

델리게이트 패턴

UIPickerView



UIPickerView

component 0

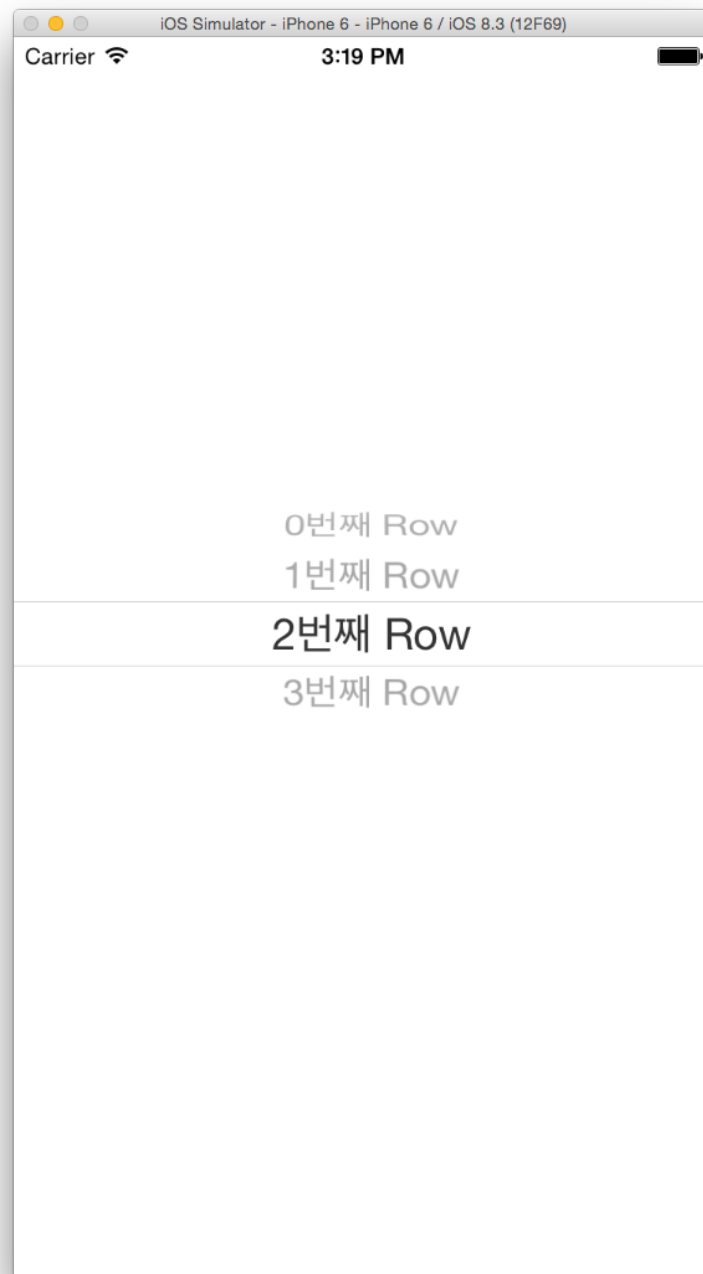
component 1



row 0

row 1

UIPickerView의 동작방식



표시 하려는 component가 몇개?

