@hjd959

Swift Code

@hjd959

Into To Variables

Data Types

يوجد نوعان من أنواع البيانات وهم متغير وثابت

ماهي المتغيرات والثوابت ؟

المتغير قيمة يمكن تغييرها لاحقاً
ويرمز لها بالرمز var وهي اختصار الى كلمة Variable

اما الثابت وهي قيمة اوليه لايمكن تغييرها
وترمز بالرمز let واختصارها Constant

What types of data can variables and constants Store ?

Variables and constants can store numbers, words , letters,True/False ..

Type	Description
Int	0,2,-2,100, …
Float	3.14,5,-12.6, …
Double	3.14,5,-12.6, …
Bool	True / False
Character	"a", "C"
String	"Swift" , "Abdulwahab"

@hjd959

Creating Variables

Declaring Variables Of Different Types

طريقة كتابة المتغيرات ، يوجد طريقتان لكتابة المتغيرات

الطريقة الأولى

```
var number : int = 4
var centimeters : Float = 5.5
var pi : Double = 3.141592653
var letter : Character = "Z"
var nameOfCar : String = "BMW"
var rainingOutside : Bool = True
```

الطريقة الثانية

```
var number = 4
var centimeters  = 5.5
var pi  = 3.141592653
var letter  = "Z"
var nameOfCar  = "BMW"
var rainingOutside  = True
```

جميعها صحيحه

Changing a variables value

تستطيع تغيير القيمه اذا كانت من نوع var

```
var petsAge = 12
```

{ nameOfVariable} = {newValue}

We can doing this :

```
petsAge = 13
petsAge = 14
petsAge = 15
```

في هذا المثال استطعنا تغيير قيمة المتغير من نوع var الى اكثر من قيمه دون ان يصبح خطاء
ولاكن التغيير من قيمة رقميه int الى قيمة نصيه او أي قيمه أخرى خطا

if we change a varibles value to a value with a different type then our change fails

@hjd959

Example

```
var petsAge = 12


petsAge = 13
petsAge = 14
petsAge = "Old"
```

ما تستطيع تغيير نوع المتغير من رقم الى نص لازم التغير يكون من نفس النوع

This statement result in an error
Don't Change Type

@hjd959

Creating Constant

Declaring Constants Of Different Types

طريقة كتابة الثوابت ، يوجد طريقتان لكتابة الثوابت

الطريقة الأولى

```
let number : int = 4
let centimeters : Float = 5.5
let pi : Double = 3.141592653
let letter : Character = "Z"
let nameOfCar : String = "BMW"
let rainingOutside : Bool = True
```

الطريقة الثانية

```
let number = 4
let centimeters  = 5.5
let pi  = 3.141592653
let letter  = "Z"
let nameOfCar  = "BMW"
let rainingOutside  = True
```

جميعها صحيحه

You can’t Changing a Constants value

لاتستطيع تغيير القيمه اذا كانت من نوع ثابت

```
let petsAge = 12
```

We can’t doing this :

```
petsAge = 13
petsAge = 14
```

well get the following compiler Error

```
let encouragement = "you can do it "
    encouragement = " you cant do it "
```

well get the following compiler Error

constants no change value

النوع الثابت let لا تستطيع تغيير قيمته ابد

@hjd959

The command Print()

دالة الطباعه

Example 1

Print("Hello From The iOS Team !")

The result

Hello From The iOS Team

Example 2

```
var question = "Ready To Write Your First Line''
print(question )

var response = "yeah ! Im ready "
print( question ,response )
```

The result

```
Ready To Write Your First Line
yeah ! Im ready
```

question and response are example of variables which we declared with the keyword Var

Print()
Is a function which prints out whatever you place in between Parentheses

print() دالة الطباعه

تطبع أي امر جوى القوسين

Tuples

ماهو التوبلز : هو كتابة متغير داخله عدة قيم مختلفه

طريقة كتابته

```
var name = (القيم والمتغيرات)
```

Example 1

```
var car = ("Morsides" , "Black" , 2018)
car.0
car.1
car.2
```

في هذا المثال استخدمنا قيم مخلفه النوع والاستدعاء اسم المتغير ورقم الخانه

الطباعه

```
Morsides
Black
2018
```

Example 2

```
var car = (name: "morsides" ,color: "black" ,year: 2018)
car.name
car.color
car.year
```

