TIMERS

//to stop a timer

timer.invalidate()

//to link a timer to a specific function

Timer.scheduledTimer(timeInterval: 1, target: self, selector: #selector(ViewController.decreaseTimer), userInfo: nil, repeats: true)

TABLES

You need to use the UITableViewDelegate, UITableViewDataSource types in class

//returns the number of rows in the table

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

return 50

}

//returns what should be inside the table cell

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

let cell = UITableViewCell(style: UITableViewCellStyle.default, reuseIdentifier: "Cell")

cell.textLabel?.text = String(indexPath.row + 1)

return cell

}

//to reload a table when the user changes screen, not when the app loads

override func viewDidAppear(\_ animated: Bool)

//to delete a row from a table

func tableView(\_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath)

STORING DATA PERMANENTLY (USER DEFAULTS)

//this saves what the user typed in

UserDefaults.standard.set(textField.text, forKey: "number")

//this retrieves what the user typed in

UserDefaults.standard.object(forKey: "number")

CONTROLLING THE KEYBOARD

You need to use the UITextFieldDelegate type in class

//this removes the keyboard when user taps elsewhere in the app

override func touchesBegan(\_ touches: Set<UITouch>, with event: UIEvent?) {

self.view.endEditing(true)

}

//this removes the keyboard when user taps return in the app

func textFieldShouldReturn(\_ textField: UITextField) -> Bool {

textField.resignFirstResponder()

return true

}

DOWNLOADING WEB CONTENT

//this creates a url

let url = URL(string: "https://www.stackoverflow.com")

//this is the action of going to the webpage and retrieving info.

let task = URLSession.shared.dataTask(with: request as URLRequest) {

data, response, error in

//to uncrypt info.

let dataString = NSString(data: unwrappedData, encoding: String.Encoding.utf8.rawValue)

DispatchQueue.main.sync(execute:

//to actívate task

task.resume()