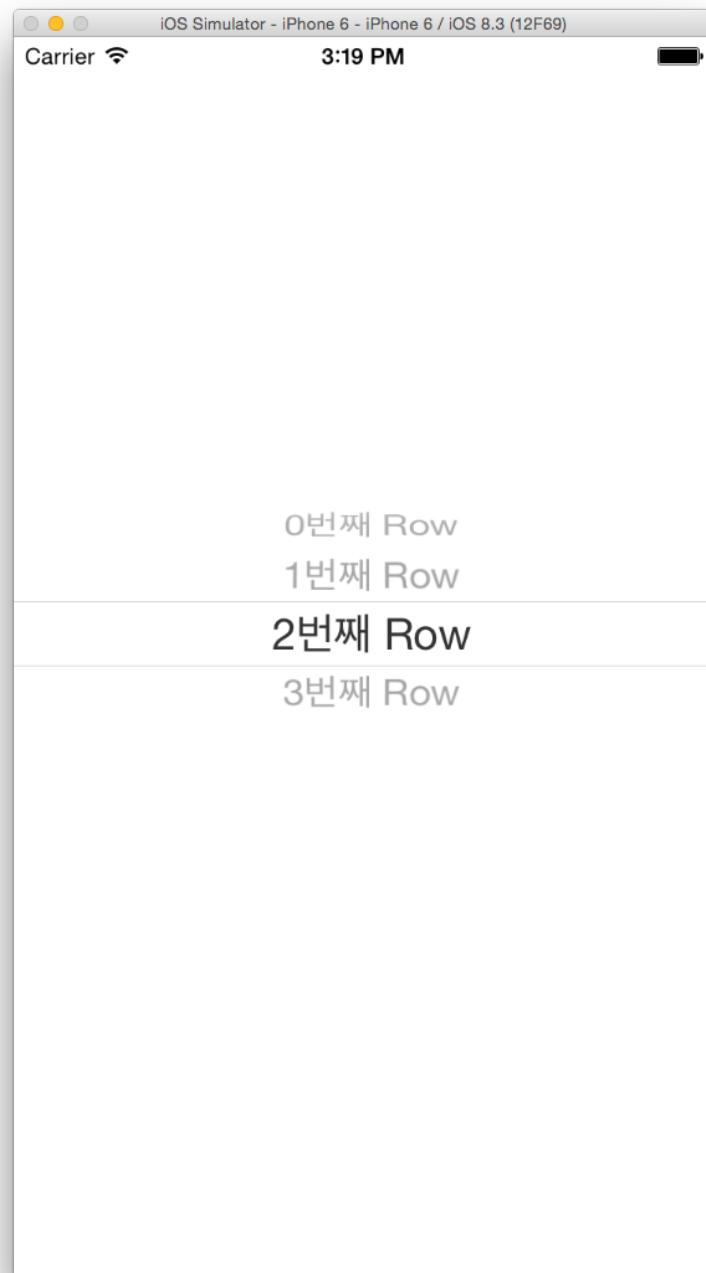
1개

0번째 component에 row가 몇개?

4개

DataSource

UIPickerView의 동작방식



```
func numberOfComponents(  
    in pickerView: UIPickerView) -> Int
```



```
func pickerView(_ pickerView: UIPickerView,  
    numberOfRowsInComponent component: Int) -> Int
```



DataSource

Protocol

특정한 목적을 가진 프로퍼티와 메소드선언들의 집합

어떠한 목적을 가진 메소드들의 집합

```
protocol protocol_name: NSObjectProtocol
{
    Method1
    Method2
    .
    .
    optional Mehtod3
    .
    .
}
```

UIPickerViewDataSource

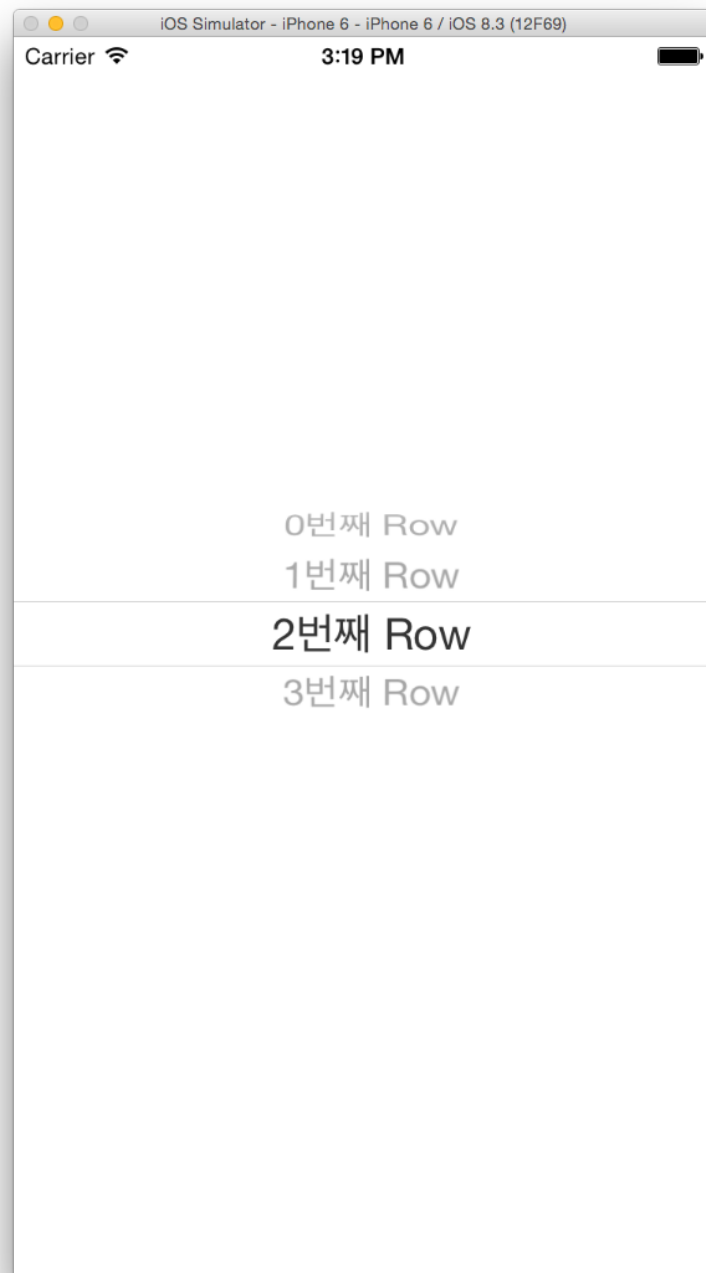
```
public protocol UIPickerViewDataSource : NSObjectProtocol {  
  
    // returns the number of 'columns' to display.  
    @available(iOS 2.0, *)  
    public func numberOfComponents(in pickerView: UIPickerView) -> Int  
  
    // returns the # of rows in each component..  
    @available(iOS 2.0, *)  
    public func pickerView(_ pickerView: UIPickerView,  
        numberOfRowsInComponent component: Int) -> Int  
}
```

