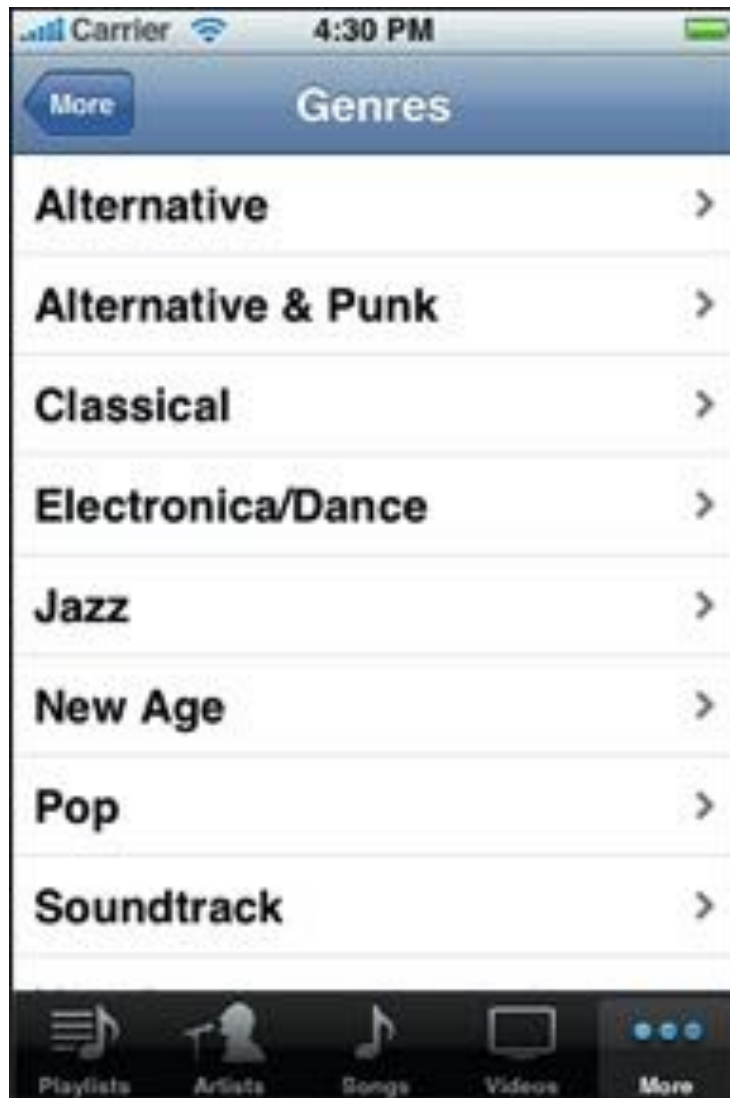

UITableView

강사 주영민

UITableView

- 리스트의 형태로 정보를 보여주는 View

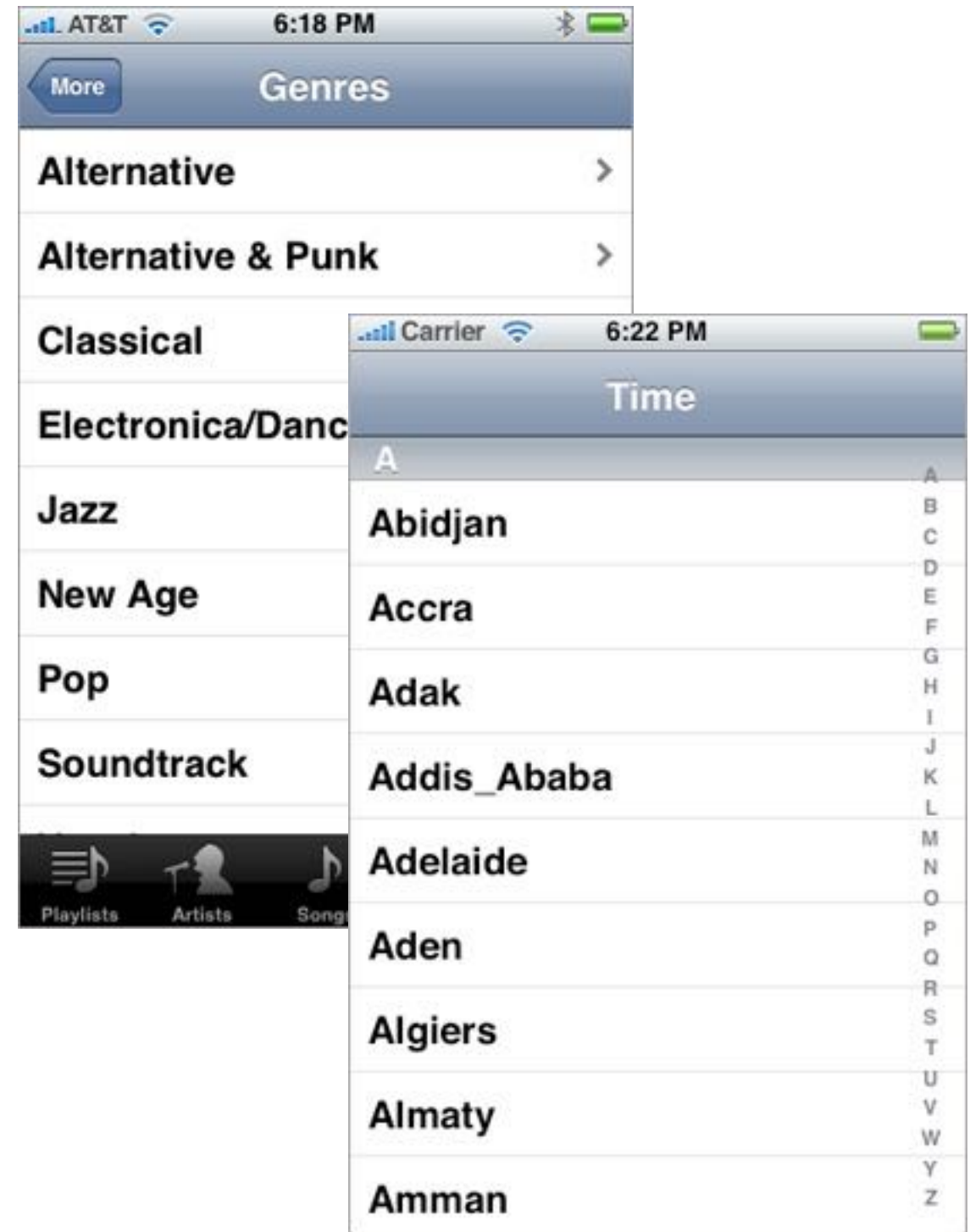


Style

- plain
- Grouped

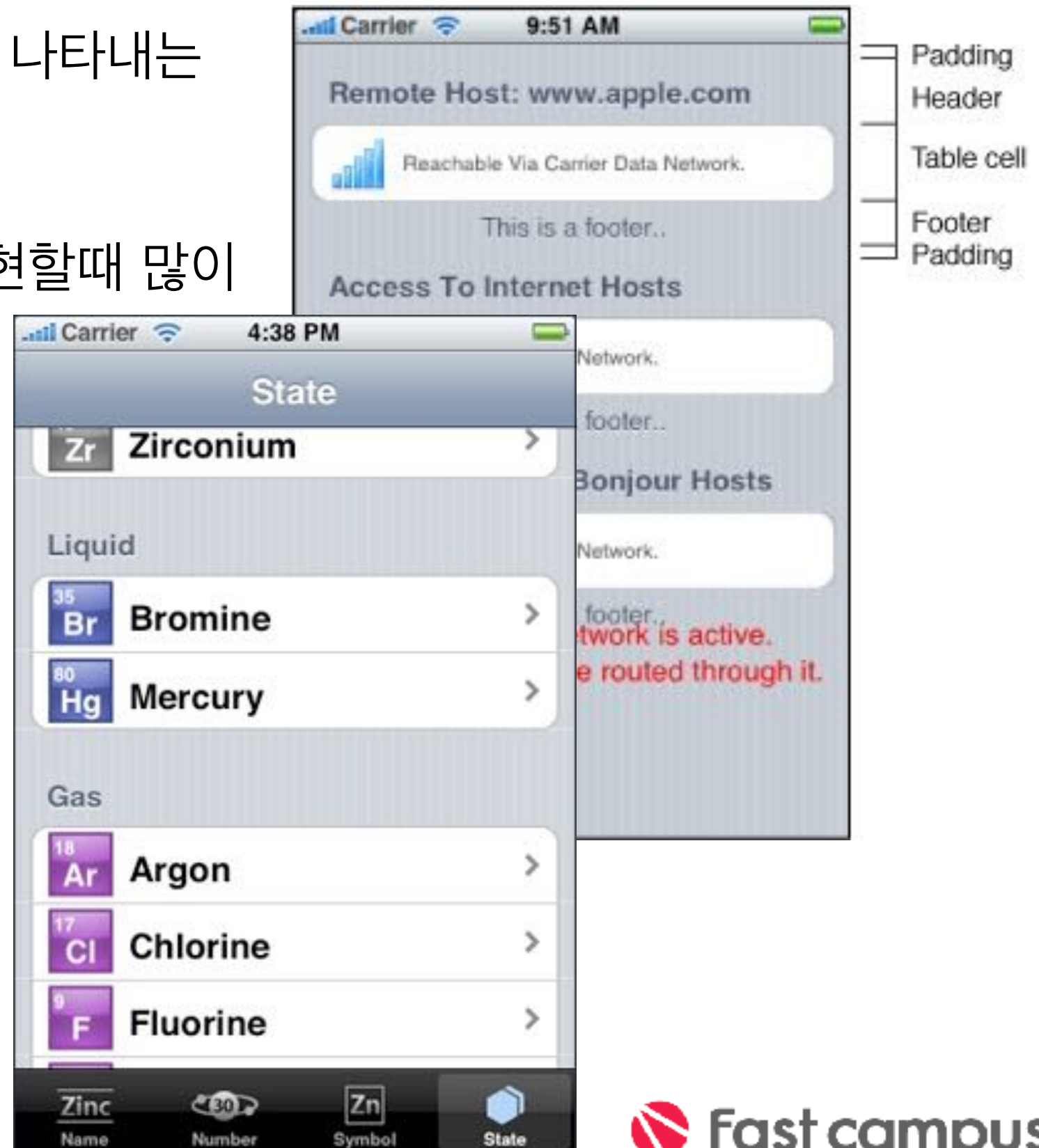
Plain Table Views

- 기본적인 TableView
- 여러개의 Section을 가질수 있다.
- 한 Section에는 여러개의 Row를 포함하고 있다.
- 각각의 Section에는 Section을 표시하는 header, Footer title을 사용할수 있다.
- Section을 빠르게 검색할수 있는 네비게이터 바를 index list 라고 부른다.



Grouped Table Views

- 각 Section을 Group의 형태로 나타내는 테이블 뷰
- 데이터의 디테일한 정보를 표현할때 많이 사용된다.