نفس المثال السابق ولاكن اضفنا اسم لكل متغير والاستدعاء كان عن طريق الاسم

If Statements

- الشروط في جميع اللغات البرمجيه قاعده ثابته ومهمه ولكل شرط جواب في حال تحقق الشرط
- الجمل الشرطية تأخذ قيمه صح او خطا بمعنى اذا كان الجواب صح نفذ لي ما بداخل الكود واذا كان خطاء ينفذ الكود الذي يليه واذا ما وجده يخرج خارج الجمله الشرطية

طريقة كتابت الجمله الشرطية

```
var name value = true or false

if value name {
    code
}
```

Example 1

```
var hungry = false

If hungry {

print("go eat" )
}
```

هذا الكود إعطاء قيمه خطا لم يدخل داخل الجمله الشرطيه لان القيمه = falseخطاء لذالك ما نفذ شي

```
// no print any thing
```

Example 2

```
var wantTeddybear = false
var havemore = true

if wantTeddybear &&  havemore

{
print("buy")
}
```

وهذا اعطا قيمه صح وقيمه خطا والمطلوب من الكود تنفيذ القيمتان معاً
لم يدخل داخل الجمله الشرطيه لان القيم مختلفات

```
// no print any thing
```

استخدمنا في المثال السابق && هذه الرموز تعني (و) بمعنى (and) أي لازم تكون جميع الشروط صحيحه لاجل

@hjd959

تنفيذ الجمله الشرطيه

Example 3

```
var wantTeddybear = false
var havemore = true

if wantTeddybear ||  havemore

{
print("buy")
}
```

الان الطباعه buy

لأننا استخدمنا الشرط || (أو) بمعنى (or)
هنا يتأكد ايهما صح بمعنى أي متغير من المتغيران صح يدخل داخل كود الجمله الشرطية

Example 4

```
 var rining = true

if rining
{
Print("watch Tv" )
}
```

هذا الكود القيمه تساوي صح لذالك نفذ بما داخل الكود
watch Tv وهو طباعه
// print watch Tv

If- Else Statements

في حال وجود اكثر من شرط مراد تنفيذه سنستخدم الأمر if else

If Else طريقة كتابة
Var name value = true or false

```
if value name {
    code
} else  if value name {
    code
} else {
    code
}
```

Example 1

```
var breakfast = true
var lunch = false
var dinner = false

if breakfast {
    print("Good morning!")
} else  if lunch {
    print("Good afternoon!")
} else if dinner {
    print("Good evening!")
} else {
    print("Hello!")
}
```

الطباعه تكون
Good morning!

في هذا المثال وجدنا ثلاث متغيرات

```
breakfast = true
lunch = false
dinner = false
```

وجد اول متغير يساوي صح لذالك نفذ اول كود وهو امر طباعة

Example 2

```
var breakfast = true
var lunch = true
var dinner = true

if breakfast {
    print("Good morning!")
} else  if lunch {
    print("Good afternoon!")
} else if dinner {
    print("Good evening!")
} else {
    print("Hello!")
}
```

الطباعه تكون

```
Good morning!
Good afternoon!
Good evening!
```

نفذ جميع المتغيرات لان جميها تساوي قيمه صح

@hjd959

If- Else – if Statements

معنا else عكس الشرط تماماً يعني اذا كان الشرط غير ذالك لا تخرج من الكود نفذ القيمه التي جوى else ومن ثم اخرج

الطريقة لكتابة if else if

Var name value = true or false

```
if value name {
    code
} else  if value name {
    code
} else {
    code
}
```

Else الاخيره معناها عكس الامر تماماً بمعنا نفذ الشرط اذا كانت القيمه خطا

Example 1

```
let hungry = false
let vegetarian = false

if hungry {
    if vegetarian {
        print("Let's eat!")
    } else {
        print("Let's eat steak!")
    }
}
```