실제 코드

```
func numberOfComponents(in pickerView: UIPickerView) -> Int{  
    return 1  
}
```

```
func pickerView(_ pickerView: UIPickerView,  
                numberOfRowsInComponent component: Int) -> Int{  
    return 4  
}
```


UIPickerView의 동작방식



0번째 표시할 데이터 주세요

0번째 데이터 여기있어요.

1번째 표시할 데이터 주세요

1번째 데이터 여기있어요.

2번째 표시할 데이터 주세요

2번째 데이터 여기있어요.

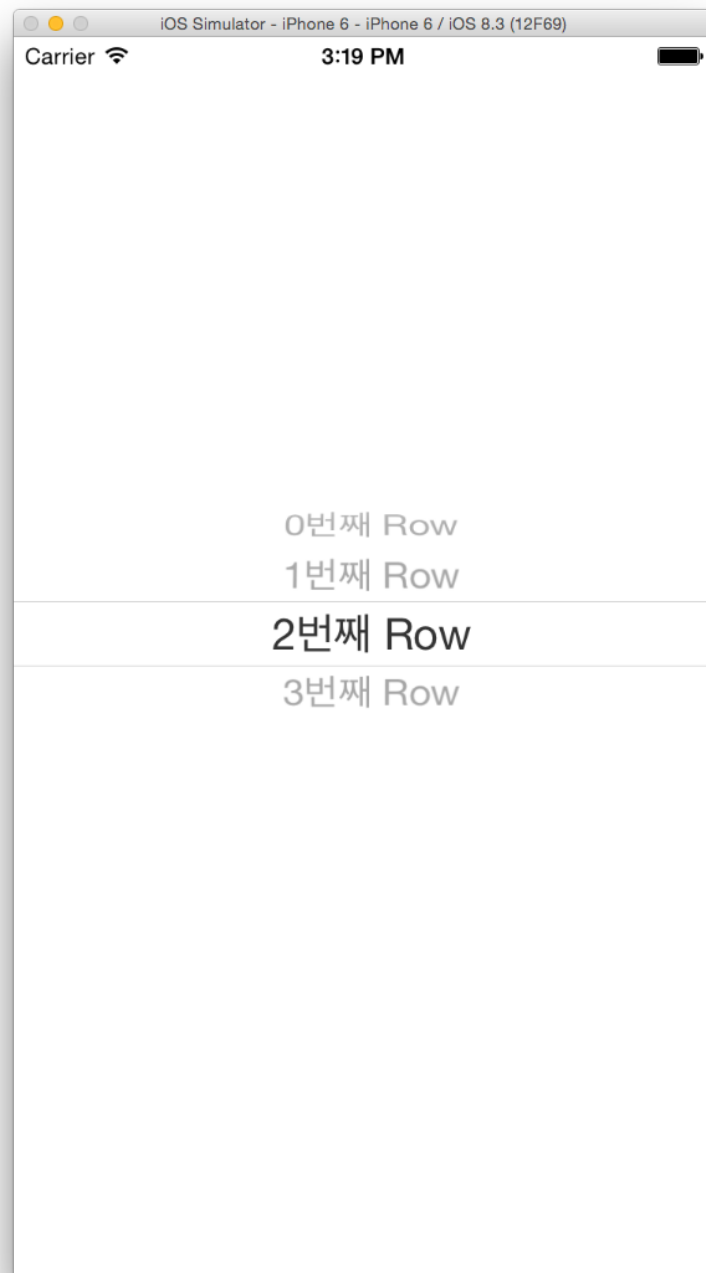
3번째 표시할 데이터 주세요

3번째 데이터 여기있어요.

```
func pickerView(_ pickerView: UIPickerView,  
               titleForRow row: Int,  
               forComponent component: Int) -> String?
```

