**mainVC.swift**

//

// mainVC.swift

// DemoSegue

//

// Created by apple on 2020/5/16.

// Copyright © 2020 AV Talk. All rights reserved.

//

**import** UIKit

**import** SQLite3

**class** mainVC: UIViewController,UIPickerViewDelegate ,UIPickerViewDataSource ,UITextFieldDelegate{

**var** SaveController :integer\_t = 0

**let** mode = ["不運動","每週輕鬆運動3~5天","每週強度中運動3~5天","每週高強度運動6~7天","每天訓練"]

**let** genderr = ["男","女"]

**var** SaveMode = "不運動"

**var** whatgender = "男"

**var** hotbase : Double = 1.0

**var** tdeenumber :Double = 0

**var** tdee :Double = 0

**@IBOutlet** **weak** **var** Picker: UIPickerView!

**@IBOutlet** **weak** **var** Save: UIButton!

**@IBOutlet** **weak** **var** Gender: UITextField!

**@IBOutlet** **weak** **var** Weight: UITextField!

**@IBOutlet** **weak** **var** Check: UITextView!

**@IBOutlet** **weak** **var** height: UITextField!

**var** db :SQLiteConnect? = **nil**

**let** sqliteURL: URL = {

**do** {

**return** **try** FileManager.default.url(for: .documentDirectory, in: .userDomainMask, appropriateFor: **nil**, create: **true**).appendingPathComponent("db.sqlite")

} **catch** {

fatalError("Error getting file URL from document directory.")

}

}()

**@IBAction** **func** jumptofirst(**\_** sender: UIButton)

{

**self**.performSegue(withIdentifier: "firstVC" , sender: **self**)

}

**@IBAction** **func** jumptothird(**\_** sender: UIButton) {

**self**.performSegue(withIdentifier: "thirdVC" , sender: **self**)

}

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

**self**.view?.endEditing(**true**)

}

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

**self**.Picker.dataSource = **self**

**self**.Picker.delegate = **self**

Weight.delegate = **self**

Gender.delegate = **self**

height.delegate = **self**

**let** sqlitePath = sqliteURL.path

db = SQLiteConnect(path: sqlitePath)

**if** **let** mydb = db {

**let** **\_** = mydb.createTable("BodySetting", columnsInfo: [

"id integer primary key autoincrement",

"gender text",

"mode text",

"weight double",

"age double",

"height double"])

**if** SaveController == 1 && Check.text == ""{

**guard** **let** text1 = Weight.text

**else**{

**return**

}

**guard** **let** text2 = Gender.text

**else**{

**return**

}

**guard** **let** text3 = height.text

**else**{

**return**

}

**guard** **let** WeightSetting = Double(text1)

**else**{

**return**

}

**guard** **let** AgeSetting = Double(text2)

**else**{

**return**

}

**guard** **let** heightSetting = Double(text3)

**else**{

**return**

}

**let** **\_** = mydb.insert("BodySetting",rowInfo: ["gender":"'\(whatgender)'","mode":"'\(SaveMode)'","weight":"'\(WeightSetting)'","age":"'\(AgeSetting)'","height":"'\(heightSetting)'"])

}

**if** SaveController == 1 && Check.text != "" {

**guard** **let** text1 = Weight.text

**else**{

**return**

}

**guard** **let** text2 = Gender.text

**else**{

**return**

}

**guard** **let** text3 = height.text

**else**{

**return**

}

**guard** **let** WeightSetting = Double(text1)

**else**{

**return**

}

**guard** **let** AgeSetting = Double(text2)

**else**{

**return**

}

**guard** **let** heightSetting = Double(text3)

**else**{

**return**

}

**let** **\_** = mydb.update(

"BodySetting",

cond: "id = 1",

rowInfo: ["gender":"'\(whatgender)'","mode":"'\(SaveMode)'","weight":"'\(WeightSetting)'","age":"'\(AgeSetting)'","height":"'\(heightSetting)'"])

}

**let** statement = mydb.fetch("BodySetting", cond: "1 == 1", order: **nil**)

**while** sqlite3\_step(statement) == SQLITE\_ROW{

**let** id = sqlite3\_column\_int(statement, 0)

**let** gender = String(cString: sqlite3\_column\_text(statement, 1))

**let** checkmode = String(cString: sqlite3\_column\_text(statement, 2))

**let** weight = sqlite3\_column\_double(statement, 3)

**let** age = sqlite3\_column\_double(statement, 4)

**let** checkheight = sqlite3\_column\_double(statement, 5)

**if** gender == "男"{

hotbase = (13.7\*weight)+(5\*checkheight)-(6.8\*age)+66

}

**if** gender == "女"{

hotbase = (9.6\*weight)+(1.8\*checkheight)-(4.7\*age)+655

}

**if** checkmode == "不運動"

{

tdeenumber = 1.2

}

**if** checkmode == "每週輕鬆運動3~5天"

{

tdeenumber = 1.375

}

**if** checkmode == "每週強度中運動3~5天"

{

tdeenumber = 1.55

}

**if** checkmode == "每週高強度運動6~7天"

{

tdeenumber = 1.725

}

**if** checkmode == "每天訓練"

{

tdeenumber = 1.9

}

tdee = ((hotbase \* tdeenumber)\*1000).rounded()/1000

Check.text = "\(id).性別：\(gender) \n體重：\(weight)\n運動習慣:\(checkmode)\n年齡:\(age)\n體重:\(checkheight)\n每日總消耗熱量:\(tdee)"

}

sqlite3\_finalize(statement)

}

}

**override** **func** didReceiveMemoryWarning() {

**super**.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

**func** numberOfComponents(in pickerView: UIPickerView) -> Int {

**return** 2

}

**func** pickerView(**\_** pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {

**if** component == 0 {

**return** mode.count

}

**return** genderr.count

}

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

**if** component == 0 {

**return** mode[row]

}

**return** genderr[row]

}

**func** pickerView(**\_** pickerView: UIPickerView, didSelectRow row: Int, inComponent component: Int) {

**if** component == 0 {

SaveMode = mode[row]

}

**else**{

whatgender = genderr[row]

}

}

**@IBAction** **func** unwind(for segue : UIStoryboardSegue)

{

}

**@IBAction** **func** SaveControll(**\_** sender: UIButton) {

SaveController = 1

viewDidLoad()

}

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

textField.resignFirstResponder()

**return** **true**

}

}

**firstVC.swift**

//

// firstVC.swift

// DemoSegue

//

// Created by apple on 2020/5/16.

