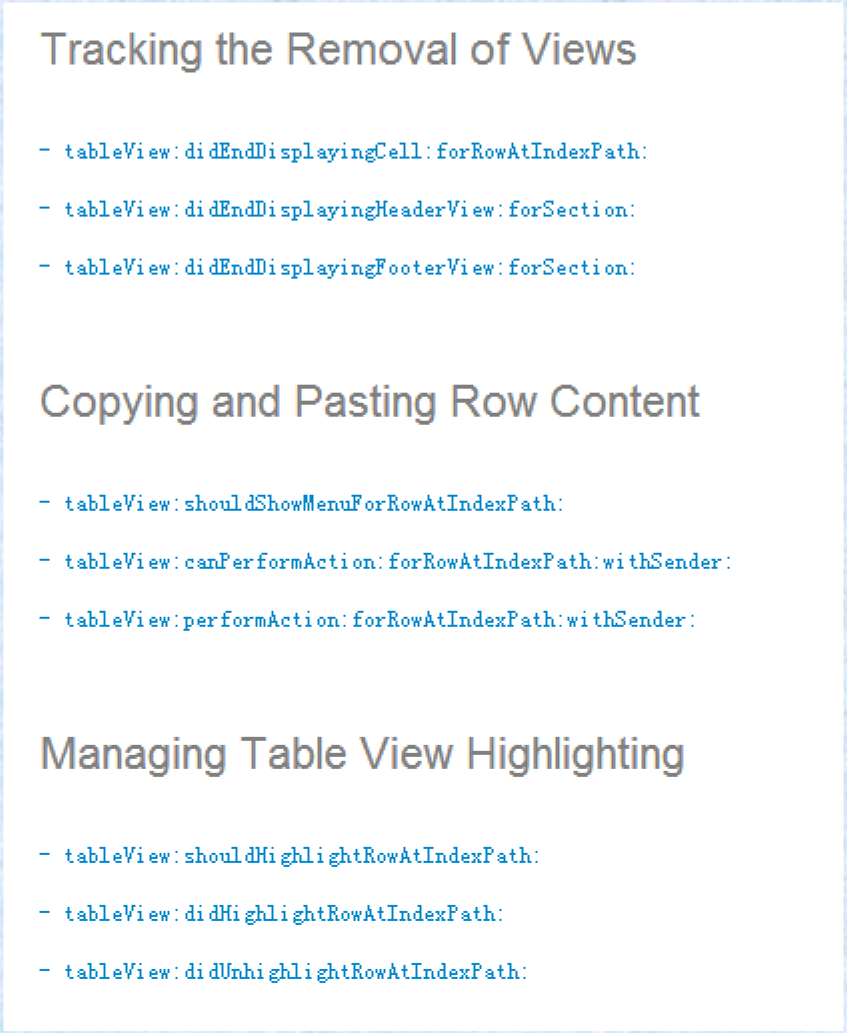
- tableView:viewForHeaderInSection:
- tableView:viewForFooterInSection:
- tableView:heightForHeaderInSection:
- tableView:estimatedHeightForHeaderInSection:
- tableView:heightForFooterInSection:
- tableView:estimatedHeightForFooterInSection:
- tableView:willDisplayHeaderView:forSection:
- tableView:willDisplayFooterView:forSection:

Editing Table Rows

- tableView:willBeginEditingRowAtIndexPath:
- tableView:didEndEditingRowAtIndexPath:
- tableView:editingStyleForRowAtIndexPath:
- tableView:titleForDeleteConfirmationButtonForRowAtIndexPath:
- tableView:shouldIndentWhileEditingRowAtIndexPath:

Reordering Table Rows

- tableView:targetIndexPathForMoveFromRowAtIndexPath:toProposedIndexPath:



Tracking the Removal of Views

- tableView:didEndDisplayingCell:forRowAtIndexPath:
- tableView:didEndDisplayingHeaderView:forSection:
- tableView:didEndDisplayingFooterView:forSection:

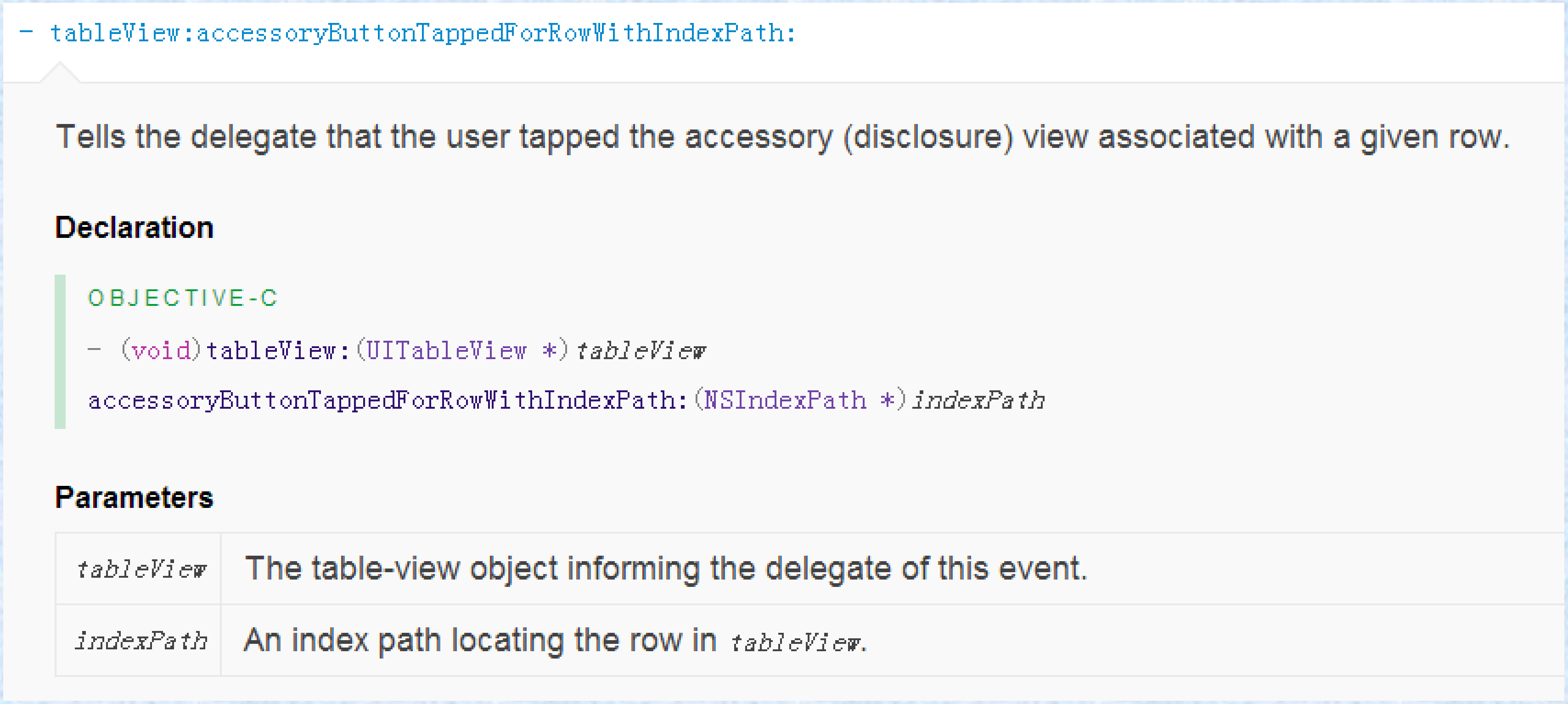
Copying and Pasting Row Content

- tableView:shouldShowMenuForRowAtIndexPath:
- tableView:canPerformAction:forRowAtIndexPath:withSender:
- tableView:performAction:forRowAtIndexPath:withSender:

Managing Table View Highlighting

- tableView:shouldHighlightRowAtIndexPath:
- tableView:didHighlightRowAtIndexPath:
- tableView:didUnhighlightRowAtIndexPath:

当点击表单元上的附件图标时触发



- tableView:accessoryButtonTappedForRowWithIndexPath:

Tells the delegate that the user tapped the accessory (disclosure) view associated with a given row.