Delegate

PickerView의 동작방식



0번째 component의
1번째 row가 선택되었습니다.



```
func pickerView(_ pickerView: UIPickerView,  
didSelectRow row: Int, inComponent component: Int)
```



Delegate

UIPickerViewDelegate

```
public protocol UIPickerViewDelegate : NSObjectProtocol {

    // returns width of column and height of row for each component.
    @available(iOS 2.0, *)
    optional public func pickerView(_ pickerView: UIPickerView, widthForComponent component: Int) -> CGFloat

    @available(iOS 2.0, *)
    optional public func pickerView(_ pickerView: UIPickerView, heightForComponent component: Int) -> CGFloat

    @available(iOS 2.0, *)
    optional public func pickerView(_ pickerView: UIPickerView, titleForRow row: Int, forComponent component: Int) -> String?

    @available(iOS 6.0, *)
    optional public func pickerView(_ pickerView: UIPickerView, attributedTitleForRow row: Int, forComponent component: Int) -> NSAttributedString? // attributed title is favored if both methods are implemented

    @available(iOS 2.0, *)
    optional public func pickerView(_ pickerView: UIPickerView, viewForRow row: Int, forComponent component: Int, reusing view: UIView?) -> UIView

    @available(iOS 2.0, *)
    optional public func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int, inComponent component: Int)
}
```

실제 코드

```
class ViewController: UIViewController, UIPickerViewDataSource,
UIPickerViewDelegate {
    @IBOutlet weak var pickerView: UIPickerView!
    override func viewDidLoad() {
        super.viewDidLoad()
        pickerView.dataSource = self
        pickerView.delegate = self
    }
    func numberOfComponents(in pickerView: UIPickerView) -> Int {
        return 1
    }
    func pickerView(_ pickerView: UIPickerView,
                    numberOfRowsInComponent component: Int) -> Int {
        return 4
    }
    func pickerView(_ pickerView: UIPickerView, titleForRow row: Int,
                    forComponent component: Int) -> String? {
        let title = "\(row)번째 Row"
        return title
    }
}
```

배열을 이용한 UIPickerView

```
class ViewController: UIViewController, UIPickerViewDelegate, UIPickerViewDataSource {
    let arrbts = ["진", "슈가", "제이홉", "RM", "지민", "뷔", "정국"]
    @IBOutlet weak var pickerView: UIPickerView!
    override func viewDidLoad() {
        super.viewDidLoad()
        pickerView.dataSource = self
        pickerView.delegate = self
    }
    func numberOfComponents(in pickerView: UIPickerView) -> Int{
        return 1
    }
    func pickerView(_ pickerView: UIPickerView,
                    numberOfRowsInComponent component: Int) -> Int{
        return arrbts.count
    }
    func pickerView(_ pickerView: UIPickerView, titleForRow row: Int,
                    forComponent component: Int) -> String?{
        let title = arrbts[row]
        return title
    }
}
```

PickerView 이벤트

```
class ViewController: UIViewController, UIPickerViewDelegate, UIPickerViewDataSource {
    var arrbts = ["진", "슈가", "제이홉", "RM", "지민", "뷔", "정국"]
    @IBOutlet weak var pickerView: UIPickerView!
    @IBOutlet weak var label: UILabel!
    override func viewDidLoad() {
        super.viewDidLoad()
        pickerView.dataSource = self
        pickerView.delegate = self
    }
    func numberOfComponents(in pickerView: UIPickerView) -> Int{
        return 1
    }
    func pickerView(_ pickerView: UIPickerView,
                    numberOfRowsInComponent component: Int) -> Int{
        return arrbts.count
    }
    func pickerView(_ pickerView: UIPickerView, titleForRow row: Int,
                    forComponent component: Int) -> String?{
        let title = arrbts[row]
        return title
    }
    func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int, inComponent
component: Int) {
        label.text = arrbts[row]
    }
}
```

Multi Component

```
class ViewController: UIViewController, UIPickerViewDataSource, UIPickerViewDelegate {  
  
    var arrbts = ["진", "슈가", "제이홉", "RM", "지민", "뷔", "정국"]  
    var arrExo = ["시우민", "디오", "첸", "백현", "세훈", "루한", "레이", "찬열", "수호", "크리스", "타오"]  
  
    @IBOutlet weak var lblbts: UILabel!  
    @IBOutlet weak var lblExo: UILabel!  
    @IBOutlet weak var pickerView: UIPickerView!  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        pickerView.dataSource = self  
        pickerView.delegate = self  
    }  
}
```