Protocol

- DataSource
- Delegte

DataSource

- 프로토콜을 사용하여 테이블뷰에서 보여줄 데이터를 관리할 대리인의 역할을 정의해 둔 것
- 역할

@requires

- 테이블 뷰의 각 섹션별 열의 개수를 설정
- Row별 Cell객체

@optional

- 테이블 뷰 섹션의 개수를 설정

등등

DataSource

```
public protocol UITableViewDataSource : NSObjectProtocol {
    @available(iOS 2.0, *)
    public func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int

    // Row display. Implementers should *always* try to reuse cells by setting each cell's
    // reuseIdentifier and querying for available reusable cells with dequeueReusableViewWithReuseIdentifier:
    // Cell gets various attributes set automatically based on table (separators) and data source
    // (accessory views, editing controls)

    @available(iOS 2.0, *)
    public func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
    UITableViewCell

    @available(iOS 2.0, *)
    optional public func numberOfSections(in tableView: UITableView) -> Int // Default is 1 if
    not implemented

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, titleForHeaderInSection section:
    Int) -> String? // fixed font style. use custom view (UILabel) if you want something different

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, titleForFooterInSection section:
    Int) -> String?
```


Delegate

- 프로토콜을 사용하여 테이블뷰의 대리자로써의 수행할 수 있는 역할들을 정의해 둔 것
- 역할
 1. 헤더 또는 풋터의 높이를 설정 : Variable height support
 2. 헤더 또는 풋터 뷰를 제공 : Section Informations.
 3. 셀을 선택하였을 때 수행할 동작 관리 : Selection
 4. 셀의 삭제 될 때의 동작 등의 관리 : Editing
 5. 기타 등등

Delegate

```
public protocol UITableViewDelegate : NSObjectProtocol, UIScrollViewDelegate {

    // Display customization

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, willDisplay cell:
UITableViewCell, forRowAt indexPath: IndexPath)

    @available(iOS 6.0, *)
    optional public func tableView(_ tableView: UITableView, willDisplayHeaderView view:
UIView, forSection section: Int)
    ...

    // Variable height support
    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, heightForRowAt indexPath:
IndexPath) -> CGFloat

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, heightForHeaderInSection
section: Int) -> CGFloat

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, heightForFooterInSection
section: Int) -> CGFloat
```

IndexPath

- cell의 위치를 나타낼 Data Type으로 Secion정보와 해당 섹션에 서의 열(Row) 정보를 가지고 있다.

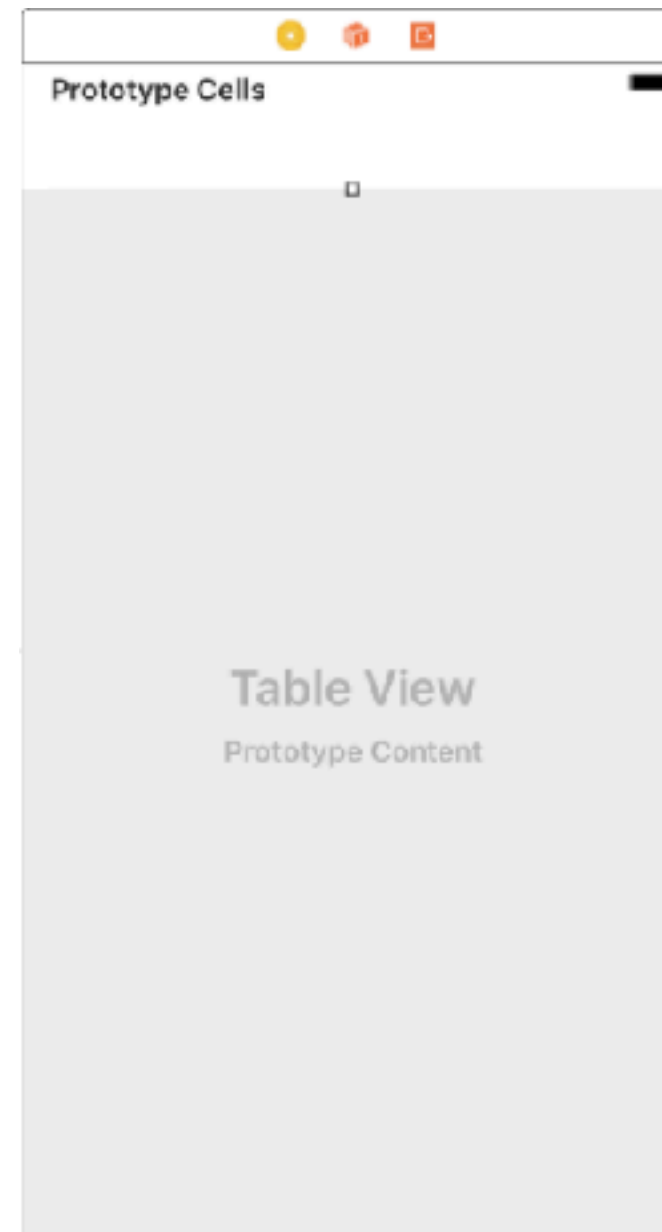
```
extension IndexPath {  
    /// Initialize for use with `UITableView` or `UICollectionView`  
    public init(row: Int, section: Int)  
  
    /// The section of this index path, when used with `UITableView`  
    ///  
    /// – precondition: The index path must have exactly two elements  
    public var section: Int  
  
    /// The row of this index path, when used with `UITableView`.  
    ///  
    /// – precondition: The index path must have exactly two elements  
    public var row: Int  
}
```

테이블 뷰 만들기

1. 테이블 UI 생성
2. DataSource 함수 만들기(section, row, 갯수)
3. 각 indexPath당 Cell설정

테이블 UI

- TableView 화면에 이동 후 AutoLayout 설정
- Table View Cell 추가



테이블 UI

- Table View Cell Identifier 설정

Table View Cell

Style

Identifier

Selection

Accessory

Editing Acc.