Declaration

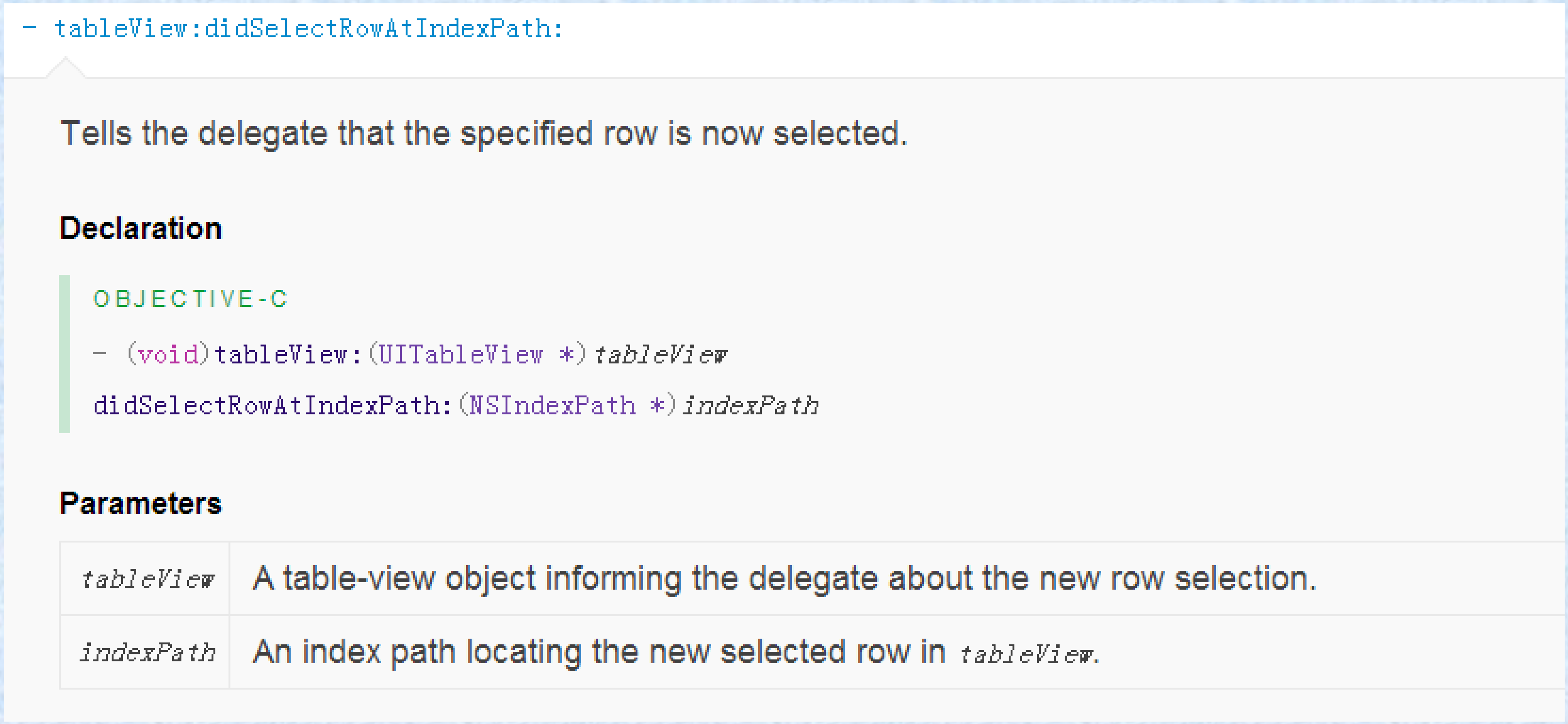
OBJECTIVE-C

```
- (void)tableView:(UITableView *)tableView
accessoryButtonTappedForRowWithIndexPath:(NSIndexPath *)indexPath
```

Parameters

tableView	The table-view object informing the delegate of this event.
indexPath	An index path locating the row in *tableView*.

当点击表单元上时触发



- tableView:didSelectRowAtIndexPath:

Tells the delegate that the specified row is now selected.