Multi Component

```
func numberOfComponents(in pickerView: UIPickerView) -> Int {  
    return 2  
}
```

```
func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component:  
Int) -> Int {  
    if component == 0 {  
        return arrbts.count  
    } else {  
        return arrExo.count  
    }  
}
```


Multi Component

```
func pickerView(_ pickerView: UIPickerView, titleForRow row: Int, forComponent
component: Int) -> String? {
    if component == 0 {
        return arrbts[row]
    } else {
        return arrExo[row]
    }
}

func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int, inComponent
component: Int) {
    if component == 0 {
        lblbts.text = arrbts[row]
    } else {
        lblExo.text = arrbts[row]
    }
}
}
```

View 재활용

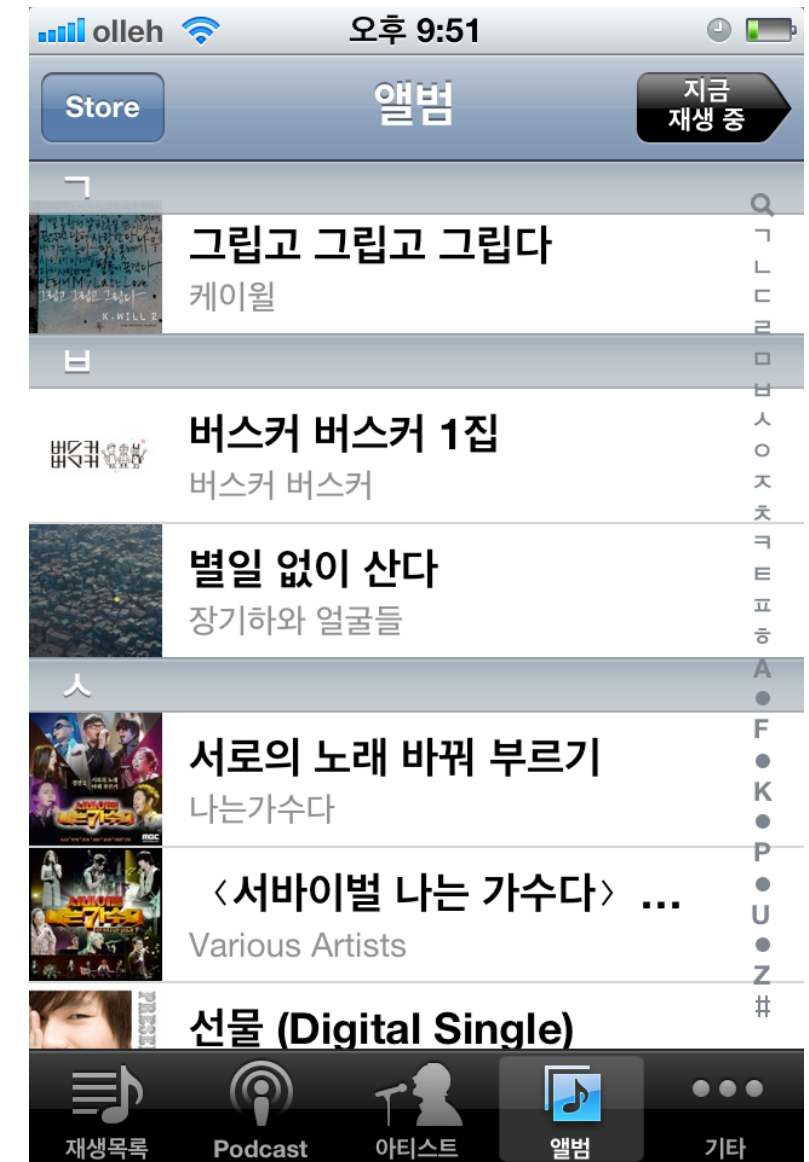
```
func pickerView(_ pickerView: UIPickerView, viewForRow row: Int, forComponent
component: Int, reusing view: UIView?) -> UIView {
    let tempLabel = view as? UILabel
    if let label = tempLabel {
        label.text = arrbts[row]
        return label
    } else {
        let label = UILabel()
        label.backgroundColor = UIColor.yellow
        label.textAlignment = .center
        label.text = arrbts[row]
        return label
    }
}
```

View 재활용

```
func pickerView(_ pickerView: UIPickerView, viewForRow row: Int, forComponent
component: Int, reusing view: UIView?) -> UIView {
    var label = view as? UILabel
    if label == nil {
        label = UILabel()
        label.backgroundColor = UIColor.yellow
        label.textAlignment = .center
    }
    label.text = arrbts[row]
    return label
}
```