// Copyright © 2020 AV Talk. All rights reserved.

//

**import** UIKit

**import** AVFoundation

**import** Vision

**var** food = ""

**var** foodgram = ""

**var** foodgramm:Double = 0

**class** firstVC: UIViewController, AVCaptureVideoDataOutputSampleBufferDelegate {

// create a label to hold the Pokemon name and confidence

**let** label: UILabel = {

**let** label = UILabel()

label.textColor = .black

label.translatesAutoresizingMaskIntoConstraints = **false**

label.text = "Label"

label.font = label.font.withSize(30)

**return** label

}()

**var** quantity = ""

**var** db :SQLiteConnect? = **nil**

**let** sqliteURL: URL = {

**do** {

**return** **try** FileManager.default.url(for: .documentDirectory, in: .userDomainMask, appropriateFor: **nil**, create: **true**).appendingPathComponent("db.sqlite")

} **catch** {

fatalError("Error getting file URL from document directory.")

}

}()

**override** **func** viewDidLoad() {

// call the parent function

**super**.viewDidLoad()

// establish the capture session and add the label

setupCaptureSession()

view.addSubview(label)

setupLabel()

}

**override** **func** didReceiveMemoryWarning() {

// call the parent function

**super**.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

**func** setupCaptureSession() {

// create a new capture session

**let** captureSession = AVCaptureSession()

// find the available cameras

**let** availableDevices = AVCaptureDevice.DiscoverySession(deviceTypes: [.builtInWideAngleCamera], mediaType: AVMediaType.video, position: .back).devices

**do** {

// select a camera

**if** **let** captureDevice = availableDevices.first {

captureSession.addInput(**try** AVCaptureDeviceInput(device: captureDevice))

}

} **catch** {

// print an error if the camera is not available

print(error.localizedDescription)

}

// setup the video output to the screen and add output to our capture session

**let** captureOutput = AVCaptureVideoDataOutput()

captureSession.addOutput(captureOutput)

**let** previewLayer = AVCaptureVideoPreviewLayer(session: captureSession)

previewLayer.frame = view.frame

view.layer.addSublayer(previewLayer)

// buffer the video and start the capture session

captureOutput.setSampleBufferDelegate(**self**, queue: DispatchQueue(label: "videoQueue"))

captureSession.startRunning()

}

**func** captureOutput(**\_** output: AVCaptureOutput, didOutput sampleBuffer: CMSampleBuffer, from connection: AVCaptureConnection)

{

// load our CoreML Food101 model

**guard** **let** model = **try**? VNCoreMLModel(for: Food101().model) **else** { **return** }

// run an inference with CoreML

**let** request = VNCoreMLRequest(model: model)

{ (finishedRequest, error) **in**

// grab the inference results

**guard** **let** results = finishedRequest.results **as**? [VNClassificationObservation] **else** { **return** }

// grab the highest confidence result

**guard** **let** Observation = results.first **else** { **return** }

// create the label text components

**let** predclass = "\(Observation.identifier)"

**let** predconfidence = String(format: "%.02f", Observation.confidence \* 100)

// set the label text

DispatchQueue.main.async(execute: {**self**.label.text = "\(predclass) \(predconfidence)%"

**if** Observation.confidence > 0.8 {

**if**(Observation.identifier == "tiramisu")

{

**self**.quantity = "\n一份100克"

}

**else** **if**(Observation.identifier == "ice\_cream")

{

**self**.quantity = "\n一球50克"

}

**else** **if**(Observation.identifier == "sushi")

{

**self**.quantity = "\n一個 40克"

}

**else** **if**(Observation.identifier == "foie\_gras")

{

**self**.quantity = "\n一份 100克"

}

**else** **if**(Observation.identifier == "cheesecake")

{

**self**.quantity = "\n一塊 100克"

}

**else** **if**(Observation.identifier == "hamburger")

{

**self**.quantity = "\n一個 180克"

}

**else** **if**(Observation.identifier == "fried\_rice")

{

**self**.quantity = "\n加飯 400克 正常量 300克"

// 880 660hot

}

**else** **if**(Observation.identifier == "french\_fries")

{

**self**.quantity = "\n 中份 120克 大份 140克"

// 880 660hot"

}

**else** **if**(Observation.identifier == "macarons")

{

**self**.quantity = "\n一個 24克"

//110hot

}

**else**

{

**self**.quantity = ""

}

**let** controller = UIAlertController(title: "辨識結果", message: Observation.identifier + **self**.quantity , preferredStyle: .alert)

controller.addTextField{(UITextField) -> Void **in**

UITextField.keyboardType = UIKeyboardType.phonePad

}

**let** okAction = UIAlertAction(title: "確定", style: .default,handler:

{**\_** **in**

**let** gram = controller.textFields?[0].text

foodgram = (gram!)

foodgramm = (foodgram **as** NSString).doubleValue

food = Observation.identifier

**self**.performSegue(withIdentifier: "seguesecondVC", sender: **nil**)

})

controller.addAction(okAction)

**let** cancelAction = UIAlertAction(title: "取消", style: .cancel,handler: **nil**)

controller.addAction(cancelAction)

**self**.present(controller, animated: **true**, completion: **nil**)

}

})

}

// create a Core Video pixel buffer which is an image buffer that holds pixels in main memory

// Applications generating frames, compressing or decompressing video, or using Core Image

// can all make use of Core Video pixel buffers

**guard** **let** pixelBuffer: CVPixelBuffer = CMSampleBufferGetImageBuffer(sampleBuffer) **else** { **return** }

// execute the request

**try**? VNImageRequestHandler(cvPixelBuffer: pixelBuffer, options: [:]).perform([request])

}

**func** setupLabel() {

// constrain the label in the center

label.centerXAnchor.constraint(equalTo: view.centerXAnchor).isActive = **true**

// constrain the the label to 50 pixels from the bottom

label.bottomAnchor.constraint(equalTo: view.bottomAnchor, constant: -75).isActive = **true**

}

**override** **func** prepare(for segue: UIStoryboardSegue, sender: **Any**?)

{

**let** now = Date()

**let** formatter = DateFormatter()

formatter.dateFormat = "yyyy-MM-dd"

**let** time = formatter.string(from:now)

**if** segue.identifier == "seguesecondVC"

{

**let** secondVC = segue.destination **as**! secondVC

//secondVC.receivedText = Textfield.text!

secondVC.receivedText = food

secondVC.datetime = time

secondVC.foodgramsecond = foodgramm

}

}

}

**secondVC.swift**

//

// secondVC.swift

// DemoSegue

//

// Created by apple on 2020/5/16.