Focus Style

Indentation
Level Width

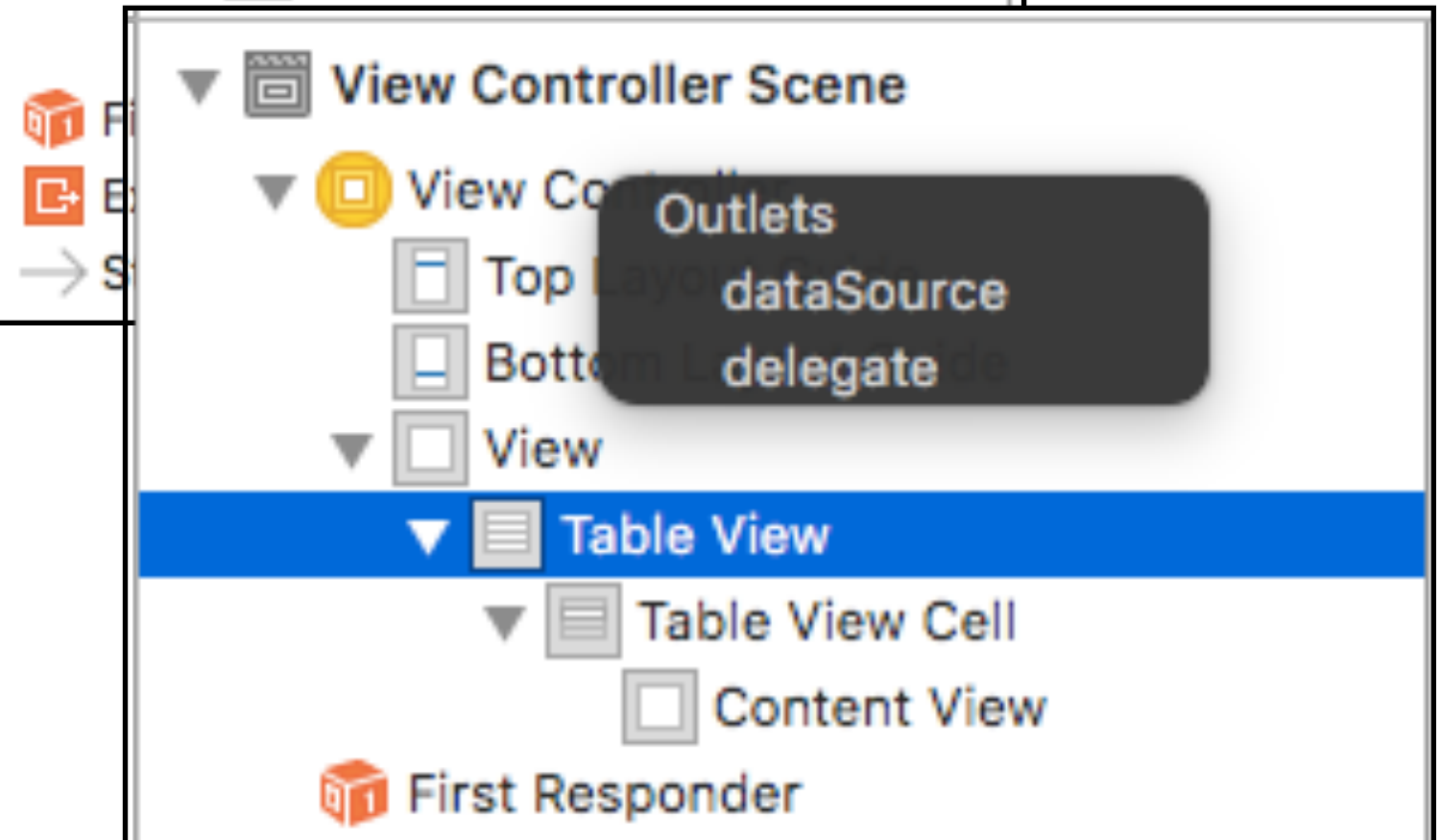
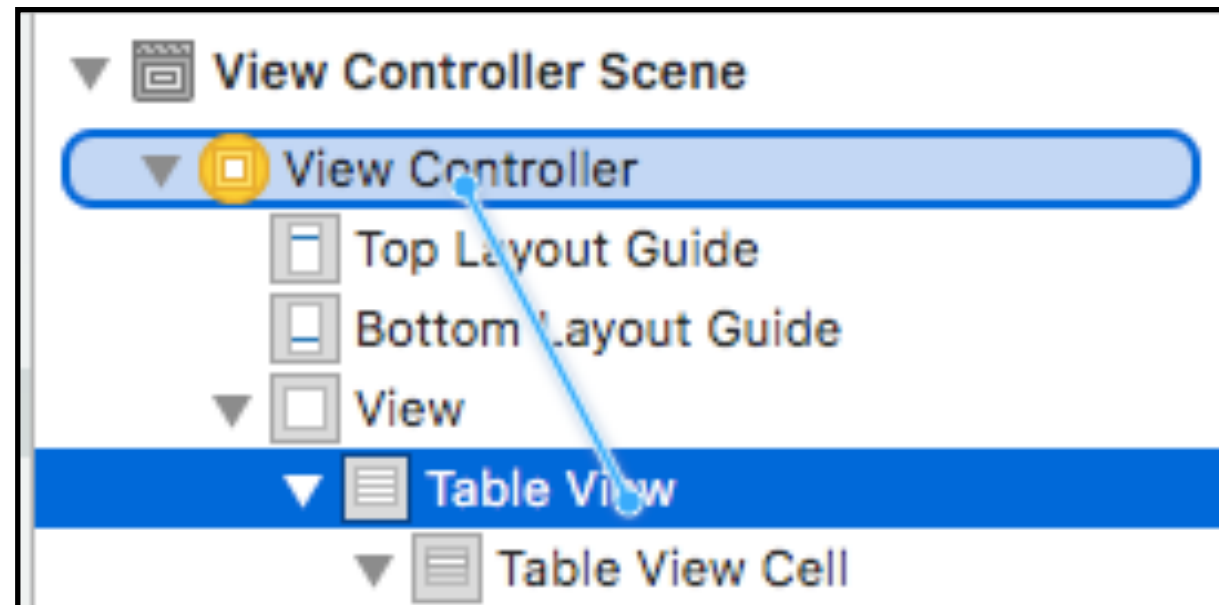
☒ Indent While Editing