UITableView

UITableView



UITableView

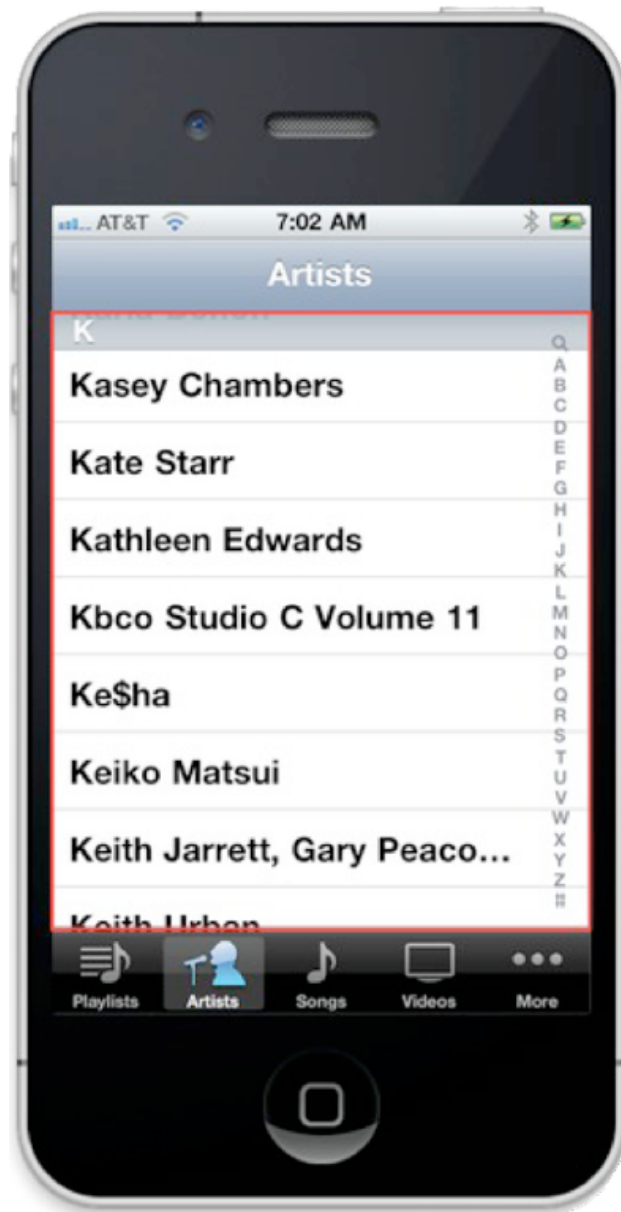


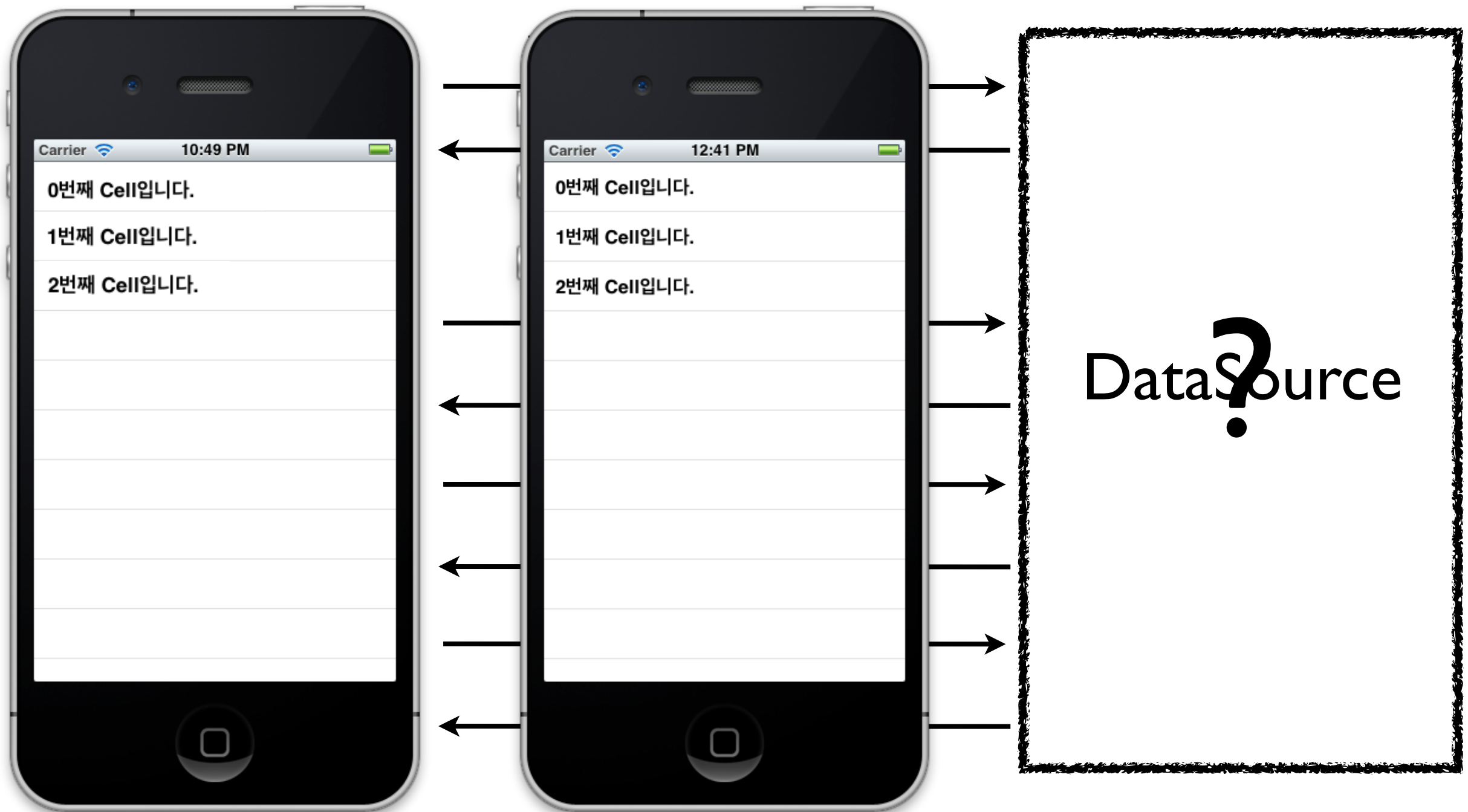
Table View
(UITableView)