// Copyright © 2020 AV Talk. All rights reserved.

//

**import** SQLite3

**import** UIKit

**struct** Foood: Codable

{

**let** hott: Double

**let** egg: Double

**let** fat: Double

**let** gdfat: Double

**let** bdfat: Double

**let** carbony: Double

**let** sugar: Double

**let** na: Double

**private** **enum** CodingKeys: String, CodingKey

{

**case** hott = "hot"

**case** egg

**case** fat

**case** gdfat

**case** bdfat

**case** carbony

**case** sugar

**case** na

}

}

**struct** Owner: Codable

{

**let** name: String

**var** fooods: [Foood]?

}

**class** secondVC: UIViewController

{

**var** SaveControll:integer\_t = 0

**var** hotcell:Double = 0

**var** eggcell:Double = 0

**var** fatcell:Double = 0

**var** gdfatcell:Double = 0

**var** bdfatcell:Double = 0

**var** carbonycell:Double = 0

**var** sugarcell:Double = 0

**var** nacell:Double = 0

**lazy** **var** tessst = ["食物名稱: \(receivedText)","熱量: \(hotcell)","蛋白質: \(eggcell)","脂肪: \(fatcell)","飽和脂肪: \(gdfatcell)","反式脂肪: \(bdfatcell)",

"碳水化合物: \(carbonycell)","糖: \(sugarcell)","鈉: \(nacell)","日期: \(datetime)"]

**var** db :SQLiteConnect? = **nil**

**let** sqliteURL: URL = {

**do** {

**return** **try** FileManager.default.url(for: .documentDirectory, in: .userDomainMask, appropriateFor: **nil**, create: **true**).appendingPathComponent("db.sqlite")

} **catch** {

fatalError("Error getting file URL from document directory.")

}

}()

//readtime and foodname

**var** receivedText:String = ""

**var** datetime = ""

**var** foodgramsecond:Double = 0

**@IBOutlet** **var** tableView: UITableView!

**@IBOutlet** **weak** **var** savedata: UIButton!

**override** **func** viewDidLoad()

{

**super**.viewDidLoad()

**let** foodOwnerJSONDATA = """

[

{

"name": "macarons",

"fooods":

[

{

"hot": 4.56,

"egg": 0.098,

"fat": 0.235,

"gdfat": 0,

"bdfat": 0,

"carbony": 0.503,

"sugar": 0.39,

"na": 0.0007

}

]

},

{

"name":"fried\_rice",

"fooods":

[

{

"hot": 2.21,

"egg": 0.05,

"fat": 0.085,

"gdfat": 0.025,

"bdfat": 0,

"carbony": 0.3,

"sugar": 0,

"na": 0.00245

}

]

},

{

"name":"hamburger",

"fooods":

[

{

"hot": 2.94,

"egg": 0.17,

"fat": 0.14,

"gdfat": 0.05,

"bdfat": 0.008,

"carbony": 0.24,

"sugar": 0.042,

"na": 0.00414

}

]

},

{

"name":"sushi",

"fooods":

[

{

"hot": 1.51,

"egg": 0.0429,

"fat": 0.0043,

"gdfat": 0.00091,

"bdfat": 0,

"carbony": 0.3173,

"sugar": 0.0571,

"na": 0.00558

}

]

},

{

"name": "ice\_cream",

"fooods":

[

{

"hot": 2.6,

"egg": 0.04,

"fat": 0.15,

"gdfat": 0.075,

"bdfat": 0,

"carbony": 0.272,

"sugar": 0.21,

"na": 0.00117

}

]

},

{

"name": "foie\_gras",

"fooods":

[

{

"hot": 4.62,

"egg": 0.11,

"fat": 0.44,

"gdfat": 0.14,

"bdfat": 0,

"carbony": 0.047,

"sugar": 0.22,

"na": 0.00697

}

]

},

{

"name": "tiramisu",

"fooods":

[

{

"hot": 3.28,

"egg": 0.044,

"fat": 0.251,

"gdfat": 0.163,

"bdfat": 0,

"carbony": 0.211,

"sugar": 0.106,

"na": 0.00105

}

]

},

{

"name": "cheesecake",

"fooods":

[

{

"hot": 3.21,

"egg": 0.06,

"fat": 0.23,

"gdfat": 0.1,

"bdfat": 0,

"carbony": 0.26,

"sugar": 0.22,

"na": 0.00438

}

]

},

{

"name": "dumplings",

"fooods":

[

{

"hot": 2.27,

"egg": 0.0659,

"fat": 0.12,

"gdfat": 0.02149,

"bdfat": 0.033,

"carbony": 0.2259,

"sugar": 0.0037,

"na": 0.00241

}

]

},

{

"name": "edamame",

"fooods":

[

{

"hot": 1.25,

"egg": 0.131,

"fat": 0.071,

"gdfat": 0.00985,

"bdfat": 0.004,

"carbony": 0.094,

"sugar": 0.022,

"na": 0.00014

}

]

},

{

"name": "filet\_mignon",

"fooods":

[

{

"hot": 1.87,

"egg": 0.222,

"fat": 0.102,

"gdfat": 0.022,

"bdfat": 0.004,

"carbony": 0.000,

"sugar": 0.000,

"na": 0.00035

}

]

},

{

"name": "french\_fries",

"fooods":

[

{

"hot": 3.2327,

"egg": 0.0381,

"fat": 0.1465,

"gdfat": 0.023,

"bdfat": 0.001,

"carbony": 0.43,

"sugar": 0.002,

"na": 0.00189

}

]

},

{

"name": "garlic\_bread",

"fooods":

[

{

"hot": 3.63,

"egg": 0.0852,

"fat": 0.1404,

"gdfat": 0.0271,

"bdfat": 0.001,

"carbony": 0.501,

"sugar": 0.0046,

"na": 0.0067

}

]

},

{

"name": "hot\_and\_sour\_soup",

"fooods":

[

{

"hot": 0.67,

"egg": 0.0619,

"fat": 0.0324,

"gdfat": 0.011,

"bdfat": 0.001,

"carbony": 0.02,

"sugar": 0.0065,

"na": 0.00637

}

]

