**UITableView分组举例**

import UIKit

class ViewController: UIViewController,UITableViewDelegate,UITableViewDataSource {

//建立一个二维数组

let cities = [["广州","深圳"],["南京","苏州","徐州"],["杭州","温州","义乌","绍兴"]]

//根据数组中元素的个数返回分组的组数

func numberOfSections(in tableView: UITableView) -> Int {

return cities.count

}

//返回每一个分组中的行数

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

return cities[section].count

}

//返回每一个cell显示的内容，列表每一行显示时都会被执行一次，用indexpath对象，可以得到当前是哪一组哪一行。可以从数组中取回相应的元素，显示到单元格上（cell.textLabel?.text）

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

//定义一个cell，withIdentifier: "cell"的意思是从缓存池中选取标识符为cell的单元格循环使用

let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)

//给cell里面显示的内容赋值，二维数组的下标分别对应组和行

cell.textLabel?.text = cities[indexPath.section][indexPath.row]

//返回cell

return cell

}

//给每一个分组加一个头部说明

func tableView(\_ tableView: UITableView, titleForHeaderInSection section: Int) -> String? {

switch section {

case 0:

return "广东省"

case 1:

return "江苏省"

default:

return "浙江省"

}

}

}

**UITableView简单举例**

import UIKit

//遵守两个协议

class ViewController: UIViewController, UITableViewDataSource,UITableViewDelegate {

//标题数组

var areas = ["徐州","济南","石家庄","武汉","长沙","三亚","杭州","合肥","桂林","广州"]

//图标名称数组

var areaImages = ["xuzhou","jinan","shijiazhuang","wuhan","changsha","sanya","hangzhou","hefei","guilin","guangzhou"]

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view, typically from a nib.

}

//返回一个Section（组）有多少行，默认只有一个组

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

return areas.count

}

//返回每一个cell显示的内容，列表每一行显示时都会被执行一次，用indexPath对象，可以得到当前是哪一行（indexPath.row）。可以从数组中取回相应的元素，显示到单元格上（cell.textLabel.text）

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

//定义一个cell，withIdentifier: "cell"意思是从缓存池中选取标识符为cell的单元格循环使用

let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)

//给cell里面显示的内容赋值，数组的下标对应每一行

cell.textLabel?.text = areas[indexPath.row]

//给cell添加图像

cell.imageView?.image = UIImage(named: areaImages[indexPath.row])

//将cell返回

return cell

}

}

**使用UITableViewController**

import UIKit

class AreaTableViewController: UITableViewController {

var areas = ["江苏徐州","山东济南","河北石家庄","湖北武汉","湖南长沙","海南三亚","浙江杭州","安徽合肥","广西桂林","广东广州"]

var areaImages = ["xuzhou","jinan","shijiazhuang","wuhan","changsha","sanya","hangzhou","hefei","guilin","guangzhou"]

override func viewDidLoad() {

super.viewDidLoad()

}

// MARK: - Table view data source

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

return areas.count

}

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

let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)

cell.textLabel?.text = areas[indexPath.row]

//通过读取数组元素得到对应图片名称，设置imageView上的图片

cell.imageView?.image = UIImage(named: areaImages[indexPath.row])

return cell

}

}