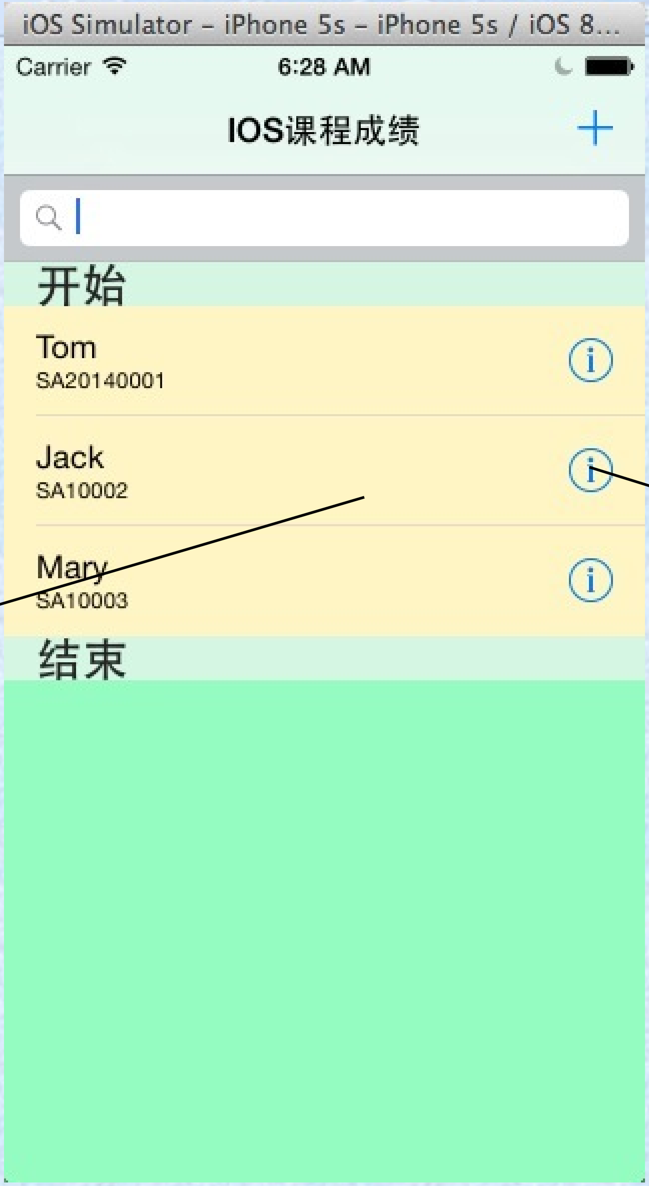
Declaration

OBJECTIVE-C

```
- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
```

Parameters

tableView	A table-view object informing the delegate about the new row selection.
indexPath	An index path locating the new selected row in *tableView*.