},

{

"name": "hot\_dog",

"fooods":

[

{

"hot": 2.47,

"egg": 0.106,

"fat": 0.148,

"gdfat": 0.0521,

"bdfat": 0.001,

"carbony": 0.184,

"sugar": 0.0000,

"na": 0.00684

}

]

},

{

"name": "onion\_rings",

"fooods":

[

{

"hot": 4.3,

"egg": 0.039,

"fat": 0.25,

"gdfat": 0.0421,

"bdfat": 0.002,

"carbony": 0.44,

"sugar": 0.05,

"na": 0.00776

}

]

},

{

"name": "peking\_duck",

"fooods":

[

{

"hot": 4.36,

"egg": 0.166,

"fat": 0.384,

"gdfat": 0.000,

"bdfat": 0.000,

"carbony": 0.06,

"sugar": 0.00,

"na": 0.00083

}

]

},

{

"name": "pizza",

"fooods":

[

{

"hot": 2.70,

"egg": 0.12,

"fat": 0.087,

"gdfat": 0.028,

"bdfat": 0.000,

"carbony": 0.359,

"sugar": 0.037,

"na": 0.00437

}

]

},

{

"name": "pizza",

"fooods":

[

{

"hot": 2.70,

"egg": 0.12,

"fat": 0.087,

"gdfat": 0.028,

"bdfat": 0.000,

"carbony": 0.359,

"sugar": 0.037,

"na": 0.00437

}

]

},

{

"name": "pork\_chop",

"fooods":

[

{

"hot": 2.30,

"egg": 0.24,

"fat": 0.14,

"gdfat": 0.043,

"bdfat": 0.001,

"carbony": 0.0,

"sugar": 0.0,

"na": 0.0074

}

]

},

{

"name": "prime\_rib",

"fooods":

[

{

"hot": 1.33,

"egg": 0.21,

"fat": 0.05,

"gdfat": 0.019,

"bdfat": 0.003,

"carbony": 0.0,

"sugar": 0.0,

"na": 0.0055

}

]

},

{

"name": "takoyaki",

"fooods":

[

{

"hot": 1.27,

"egg": 0.07,

"fat": 0.055,

"gdfat": 0,

"bdfat": 0,

"carbony": 0.146,

"sugar": 0.0166,

"na": 0.0033

}

]

},

{

"name": "steak",

"fooods":

[

{

"hot": 1.33,

"egg": 0.21,

"fat": 0.05,

"gdfat": 0.019,

"bdfat": 0.003,

"carbony": 0,

"sugar": 0,

"na": 0.00055

}

]

},

{

"name": "apple\_pie",

"fooods":

[

{

"hot": 3.22,

"egg": 0.031,

"fat": 0.16,

"gdfat": 0.04,

"bdfat": 0.06,

"carbony": 0.44,

"sugar": 0.139,

"na": 0.00199

}

]

},

{

"name": "baby\_back\_ribs",

"fooods":

[

{

"hot": 2.77,

"egg": 0.15,

"fat": 0.23,

"gdfat": 0.08,

"bdfat": 0.002,

"carbony": 0,

"sugar": 0,

"na": 0.0008

}

]

},

{

"name": "bibimbap",

"fooods":

[

{

"hot": 1.465,

"egg": 0.051,

"fat": 0.035,

"gdfat": 0.009,

"bdfat": 0.04,

"carbony": 0.225,

"sugar": 0.015,

"na": 0.00108

}

]

},

{

"name": "bread\_pudding",

"fooods":

[

{

"hot": 6.4,

"egg": 0.053,

"fat": 0.047,

"gdfat": 0.016,

"bdfat": 0.016,

"carbony": 0.23,

"sugar": 0.147,

"na": 0.00236

}

]

},

{

"name": "caesar\_salad",

"fooods":

[

{

"hot": 0.44,

"egg": 0.032,

"fat": 0.02,

"gdfat": 0.012,

"bdfat": 0.001,

"carbony": 0.043,

"sugar": 0.021,

"na": 0.00083

}

]

},

{

"name": "chicken\_curry",

"fooods":

[

{

"hot": 4.15,

"egg": 0.13,

"fat": 0.12,

"gdfat": 0,

"bdfat": 0,

"carbony": 0.63,

"sugar": 0.024,

"na": 0.0004

}

]

}, {

"name": "chicken\_wings",

"fooods":

[

{

"hot": 2.17,

"egg": 0.188,

"fat": 0.142,

"gdfat": 0.039,

"bdfat": 0,

"carbony": 0.036,

"sugar": 0,

"na": 0.00509

}

]

}, {

"name": "cup\_cakes",

"fooods":

[

{

"hot": 3.05,

"egg": 0.043,

"fat": 0.037,

"gdfat": 0.011,

"bdfat": 0.018,

"carbony": 0.67,

"sugar": 0.255,

"na": 0.00413

}

]

}, {

"name": "club\_sandwich",

"fooods":

[

{

"hot": 2.6,

"egg": 0.21,

"fat": 0.096,

"gdfat": 0,

"bdfat": 0,

"carbony": 0.24,

"sugar": 0.055,

"na": 0.00735

}

]

}, {

"name": "creme\_brulee",

"fooods":

[

{

"hot": 2.79,

"egg": 0.046,

"fat": 0.231,

"gdfat": 0.143,

"bdfat": 0,

"carbony": 0.132,

"sugar": 0.103,

"na": 0.00041

}

]

},

{

"name": "donuts",

"fooods":

[

{

"hot": 4.52,

"egg": 0.049,

"fat": 0.25,

"gdfat": 0.13,

"bdfat": 0.11,

"carbony": 0.51,

"sugar": 0.27,

"na": 0.00326

}

]

}

]

""".data(using: .utf8)!