☐ Shows Re-order Controls

Separator

테이블 UI

- DataSource 연결



UITableViewDataSource 구현

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

```
    //섹션별 row의 갯수 리턴 코드 구현
```

```
}
```

```
func tableView(_ tableView: UITableView, cellForRowAt  
indexPath: IndexPath) -> UITableViewCell {
```

```
    //cell 구현
```

```
}
```


TableViewCell Reusable

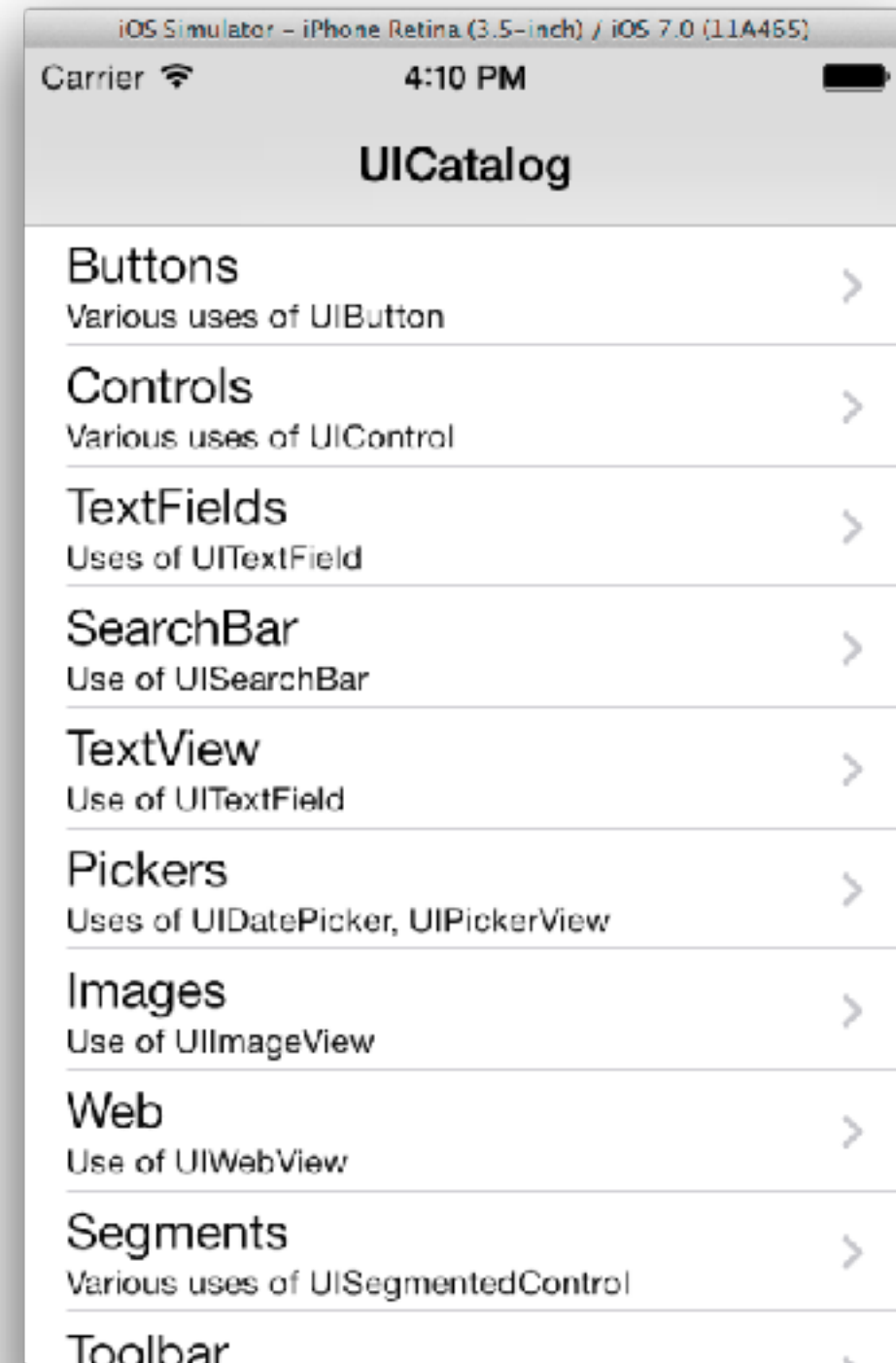
```
func tableView(_ tableView: UITableView,
               cellForRowAt indexPath: IndexPath) -> UITableViewCell
{
    //cell instance 가져오기
    let cell = tableView.dequeueReusableCell(withIdentifier:
"Reuse ID", for: indexPath)

    //Cell Data 설정
    cell.textLabel?.text = "\(indexPath.row)"

    return cell
}
```

재사용(reuse)

- 테이블 뷰는 한 줄 한 줄이 굉장히 비슷한 모습을 가지고 있다.
- 수백개의 테이블 리스트가 있다고 가정할 때...
- 공통적인 레이아웃을 사용하여 여러번 화면을 보여줄 필요가 있는 경우 Cell을 재사용



같이 해봐요

- 기본 TableView 만들기

UITableViewDelegate func

- Delegate 확인 해보기

실습

- 친구 이름 Array만들기
- tableView 만들기
- 친구이름 tableView에 뿌리기

UITableViewCell

- display text, images, or other kinds of content.
- background views for both normal and selected states.
- Cells can also have accessory views,

