Class

UITableViewCell

A cell in a table view.

Declaration

```
class UITableViewCell : UIView
```

Overview

This class includes properties and methods for setting and managing cell content and background (including text, images, and custom views), managing the cell selection and highlight state, managing accessory views, and initiating the editing of the cell contents.

When creating cells, you can customize them yourself or use one of several predefined styles. The predefined cell styles are the simplest option. With the predefined styles, the cell provides label and image subviews whose positions and styling are fixed. All you have to do is provide the text and image content to go into those fixed views. To use a cell with a predefined style, initialize it using the `init(style:reuseIdentifier:)` method or configure the cell with that style in Xcode. To set the text and images of the cell, use the `textLabel`, `detailTextLabel`, and `imageView` properties.

If you want to go beyond the predefined styles, you can add subviews to the `contentView`property of the cell. When adding subviews, you are responsible for positioning those views and setting their content yourself.

Whether you use a predefined or custom cell, you can change the cell's background using the `backgroundView` property or by changing the inherited `backgroundColor` property. In iOS 7, cells have a white background by default; in earlier versions of iOS, cells inherit the background color of the enclosing table view. If you want to change the background color of a cell, do so in the `tableView(_:willDisplay:forRowAt:)` method of your table view delegate.

Topics

Initializing a UITableViewCell Object

`init(style: UITableViewCell.CellStyle, reuseIdentifier: String?)`

Initializes a table cell with a style and a reuse identifier and returns it to the caller.

`init?(coder: NSCoder)`

Reusing Cells

```
var reuseIdentifier: String?
```

A string used to identify a cell that is reusable.

```
func prepareForReuse()
```

Prepares a reusable cell for reuse by the table view's delegate.

Managing the Predefined Content

```
var textLabel: UILabel?
```

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

```
var detailTextLabel: UILabel?
```

Returns the secondary label of the table cell if one exists.

```
var imageView: UIImageView?
```

Returns the image view of the table cell.

Accessing Views of the Cell Object

```
var contentView: UIView
```

Returns the content view of the cell object.

```
var backgroundView: UIView?
```

The view used as the background of the cell.

```
var selectedBackgroundView: UIView?
```

The view used as the background of the cell when it is selected.

```
var multipleSelectionBackgroundView: UIView?
```

The background view to use for a selected cell when the table view allows multiple row selections.

Managing Accessory Views

```
var accessoryType: UITableViewCell.AccessoryType
```

The type of standard accessory view the cell should use (normal state).

```
var accessoryView: UIView?
```

A view that is used, typically as a control, on the right side of the cell (normal state).

```
var editingAccessoryType: UITableViewCell.AccessoryType
```

The type of standard accessory view the cell should use in the table view’s editing state.

```
var editingAccessoryView: UIView?
```

A view that is used typically as a control on the right side of the cell when it is in editing mode.

Managing Cell Selection and Highlighting

```
var isSelected: Bool
```

A Boolean value that indicates whether the cell is selected.

```
var selectionStyle: UITableViewCell.SelectionStyle
```

The style of selection for a cell.

```
func setSelected(Bool, animated: Bool)
```

Sets the selected state of the cell, optionally animating the transition between states.

```
var isHighlighted: Bool
```

A Boolean value that indicates whether the cell is highlighted.

```
func setHighlighted(Bool, animated: Bool)
```

Sets the highlighted state of the cell, optionally animating the transition between states.

Editing the Cell

```
var isEditing: Bool
```

A Boolean value that indicates whether the cell is in an editable state.

```
func setEditing(Bool, animated: Bool)
```

Toggles the receiver into and out of editing mode.

```
var editingStyle: UITableViewCell.EditingStyle
```

The editing style of the cell.

```
var showingDeleteConfirmation: Bool
```

Returns whether the cell is currently showing the delete-confirmation button.

```
var showsReorderControl: Bool
```

A Boolean value that determines whether the cell shows the reordering control.

Dragging the Row

```
var userInteractionEnabledWhileDragging: Bool
```

A Boolean value indicating whether users can interact with a cell while it is being dragged.

```
func dragStateDidChange(UITableViewCell.DragState)
```

Notifies the cell that its drag status changed.

```
enum UITableViewCell.DragState
```

Constants indicating the current state of a row involved in a drag operation.

Adjusting to State Transitions

```
func willTransition(to: UITableViewCell.StateMask)
```

Called on the cell just before it transitions between cell states.

```
func didTransition(to: UITableViewCell.StateMask)
```

Called on the cell just after it transitions between cell states.

Managing Content Indentation

```
var indentationLevel: Int
```

The indentation level of the cell’s content.

```
var indentationWidth: CGFloat
```

The width for each level of indentation of a cell's content.

```
var shouldIndentWhileEditing: Bool
```

A Boolean value that controls whether the cell background is indented when the table view is in editing mode.

```
var separatorInset: UIEdgeInsets
```

The inset values for the cell’s content.

Managing Focus

```
var focusStyle: UITableViewCell.FocusStyle
```

The appearance of the cell when focused.

Constants

`enum UITableViewCell.CellStyle`

An enumeration for the various styles of cells.

`enum UITableViewCell.SelectionStyle`

The style of selected cells.

`enum UITableViewCell.EditingStyle`

The editing control used by a cell.

`enum UITableViewCell.AccessoryType`

The type of standard accessory control used by a cell.

`struct UITableViewCell.StateMask`

Constants used to determine the new state of a cell as it transitions between states.

`enum UITableViewCell.SeparatorStyle`

The style for cells used as separators.

`enum UITableViewCell.FocusStyle`

The style of focused cells.

Relationships

Inherits From

`UIView`