**let** foodownerdecoder = JSONDecoder()

**do** {

**let** foodOwners = **try** foodownerdecoder.decode([Owner].**self**, from: foodOwnerJSONDATA )

**var** yesno = 0

**for** owner **in** foodOwners

{

// if owner.name == receivedText

//print("\(owner.name)")

**if** owner.name == receivedText

{

**if** **let** fooods = owner.fooods, fooods.count > 0

{

**for** foood **in** fooods

{

hotcell = (foood.hott\*foodgramsecond\*1000).rounded()/1000

eggcell = (foood.egg\*foodgramsecond\*1000).rounded()/1000

fatcell = (foood.fat\*foodgramsecond\*1000).rounded()/1000

gdfatcell = (foood.gdfat\*foodgramsecond\*1000).rounded()/1000

bdfatcell = (foood.bdfat\*foodgramsecond\*1000).rounded()/1000

carbonycell = (foood.carbony\*foodgramsecond\*1000).rounded()/1000

sugarcell = (foood.sugar\*foodgramsecond\*1000).rounded()/1000

nacell = (foood.na\*foodgramsecond\*1000).rounded()/1000

// print("name \(owner.name) hot: \(foood.hott\*foodgramsecond) egg: \(foood.egg\*foodgramsecond) fat: \(foood.fat\*foodgramsecond) na: \(foood.na)")

yesno = 1

**break**

}

}

}

}

**if**(yesno == 0)

{

print("dont have \(receivedText) data")

}

}

**catch**{

print("Faaaaailed: \(error.localizedDescription)")

}

// 資料庫檔案的路徑

**let** sqlitePath = sqliteURL.path

// 印出儲存檔案的位置

print(sqlitePath)

// SQLite 資料庫

db = SQLiteConnect(path: sqlitePath)

**if** **let** mydb = db {

// create table

**let** **\_** = mydb.createTable("DateRecord", columnsInfo: [

"id integer primary key autoincrement",

"name text",

"hotdata double",

"eggdata double",

"fatdata double",

"gdfatdata double",

"bdfatdata double",

"carbonydata double",

"sugardata double",

"nadata double",

"date text"])

// insert

**if** SaveControll == 1 {

**let** **\_** = mydb.insert("DateRecord",rowInfo: ["name":"'\(receivedText)'","hotdata":"'\(hotcell)'","eggdata":"'\(eggcell)'","fatdata":"'\(fatcell)'","gdfatdata":"'\(gdfatcell)'","bdfatdata":"'\(bdfatcell)'","carbonydata":"'\(carbonycell)'","sugardata":"'\(sugarcell)'","nadata":"'\(nacell)'","date":"'\(datetime)'",])

//test average

/\* let \_ = mydb.insert("DateRecord",rowInfo: ["name":"'\(receivedText)'","hotdata":"'\(hotcell)'","eggdata":"'\(eggcell)'","fatdata":"'\(fatcell)'","gdfatdata":"'\(gdfatcell)'","bdfatdata":"'\(bdfatcell)'","carbonydata":"'\(carbonycell)'","sugardata":"'\(sugarcell)'","nadata":"'\(nacell)'","date":"'2020-08-13'",])

\*/ }

// select

**let** statement = mydb.fetch("DateRecord", cond: "1 == 1", order: **nil**)

**while** sqlite3\_step(statement) == SQLITE\_ROW{

**let** id = sqlite3\_column\_int(statement, 0)

**let** name = String(cString: sqlite3\_column\_text(statement, 1))

**let** hotdata = sqlite3\_column\_double(statement, 2)

**let** eggdata = sqlite3\_column\_double(statement, 3)

**let** fatdata = sqlite3\_column\_double(statement, 4)

**let** gdfatdata = sqlite3\_column\_double(statement, 5)

**let** bdfatdata = sqlite3\_column\_double(statement, 6)

**let** carbonydata = sqlite3\_column\_double(statement, 7)

**let** sugardata = sqlite3\_column\_double(statement, 8)

**let** nadata = sqlite3\_column\_double(statement, 9)

**let** date = String(cString: sqlite3\_column\_text(statement, 10))

print("\(id). \(name) 熱量：\(hotdata) 蛋白質:\(eggdata) 脂肪:\(fatdata) 飽和脂肪:\(gdfatdata) 反式脂肪:\(bdfatdata) 碳水化合物:\(carbonydata) 糖: \(sugardata) 鈉:\(nadata ) 日期:\(date)")

}

sqlite3\_finalize(statement)

**let** **\_** = mydb.delete("DateRecord", cond: "id = 32")

}

tableView.delegate = **self**

tableView.dataSource = **self**

}

**@IBAction** **func** SaveAction(**\_** sender: UIButton) {

**let** controller = UIAlertController(title: "確認儲存", message:"" , preferredStyle: .alert)

**let** okAction = UIAlertAction(title: "確定", style: .default,handler:

{**\_** **in**

**self**.SaveControll = 1

**self**.viewDidLoad()

**self**.performSegue(withIdentifier: "returnmain", sender: **self**)

})

controller.addAction(okAction)

**let** cancelAction = UIAlertAction(title: "取消", style: .cancel,handler: **nil**)

controller.addAction(cancelAction)

**self**.present(controller, animated: **true**, completion: **nil**)

}

}

**extension** secondVC: UITableViewDelegate

{

**func** tableView(**\_** tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

print("")

}

}

**extension** secondVC: UITableViewDataSource{

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

**return** 1

}

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

**return** tessst.count

}

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

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

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

**return** cell

}

}

**thirdVC.swift**

//

// thirdVC.swift

// DemoSegue

//

// Created by apple on 2020/5/16.

// Copyright © 2020 AV Talk. All rights reserved.

//

**import** UIKit

**import** SQLite3

**class** thirdVC: UIViewController {

//資料庫

**var** db :SQLiteConnect? = **nil**

**let** sqliteURL: URL = {

**do** {

**return** **try** FileManager.default.url(for: .documentDirectory, in: .userDomainMask, appropriateFor: **nil**, create: **true**).appendingPathComponent("db.sqlite")

} **catch** {

fatalError("Error getting file URL from document directory.")

}

}()

**var** times : String = ""

**@IBOutlet** **weak** **var** DatePicker: UIDatePicker!

**@IBOutlet** **weak** **var** TimeText: UITextField!

**@IBOutlet** **weak** **var** DateRecord: UITextView!

// @IBOutlet weak var advice: UITextView!