UITableViewCell

- contentView
 - .textLabel
 - detailTextLabel
 - imageView
- accessoryView

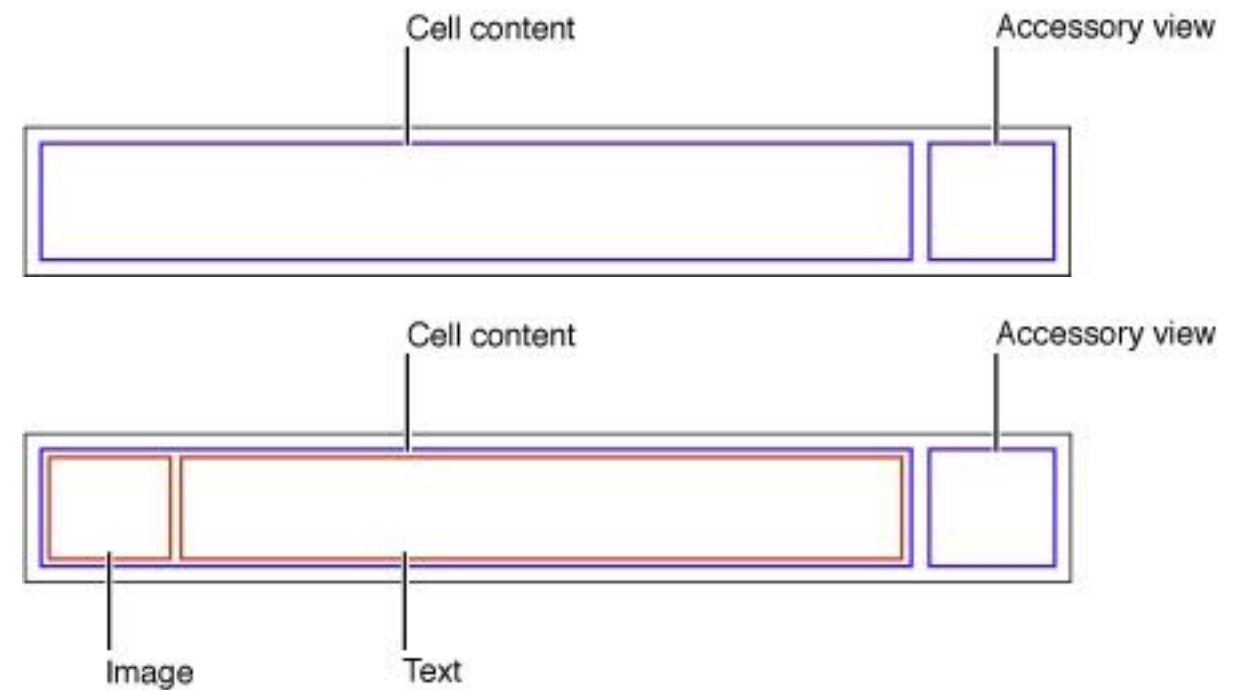












Table View Cells Style

UITableViewCellStyleDefault	
	Clown Fish
	Dew Drop
	Earth
	Earth Horizon
	Flowing Rock
	Lady Bug
	Snowy Hills
	Stones
	Sweeping Current
	Zebra