K		Q
Kasey Chambers		A
Kate Starr		B
Kathleen Edwards		C
Kbco Studio C Volume 11		D
Ke\$ha		E
Keiko Matsui		F
Keith Jarrett, Gary Peaco...		G
Keith Urban		H
		I
		J
		K
		L
		M
		N
		O
		P
		Q
		R
		S
		T
		U
		V
		W
		X
		Y
		Z
		#

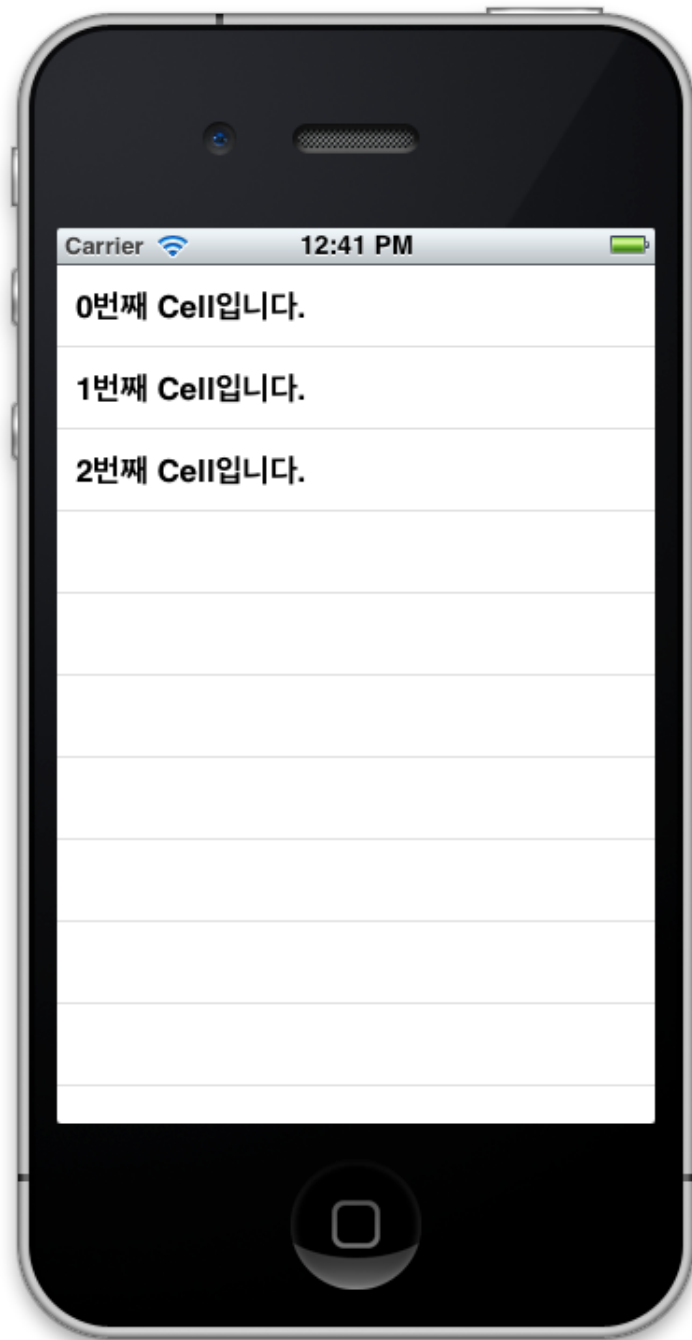
Table View Cell
(UITableViewCell)