//today total

**var** hotmath :Double = 0

**var** todayhotmath :Double = 0

**var** eggmath :Double = 0

**var** todayeggmath :Double = 0

**var** fatmath :Double = 0

**var** todayfatmath :Double = 0

**var** carbonymath :Double = 0

**var** todaycarbonymath :Double = 0

**var** sugarmath :Double = 0

**var** todaysugarmath :Double = 0

**var** namath :Double = 0

**var** todaynamath :Double = 0

**var** a = 0

**var** AvgHot : Double = 0

**var** AvgEgg : Double = 0

**var** AvgFat : Double = 0

**var** AvgGDFat : Double = 0

**var** AvgBDfat : Double = 0

**var** AvgCar : Double = 0

**var** AvgSug : Double = 0

**var** AvgNa : Double = 0

**var** ControllAdvice = 0

**var** saveD1 : Double = 0

**var** saveD2 : Double = 0

**var** saveD3 : Double = 0

**var** saveD4 : Double = 0

**var** saveD5 : Double = 0

**var** saveD6 : Double = 0

**var** saveD7 : Double = 0

**var** showday = ""

**var** showhot = ""

**var** showegg = ""

**var** showfat = ""

**var** showbdfat = ""

**var** showgdfat = ""

**var** showcar = ""

**var** showsugar = ""

**var** showna = ""

**var** showadvice1 = ""

**var** showadvice2 = ""

**var** showadvice3 = ""

**var** showadvice4 = ""

**var** genderbmr :Double = 0

**var** tdeenumber :Double = 0

**var** lasttdee :Double = 0

**var** tdeecarbony :Double = 0

**var** tdeeegg :Double = 0

**var** tdeefat :Double = 0

**var** tdeena :Double = 0

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

**var** FoodAdvice1 = ""

**var** FoodAdvice2 = ""

**var** FoodAdvice3 = ""

**var** FoodAdvice4 = ""

**let** nowday = Date()

**let** nowday1 = nowday - 86400

**let** nowday2 = nowday - (86400\*2)

**let** nowday3 = nowday - (86400\*3)

**let** nowday4 = nowday - (86400\*4)

**let** nowday5 = nowday - (86400\*5)

**let** nowday6 = nowday - (86400\*6)

**let** nowday7 = nowday - (86400\*7)

**let** formatter = DateFormatter()

formatter.dateFormat = "yyyy-MM-dd"

**let** nowdaytime = formatter.string(from:nowday)

**let** nowdaytime1 = formatter.string(from:nowday1)

**let** nowdaytime2 = formatter.string(from:nowday2)

**let** nowdaytime3 = formatter.string(from:nowday3)

**let** nowdaytime4 = formatter.string(from:nowday4)

**let** nowdaytime5 = formatter.string(from:nowday5)

**let** nowdaytime6 = formatter.string(from:nowday6)

**let** nowdaytime7 = formatter.string(from:nowday7)

//不可被輸入

TimeText.isEnabled = **false**

DateRecord.text = ""

DateRecord.isEditable = **false**

// DateRecord.isEnabled = false

// 資料庫檔案的路徑

**let** sqlitePath = sqliteURL.path

db = SQLiteConnect(path: sqlitePath)

**if** **let** mydb = db {

// create table

**let** **\_** = mydb.createTable("DateRecord", columnsInfo: [

"id integer primary key autoincrement",

"name text",

"hotdata double",

"eggdata double",

"fatdata double",

"gdfatdata double",

"bdfatdata double",

"carbonydata double",

"sugardata double",

"nadata double",

"date text"])

**let** statement = mydb.fetch("BodySetting", cond: "1 == 1", order: **nil**)

**while** sqlite3\_step(statement) == SQLITE\_ROW{

// let id = sqlite3\_column\_int(statement, 0)

**let** gender = String(cString: sqlite3\_column\_text(statement, 1))

**let** checkmode = String(cString: sqlite3\_column\_text(statement, 2))

**let** weight = sqlite3\_column\_double(statement, 3)

**let** age = sqlite3\_column\_double(statement, 4)

**let** checkheight = sqlite3\_column\_double(statement, 5)

**if** checkmode == "不運動"

{

tdeenumber = 1.2

}

**if** checkmode == "每週輕鬆運動3~5天"

{

tdeenumber = 1.375

}

**if** checkmode == "每週強度中運動3~5天"

{

tdeenumber = 1.55

}

**if** checkmode == "每週高強度運動6~7天"

{

tdeenumber = 1.725

}

**if** checkmode == "每天訓練"

{

tdeenumber = 1.9

}

**if** gender == "男"{

genderbmr = (13.7\*weight)+(5\*checkheight)-(6.8\*age)+66

}

**if** gender == "女"{

genderbmr = (9.6\*weight)+(1.8\*checkheight)-(4.7\*age)+655

}

lasttdee = ((genderbmr \* tdeenumber)\*1000).rounded()/1000

//print(lasttdee)

//g

tdeecarbony = (lasttdee \* 0.56)/4

tdeeegg = (lasttdee \* 0.17)/4

tdeefat = (lasttdee \* 0.27)/9

tdeena = 6

}

sqlite3\_finalize(statement)

//搜尋

**let** statementD = mydb.fetch("DateRecord", cond: "1 == 1", order: **nil**)

**while** sqlite3\_step(statementD) == SQLITE\_ROW{

**let** id = sqlite3\_column\_int(statementD, 0)

**let** name = String(cString: sqlite3\_column\_text(statementD, 1))

**let** hotdata = sqlite3\_column\_double(statementD, 2)

**let** eggdata = sqlite3\_column\_double(statementD, 3)

**let** fatdata = sqlite3\_column\_double(statementD, 4)

**let** gdfatdata = sqlite3\_column\_double(statementD, 5)

**let** bdfatdata = sqlite3\_column\_double(statementD, 6)

**let** carbonydata = sqlite3\_column\_double(statementD, 7)

**let** sugardata = sqlite3\_column\_double(statementD, 8)

**let** nadata = sqlite3\_column\_double(statementD, 9)

**let** date = String(cString: sqlite3\_column\_text(statementD, 10))

**if** date == times {

**let** text1 : String? = DateRecord.text

**let** text2 : String? = "\(id). \(name) \n熱量：\(hotdata)大卡 \n蛋白質:\(eggdata)公克 \n脂肪:\(fatdata)公克 \n飽和脂肪:\(gdfatdata)公克 \n反式脂肪:\(bdfatdata)公克 \n碳水化合物:\(carbonydata)公克 \n糖: \(sugardata)公克 \n鈉:\(nadata )公克 \n日期:\(date)\n"

DateRecord.text = text1! + text2!