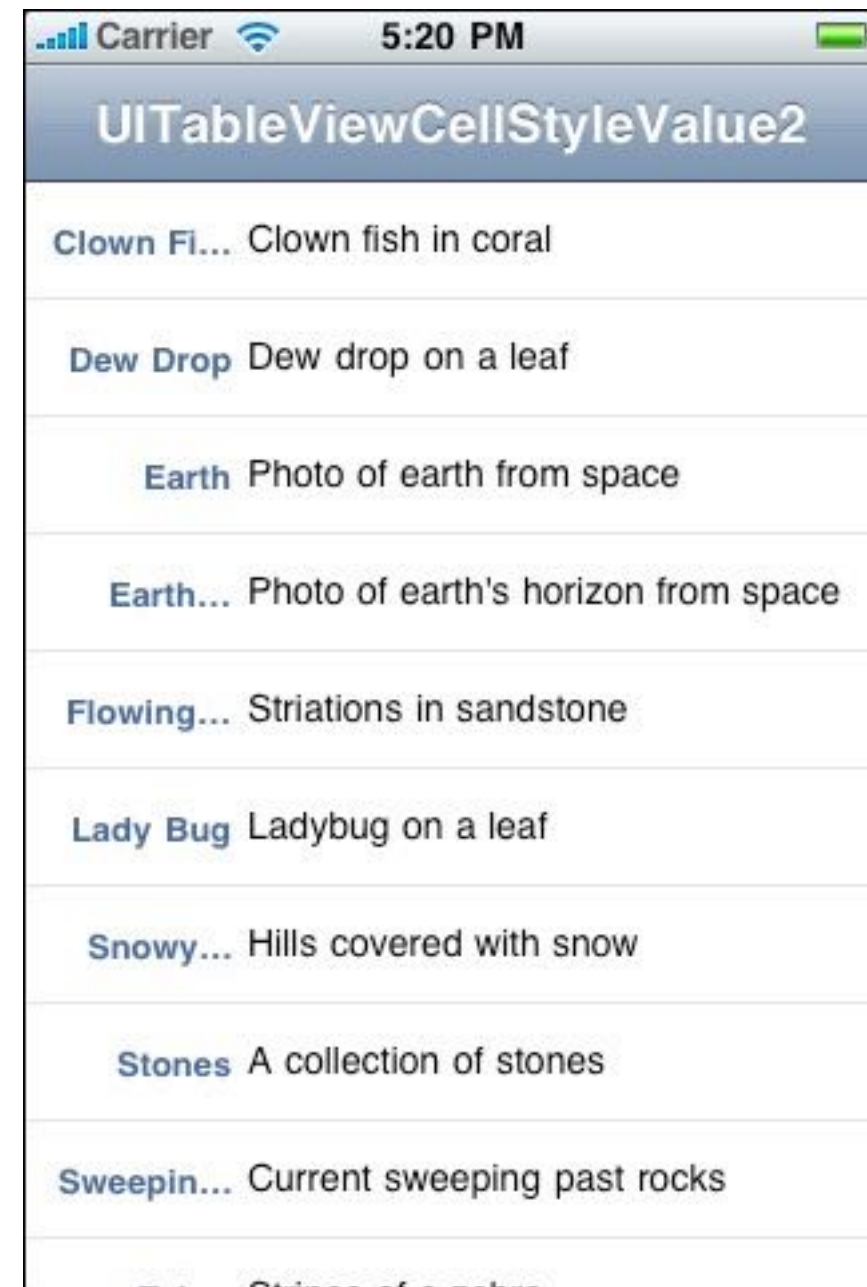
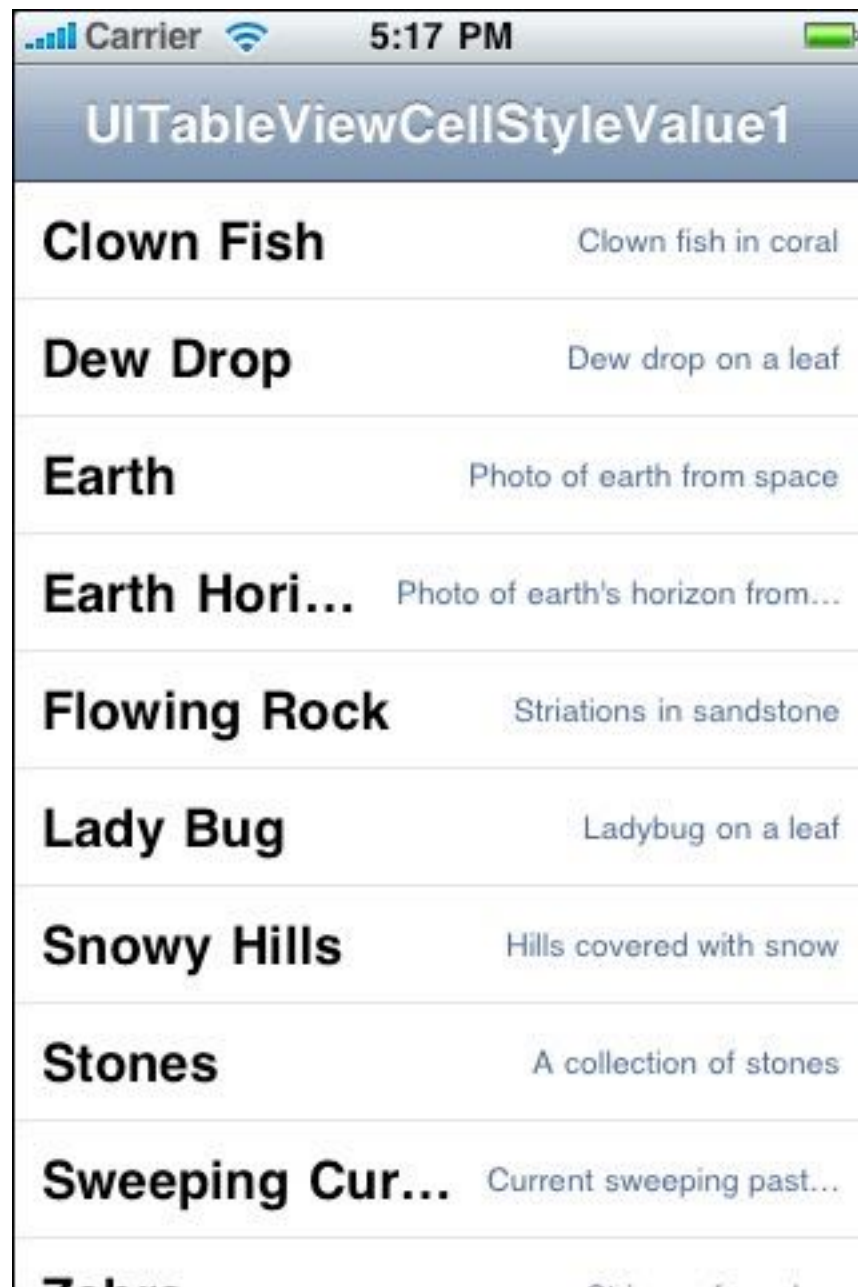
UITableViewCellStyleSubtitle	
	Clown Fish Clown fish in coral
	Dew Drop Dew drop on a leaf
	Earth Photo of earth from space
	Earth Horizon Photo of earth's horizon from space
	Flowing Rock Striations in sandstone
	Lady Bug Ladybug on a leaf
	Snowy Hills Hills covered with snow
	Stones A collection of stones
	Sweeping Current Current sweeping past rocks
	Zebra

Table View Cells Style



TableView Cell 만들기

- 다양한 타입의 Cell을 만들어 봅시다.