K		Q
Kasey Chambers		A
Kate Starr		B
Kathleen Edwards		C
Kbco Studio C Volume 11		D
Ke\$ha		E
Keiko Matsui		F
Keith Jarrett, Gary Peaco...		G
Keith Urban		H
		I
		J
		K
		L
		M
		N
		O
		P
		Q
		R
		S
		T
		U
		V
		W
		X
		Y
		Z
		#

UITableView의 동작방식



UITableView의 동작방식



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

→

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

→

DataSource

UITableViewDataSource Protocol

```
public protocol UITableViewDataSource : NSObjectProtocol {

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

    @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?

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

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool

    @available(iOS 2.0, *)
    optional public func sectionIndexTitles(for tableView: UITableView) -> [String]?

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, sectionForSectionIndexTitle title: String, at index: Int) -> Int

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath)

    @available(iOS 2.0, *)
    optional public func tableView(_ tableView: UITableView, moveRowAt sourceIndexPath: IndexPath, to destinationIndexPath: IndexPath)

}
```

UITableView의 동작방식



몇 번째 섹션의
몇 번째 셀이 터치 되었어요.

```
func tableView(_tableView: UITableView,  
didSelectRowAtIndexPath indexPath: IndexPath)
```

Delegate

UITableViewDelegate Protocol

```
public protocol UITableViewDelegate : NSObjectProtocol, UIScrollViewDelegate {
    @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)

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

    @available(iOS 6.0, *)
    optional public func tableView(_ tableView: UITableView, didEndDisplaying cell: UITableViewCell, forRowAt indexPath:
IndexPath)

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

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

    @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

    RowAt indexPath: IndexPath) -> Bool

    @available(iOS 5.0, *)
    optional public func tableView(_ tableView: UITableView, canPerformAction action: Selector, forRowAt indexPath: IndexPath,
withSender sender: Any?) -> Bool

    @available(iOS 5.0, *)
    optional public func tableView(_ tableView: UITableView, performAction action: Selector, forRowAt indexPath: IndexPath,
withSender sender: Any?)
```

IndexPath

Creating an Index Path Object

```
init(forRow:inSection:)
```

```
init(forItem:inSection:)
```

Getting the Section Index

```
section Property
```

Getting the Index of a Row or Item

```
row Property
```

```
item Property
```

기본코드

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return 10  
}
```

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
    let cell = UITableViewCell()  
    cell.textLabel?.text = "\(indexPath.row)번째 Row"  
    return cell  
}
```

TableViewCell 재사용

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return 10  
}
```

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
    var cell:UITableViewCell? = tableView.dequeueReusableCell(withIdentifier: "cell")  
    if cell == nil {  
        cell = UITableViewCell(style: .default, reuseIdentifier: "cell")  
    }  
    cell!.textLabel?.text = "\((indexPath.row)번째 Row"  
    return cell!  
}
```

TableViewCell 재사용

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return 10  
}
```

```
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
    let tmpCell = tableView.dequeueReusableCell(withIdentifier: "cell")  
    if let cell = tmpCell {  
        cell.textLabel?.text = "\(indexPath.row)번째 Row"  
        return cell  
    } else {  
        let cell = UITableViewCell(style: .default, reuseIdentifier: "cell")  
        cell.textLabel?.text = "\(indexPath.row)번째 Row"  
        return cell  
    }  
}
```

TableViewCell 재사용

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    return 10  
}  
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->  
UITableViewCell {  
    let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)  
    //storyboard에서 PrototypeCell의 갯수를 1개이상한후 identifier를 설정해주어야함  
  
    cell.textLabel?.text = "\((indexPath.row)번째 Row"  
    return cell  
}
```


배열내용 출력

```
@IBOutlet weak var tableView: UITableView!  
let arrbts = ["진", "슈가", "제이홉", "RM", "지민", "뷔", "정국"]  
override func viewDidLoad() {  
    super.viewDidLoad()  
    tableView.dataSource = self  
    tableView.delegate = self  
}  
func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) ->  
Int {  
    return arrbts.count  
}  
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->  
UITableViewCell {  
    let cell = tableView.dequeueReusableCell(withIdentifier: "cell",  
                                             for: indexPath)  
  
    cell.textLabel?.text = arrbts[indexPath.row]  
    return cell  
}
```