}

**if**(nowdaytime == date)

{

hotmath = hotmath + hotdata

eggmath = eggmath + eggdata

fatmath = fatmath + fatdata

carbonymath = carbonymath + carbonydata

sugarmath = sugarmath + sugardata

namath = namath + nadata

}

**if**(ControllAdvice == 0 ){

**if**(nowdaytime1 == date || nowdaytime2 == date || nowdaytime3 == date || nowdaytime4 == date || nowdaytime5 == date || nowdaytime6 == date || nowdaytime7 == date){

AvgHot = AvgHot + hotdata

AvgEgg = AvgEgg + eggdata

AvgFat = AvgFat + fatdata

AvgGDFat = AvgGDFat + gdfatdata

AvgBDfat = AvgBDfat + bdfatdata

AvgCar = AvgCar + carbonydata

AvgSug = AvgSug + sugardata

AvgNa = AvgNa + nadata

**if**(nowdaytime1 == date){

saveD1 = 1

}

**if**(nowdaytime2 == date){

saveD2 = 1

}

**if**(nowdaytime3 == date){

saveD3 = 1

}

**if**(nowdaytime4 == date){

saveD4 = 1

}

**if**(nowdaytime5 == date){

saveD5 = 1

}

**if**(nowdaytime6 == date){

saveD6 = 1

}

**if**(nowdaytime7 == date){

saveD7 = 1

}

}

}

}

sqlite3\_finalize(statementD)

**let** SaveAvgDay = saveD1 + saveD2 + saveD3 + saveD4 + saveD5 + saveD6 + saveD7

**if**(ControllAdvice == 0){

AvgHot = AvgHot / SaveAvgDay

AvgEgg = AvgEgg / SaveAvgDay

AvgFat = AvgFat / SaveAvgDay

AvgBDfat = AvgBDfat / SaveAvgDay

AvgGDFat = AvgGDFat / SaveAvgDay

AvgCar = AvgCar / SaveAvgDay

AvgSug = AvgSug / SaveAvgDay

AvgNa = AvgNa / SaveAvgDay

}

**if**( AvgHot > lasttdee + 100 ){

FoodAdvice1 = "攝取熱量嚴重過量\n"

}

**if**( AvgHot < lasttdee + 50 && AvgHot > lasttdee){

FoodAdvice1 = "攝取熱量輕微過量\n"

}

**if**( AvgHot <= lasttdee + 100 && AvgHot >= lasttdee + 50){

FoodAdvice1 = "攝取熱量中度過量\n"

}

**if**( AvgEgg > tdeeegg + 17/4){

FoodAdvice3 = "蛋白質嚴重過量\n"

}

**if**( AvgEgg < tdeeegg + 8.5/4 && AvgEgg > tdeeegg ){

FoodAdvice3 = "蛋白質輕微過量\n"

}

**if**( AvgEgg <= tdeeegg + 17/4 && AvgEgg >= tdeeegg + 8.5/4 ){

FoodAdvice3 = "蛋白質中度過量\n"

}

**if**( AvgCar > tdeecarbony + 56/4 ){

FoodAdvice2 = "碳水化合物嚴重過量\n"

}

**if**( AvgCar > tdeecarbony && AvgCar < tdeecarbony + 28/4 ){

FoodAdvice2 = "碳水化合物輕微過量\n"

}

**if**( AvgCar <= tdeecarbony + 56/4 && AvgCar >= tdeecarbony + 28/4 ){

FoodAdvice2 = "碳水化合物中度過量\n"

}

**if**( AvgFat > tdeefat + 27/4 ){

FoodAdvice4 = "脂肪嚴重過量\n"

}

**if**( AvgFat > tdeefat && AvgFat < tdeefat + 13.5/4 ){

FoodAdvice4 = "脂肪輕微過量\n"

}

**if**( AvgFat <= tdeefat + 27/4 && AvgFat >= tdeefat + 13.5/4 ){

FoodAdvice4 = "脂肪中度過量\n"

}

**let** SaveAvgDay1 = String( format: "%.0f" , SaveAvgDay)

**let** AvgHot1 = String( format: "%.0f" , AvgHot)

**let** AvgEgg1 = String( format: "%.1f" , AvgEgg)

**let** AvgFat1 = String( format: "%.1f" , AvgFat)

**let** AvgBDfat1 = String( format: "%.1f" , AvgBDfat)

**let** AvgGDFat1 = String( format: "%.1f" , AvgGDFat)

**let** AvgCar1 = String( format: "%.1f" , AvgCar)

**let** AvgSug1 = String( format: "%.1f" , AvgSug)

**let** AvgNa1 = String( format: "%.1f" , AvgNa)

**let** sumAdvice = FoodAdvice1 + FoodAdvice2 + FoodAdvice3 + FoodAdvice4

showday = SaveAvgDay1

showhot = AvgHot1

showegg = AvgEgg1

showfat = AvgFat1

showbdfat = AvgBDfat1

showgdfat = AvgGDFat1

showcar = AvgCar1

showsugar = AvgSug1

showna = AvgNa1

showadvice1 = FoodAdvice1

showadvice2 = FoodAdvice2

showadvice3 = FoodAdvice3

showadvice4 = FoodAdvice4

/\* advice.text = "紀錄天數:\(SaveAvgDay1)\n平均熱量:\(AvgHot1)大卡\n平均蛋白質:\(AvgEgg1)公克\n平均脂肪\(AvgFat1)公克\n平均飽和脂肪:\(AvgGDFat1)公克\n平均反式脂肪:\(AvgBDfat1)公克\n平均碳水化合物:\(AvgCar1)公克\n平均糖:\(AvgSug1)公克\n平均鈉\(AvgNa1)公克\n營養攝取量:\n\(sumAdvice)"

\*/

ControllAdvice = 1

}

**if**(a==0)

{

todayhotmath = (hotmath\*1000).rounded()/1000

todayeggmath = (eggmath\*1000).rounded()/1000

todayfatmath = (fatmath\*1000).rounded()/1000

todaycarbonymath = (carbonymath\*1000).rounded()/1000

todaysugarmath = (sugarmath\*1000).rounded()/1000

todaynamath = (namath\*1000).rounded()/1000

a = 1

}

}

**@IBAction** **func** averageAction(**\_** sender: UIButton)

{

**let** optionMenu = UIAlertController(title: "Today" , message: "", preferredStyle: .actionSheet)

**let** course1 = UIAlertAction(title: "紀錄天數: \(showday)", style: .default)

optionMenu.addAction(course1)

**let** course2 = UIAlertAction(title: "平均熱量: \(showhot)", style: .default)

optionMenu.addAction(course2)

**let** course3 = UIAlertAction(title: "平均蛋白質: \(showegg)", style: .default)

optionMenu.addAction(course3)