UITableView Practice

강사 주영민

기본 테이블 뷰 만들기

7	
8	
9	
10	
11	
12	
13	

데이터를 통한 테이블 뷰 관리

Emu
Giraffe
Greater Rhea
Hippopotamus
Horse
Koala
Lion

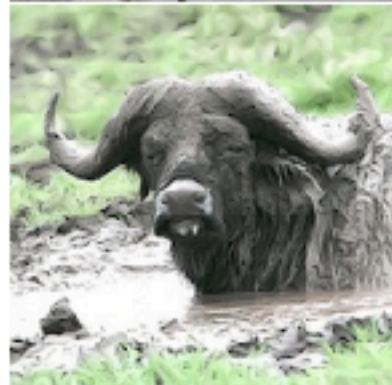
Cell에 image 넣기



Bear





Black Swan



Buffalo

Cell에 Accessory 넣기

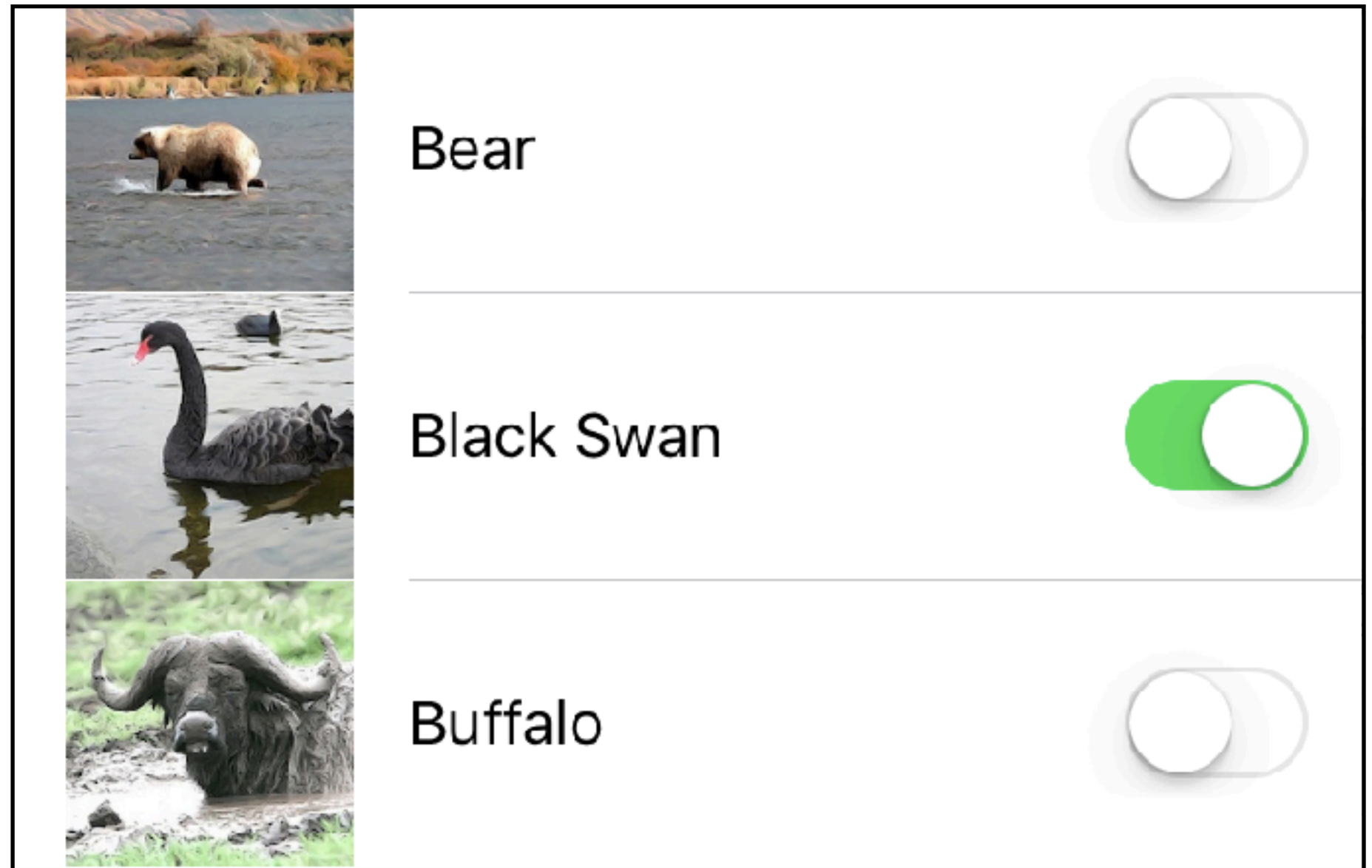
	Bear	✓
	Black Swan	✓

```
typedef NS_ENUM(NSInteger,  
UITableViewCellAccessoryType) {  
    UITableViewCellAccessoryNone,  
    UITableViewCellAccessoryDisclosureIndicator,  
    UITableViewCellAccessoryDetailDisclosureButton,  
    UITableViewCellAccessoryCheckmark,  
    UITableViewCellAccessoryDetailButton  
};
```

✓

Cell에 Custom accessory

```
@property (nonatomic, strong, nullable) UIView *accessoryView;
```


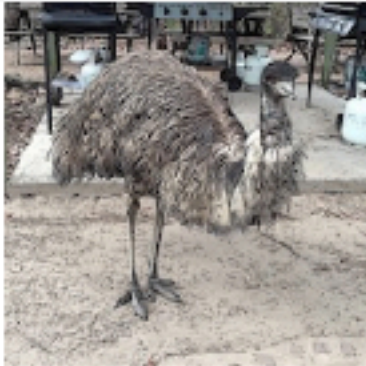



Row 선택시 행동 정하기

	Bear	✓
	Black Swan	✓
	Buffalo	✓

```
1 TableViewExample[98079:9282684] Black Swan이 선택 되었습니다.  
4 TableViewExample[98079:9282684] Camel이 선택 되었습니다.  
2 TableViewExample[98079:9282684] Tasmania Devil이 선택 되었습니다.
```

Section 나누기 & 타이틀 지정

T		
	Tasmania Devil	✓
E		
	Emu	✓
M		
		

TableHeaderView 만들기

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