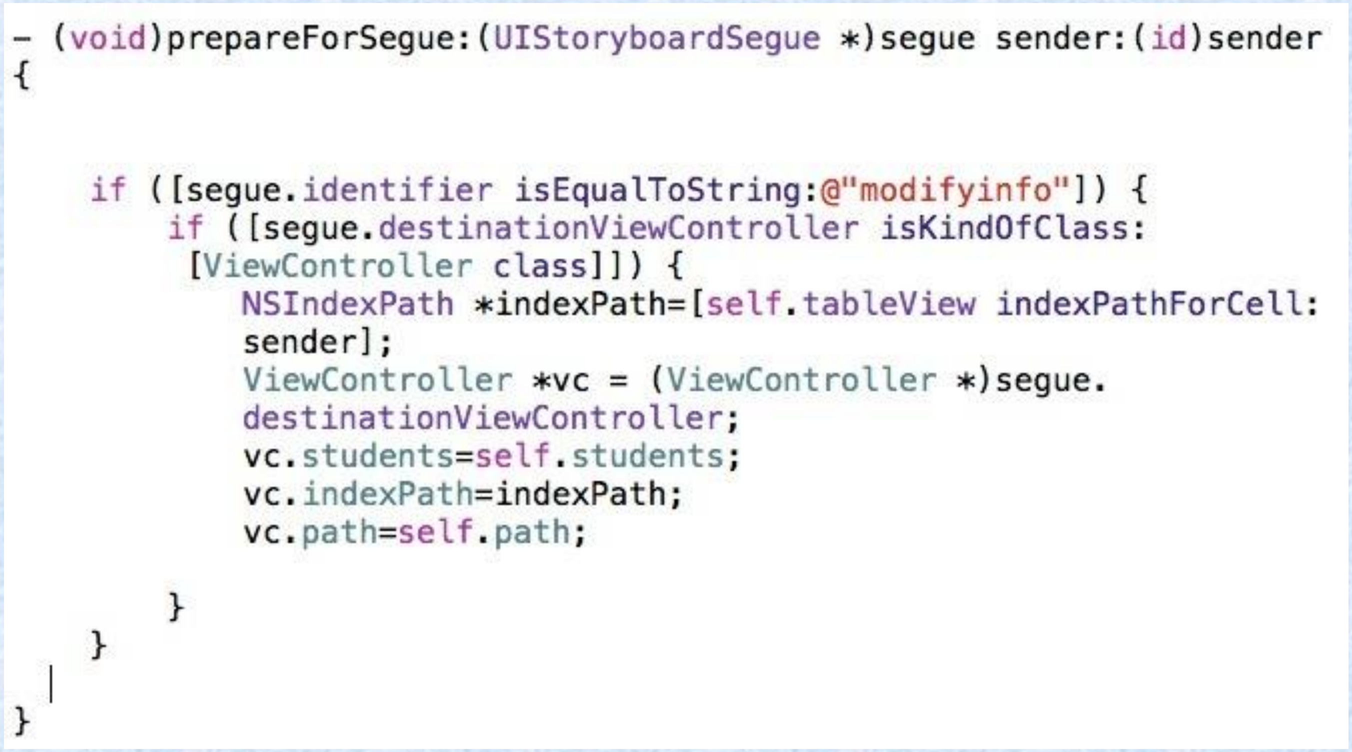


触发didSelectRowAtIndexPath

触发accessoryButtonTappedForRowWithIndexPath:

五、表视图中场景过渡

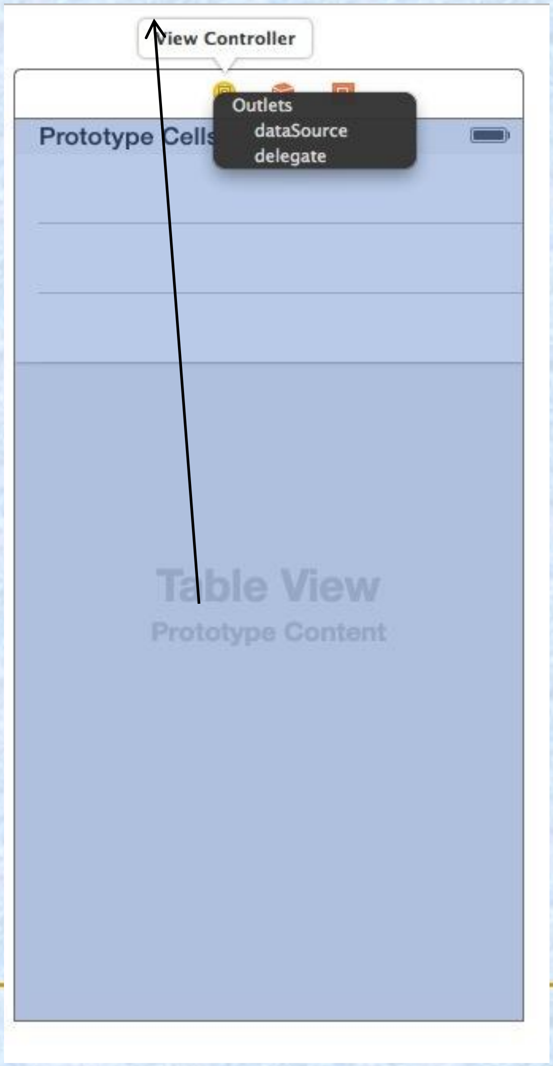
同故事板中多场景过渡一样，也可以通过选取表单元来触发场景过渡。



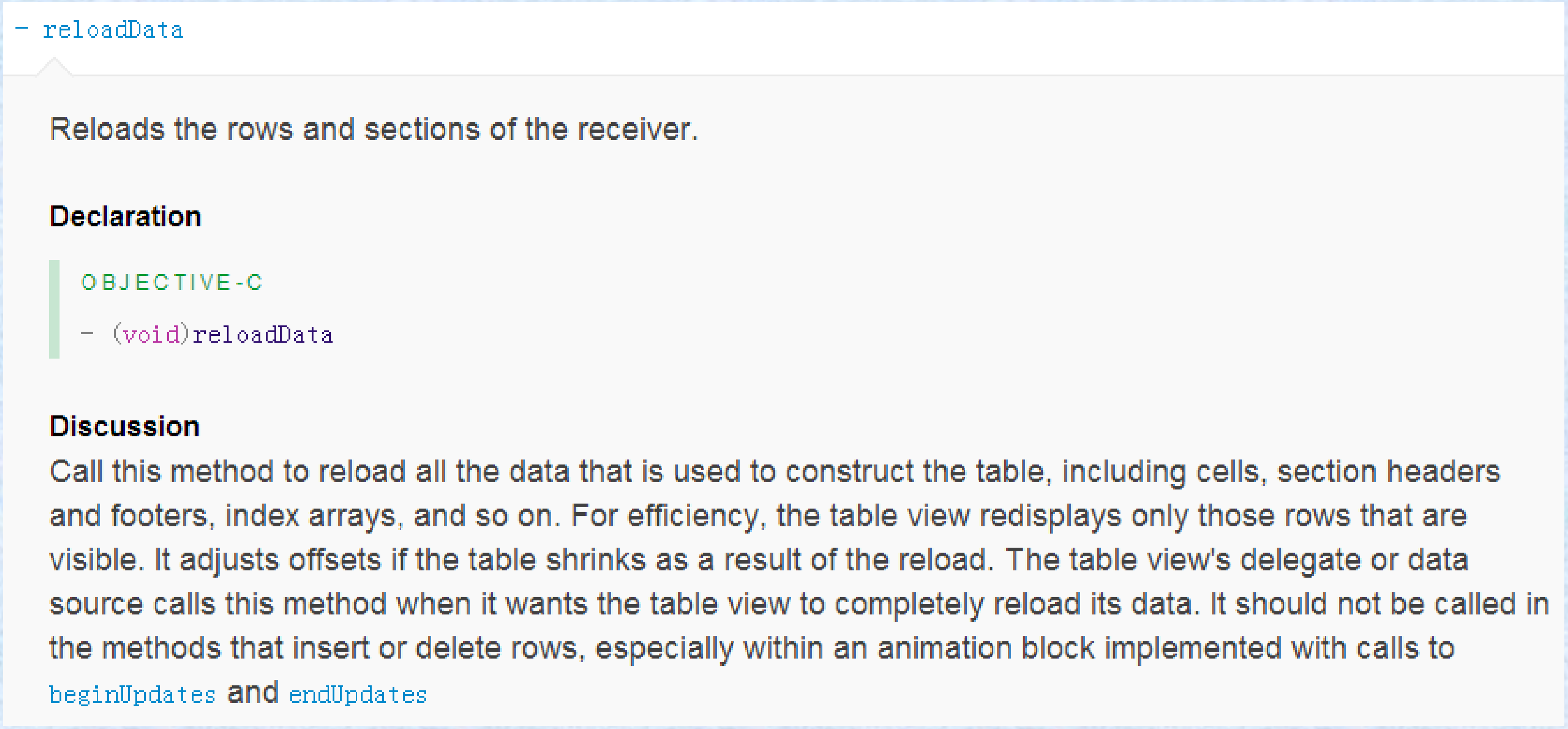
```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{


    if ([segue.identifier isEqualToString:@"modifyinfo"]) {
        if ([segue.destinationViewController isKindOfClass:
         [ViewController class]]) {
            NSIndexPath *indexPath=[self.tableView indexPathForCell:
            sender];
            ViewController *vc = (ViewController *)segue.
            destinationViewController;
            vc.students=self.students;
            vc.indexPath=indexPath;
            vc.path=self.path;

        }
    }

}
```

六、表视图控制器（UITableViewController）

如果使用通用视图控制器， 那么需要手动为表视图添加数据源和委托。而使用表视图控制则省去了这一步，自动完成数据源和委托设置。



视图的刷新



`- reloadData`

Reloads the rows and sections of the receiver.