**let** course4 = UIAlertAction(title: "平均脂肪: \(showfat)", style: .default)

optionMenu.addAction(course4)

**let** course5 = UIAlertAction(title: "平均飽和脂肪: \(showgdfat)", style: .default)

optionMenu.addAction(course5)

**let** course6 = UIAlertAction(title: "平均反式脂肪: \(showbdfat)", style: .default)

optionMenu.addAction(course6)

**let** course7 = UIAlertAction(title: "平均碳水化合物: \(showcar)", style: .default)

optionMenu.addAction(course7)

**let** course8 = UIAlertAction(title: "平均糖: \(showsugar)", style: .default)

optionMenu.addAction(course8)

**let** course9 = UIAlertAction(title: "平均鈉: \(showna)", style: .default)

optionMenu.addAction(course9)

**if**(showadvice1 != "")

{

**let** course10 = UIAlertAction(title: "營養攝取量: \(showadvice1) ) ", style: .default)

course10.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course10)

}

**if**(showadvice2 != "")

{

**let** course11 = UIAlertAction(title: "營養攝取量: \(showadvice2) ) ", style: .default)

course11.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course11)

}

**if**(showadvice3 != "")

{

**let** course12 = UIAlertAction(title: "營養攝取量: \(showadvice3) ) ", style: .default)

course12.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course12)

}

**if**(showadvice4 != "")

{

**let** course13 = UIAlertAction(title: "營養攝取量: \(showadvice4) ) ", style: .default)

course13.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course13)

}

**let** cancel = UIAlertAction(title: "返回", style: .cancel)

optionMenu.addAction(cancel)

**self**.present(optionMenu, animated: **true**, completion: **nil**)

}

**@IBAction** **func** nowACtion(**\_** sender: UIButton)

{

**let** optionMenu = UIAlertController(title: "Today" , message: "", preferredStyle: .actionSheet)

//hot

**if**(todayhotmath < lasttdee\*0.6){

**let** course1 = UIAlertAction(title: "熱量: \(todayhotmath)", style: .default)

course1.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course1)

}

**else** **if** (lasttdee\*0.6<todayhotmath && lasttdee > todayhotmath){

**let** course1 = UIAlertAction(title: "熱量: \(todayhotmath)", style: .default)

course1.setValue(UIColor.orange, forKey: "titleTextColor")

optionMenu.addAction(course1)

}

**else**

{

**let** course1 = UIAlertAction(title: "熱量: \(todayhotmath)", style: .default)

course1.setValue(UIColor.red, forKey: "titleTextColor")

optionMenu.addAction(course1)

}

//egg

**if**(todayeggmath < tdeeegg\*0.6){

**let** course2 = UIAlertAction(title: "蛋白質: \(todayeggmath)", style: .default)

course2.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course2)

}

**else** **if** (tdeeegg\*0.6<todayeggmath && tdeeegg > todayeggmath){

**let** course2 = UIAlertAction(title: "蛋白質: \(todayeggmath)", style: .default)

course2.setValue(UIColor.orange, forKey: "titleTextColor")

optionMenu.addAction(course2)

}

**else**

{

**let** course2 = UIAlertAction(title: "蛋白質: \(todayeggmath)", style: .default)

course2.setValue(UIColor.red, forKey: "titleTextColor")

optionMenu.addAction(course2)

}

//fat

**if**(todayfatmath < tdeefat\*0.6){

**let** course3 = UIAlertAction(title: "脂肪: \(todayfatmath)", style: .default)

course3.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course3)

}

**else** **if** (tdeefat\*0.6<todayfatmath && tdeefat > todayfatmath){

**let** course3 = UIAlertAction(title: "脂肪: \(todayfatmath)", style: .default)

course3.setValue(UIColor.orange, forKey: "titleTextColor")

optionMenu.addAction(course3)

}

**else**

{

**let** course3 = UIAlertAction(title: "脂肪: \(todayfatmath)", style: .default)

course3.setValue(UIColor.red, forKey: "titleTextColor")

optionMenu.addAction(course3)

}

//carbony

**if**(todaycarbonymath < tdeecarbony\*0.6){

**let** course4 = UIAlertAction(title: "碳水化合物: \(todaycarbonymath)", style: .default)

course4.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course4)

}

**else** **if** (tdeecarbony\*0.6<todaycarbonymath && tdeecarbony > todaycarbonymath){

**let** course4 = UIAlertAction(title: "碳水化合物: \(todaycarbonymath)", style: .default)

course4.setValue(UIColor.orange, forKey: "titleTextColor")

optionMenu.addAction(course4)

}

**else**

{

**let** course4 = UIAlertAction(title: "碳水化合物: \(todaycarbonymath)", style: .default)

course4.setValue(UIColor.red, forKey: "titleTextColor")

optionMenu.addAction(course4)

}

//sugar

**let** course5 = UIAlertAction(title: "糖: \(todaysugarmath)", style: .default)

course5.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course5)

//na

**if**(todaynamath < tdeena\*0.6){

**let** course6 = UIAlertAction(title: "納: \(todaynamath)", style: .default)

course6.setValue(UIColor.black, forKey: "titleTextColor")

optionMenu.addAction(course6)

}

**else** **if** (tdeena\*0.6<todaynamath && tdeena > todaynamath){

**let** course6 = UIAlertAction(title: "納: \(todaynamath)", style: .default)

course6.setValue(UIColor.orange, forKey: "titleTextColor")

optionMenu.addAction(course6)

}

**else**

{

**let** course6 = UIAlertAction(title: "納: \(todaynamath)", style: .default)

course6.setValue(UIColor.red, forKey: "titleTextColor")

optionMenu.addAction(course6)

}

**let** cancel = UIAlertAction(title: "返回", style: .cancel)

optionMenu.addAction(cancel)

**self**.present(optionMenu, animated: **true**, completion: **nil**)

}

**@IBAction** **func** DateAction(**\_** sender: UIDatePicker) {

DateRecord.text = ""

**let** dateValue = DateFormatter()

dateValue.dateFormat = "yyyy-MM-dd" // 設定要顯示在Text Field的日期時間格式

dateValue.locale = Locale(identifier: "zh\_Hant\_TW")

times = dateValue.string(from: DatePicker.date)

TimeText.text = dateValue.string(from: DatePicker.date) // 更新Text Field的內容

viewDidLoad()

}

}