Declaration

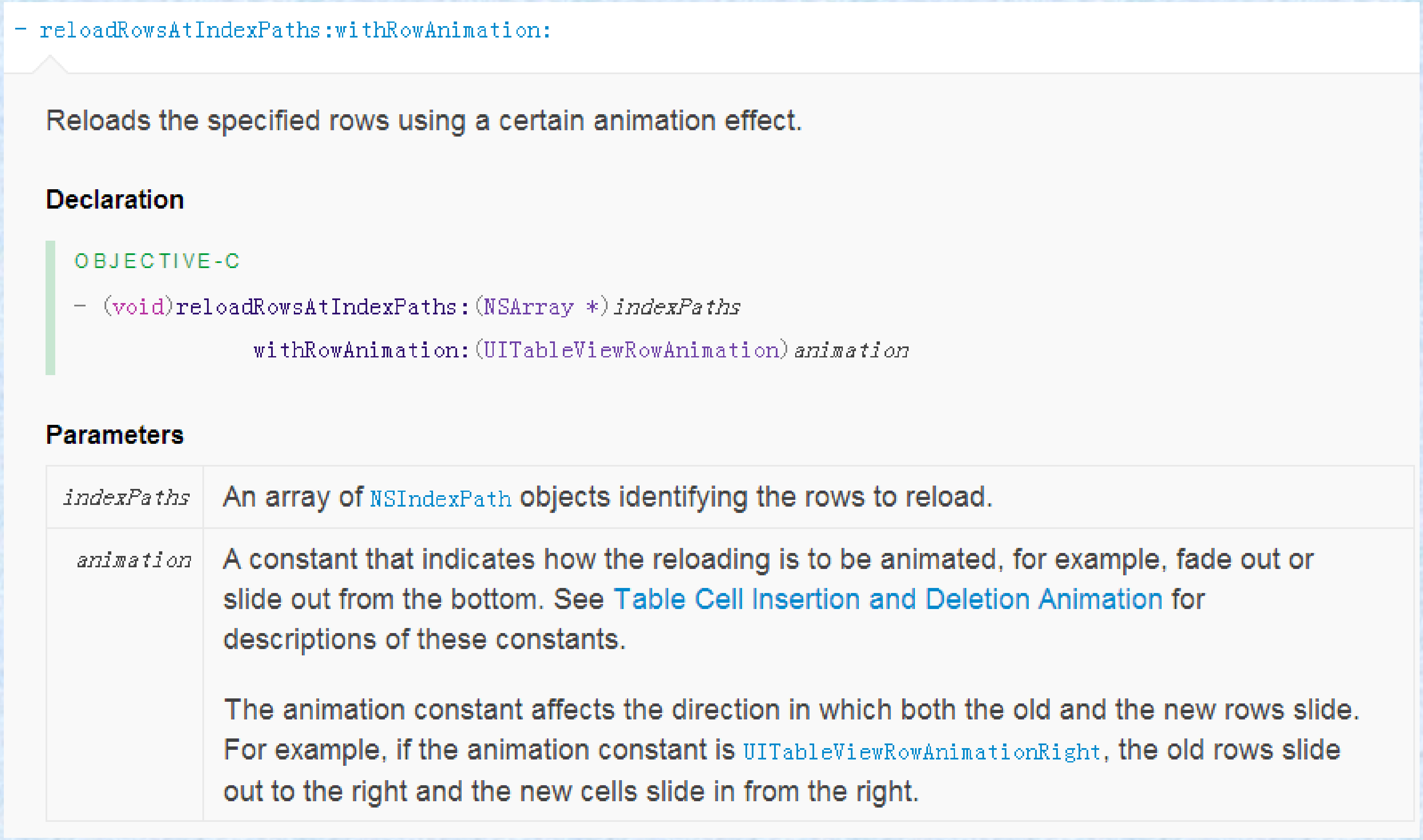
OBJECTIVE-C

```
- (void)reloadData
```

Discussion

Call this method to reload all the data that is used to construct the table, including cells, section headers and footers, index arrays, and so on. For efficiency, the table view redisplays only those rows that are visible. It adjusts offsets if the table shrinks as a result of the reload. The table view's delegate or data source calls this method when it wants the table view to completely reload its data. It should not be called in the methods that insert or delete rows, especially within an animation block implemented with calls to `beginUpdates` and `endUpdates`

视图的刷新



- reloadRowsAtIndexPaths:withRowAnimation:

Reloads the specified rows using a certain animation effect.

Declaration

OBJECTIVE-C

```
- (void)reloadRowsAtIndexPaths:(NSArray *)indexPaths
                withRowAnimation:(UITableViewRowAnimation)animation
```

Parameters

indexPaths	An array of `NSIndexPath` objects identifying the rows to reload.
animation	A constant that indicates how the reloading is to be animated, for example, fade out or slide out from the bottom. See Table Cell Insertion and Deletion Animation for descriptions of these constants. The animation constant affects the direction in which both the old and the new rows slide. For example, if the animation constant is `UITableViewRowAnimationRight`, the old rows slide out to the right and the new cells slide in from the right.