الطباعه تكون

```
Let's eat steak!
```

لان جميع الشرطين خطا

هذا نفس المثال السابق ولاكن بصيغه مختلفه

```
if hungry && vegetarian {
    print("Let's eat!")
} else if hungry && !vegetarian {
    print("Let's eat steak!")
```

Switch

نفس القاعده السابقه if ولاكن بطريقة مختلفه ومرتبه والأداء واحد

طريقة كتابة دالة switch

```
switch nameOfSwitch {
case 1: code
case 2: code
...
...
...

default: code
}
```

Example 1

```
var month = 2

switch month {
case 1: print("January")
case 2: print("February")
default: print("Unknown month")
}

print(month)
```

الطباعه

```
February
```

```
let meal = "breakfast"
switch meal {
    case "breakfast":
        print("Good morning!")
    case "lunch":
        print("Good afternoon!")
    case "dinner":
        print("Good evening!")
    default:
        print("Hello!")
}
```

الطباعة

```
Good morning!
```

مثال درجات الطلاب على if و switch

```
var dgOfStudent:Int = 76

if dgOfStudent >= 95 {
    print("A+")
} else if dgOfStudent >= 90 {
    print("A")
}else if dgOfStudent >= 85 {
    print("B+")
}else if dgOfStudent >= 80 {
    print("B")
}else if dgOfStudent >= 75 {
    print("C+")
}else if dgOfStudent >= 70 {
    print("C")
}else if dgOfStudent >= 65 {
    print("D+")
}else if dgOfStudent >= 60 {
    print("D")
}else   {
    print("F")
}

switch dgOfStudent {

case 95...100 : print("A+")
case 90...95 : print("A")
case 85...90 : print("B+")
case 80...85 : print("B")
case 75...80 : print("C+")
case 70...75 : print("C")
case 65...70 : print("D+")
case 60...65 : print("D")
default: print("F")

}
```

ملاحظة الثلاث نقاط تعني(من) العدد (الى) العدد

من ٩٥ الى ١٠٠

@hjd959

Loops while

اللوب بختصار حلقة تكرار
هي عبارة عن تكرار امر ما أكثر من مره دون الحاجه الى كتابة الكود مجدداً

طريقة كتابة While

```
While (condition) {
 cod
}

var I = 0

while(I<10) {
    print(I)
    I += 1

}
```

معنى هذا الامر اذا كان الشرط اقل من ١٠ نفذ لي الامر

FOR

الفور شبيه while

طريقة كتابة for

```
for ( condition ) {

    code

}
```

Example 1

```
for i in 1...10 {
    print(i)
}
```

هذا المثال عرفنا متغير من نوع String وهو i واعطينا قيمه ابتدائيه من واحد الى ١٠ والامر اطبع i الطباعه

```
1
2
3
4
5
6
7
8
9
10
```

الثلاث نقاط تدل على (العدد الى العدد) بختصار اذا اردت كتابه من العدد ٥ الى العدد ٢٠ تكتب بهذا الشكل 5...20

Example 2

```
for _ in 1...10 {

    print("Hello")
}
```

```
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
```

في هذا المثال ما عرفنا متغير المراد من هذا الامر طباعة Hello عشر مرات
معناه اطبع لي المتغير الذي داخل جملة print

Example 3

```
let word = "Mississippi"
for c in word {
    print(c)
}
```

الامر هذا اطبع كل حرف لحاله
الطباعه تكون

```
M
i
s
s
i
s
s
i
p
p
i
```

Array

المصفوفه هي عبارة عن متغير من نوع مصفوفه يحوي بداخله مجموعه من القيم ولاكن يشترط ان تكون هذه القيم من النوع نفسه

طريقة كتابة مصفوفه ،، يوجد طريقتان لكتابة المصفوفه

```
1  var numbers = Array<Double>()
2  var moreNumbers = [Double]()
```

كذالك الطريقتان أعلاه تعتبر مصفوفه فارغه

```
numbers.count
```

المقصود بها عدد العناصر المتواجده داخل المصفوفه (الطباعه صفر)

```
numbers.append(10.5)
```

المقصود بها اضف الرقم داخل المصفوفه

```
numbers.insert(12.3, at: 0)
```

المقصود بها اضف الرقم في المكان رقم صفر أي الخانه الأول من المصفوفه

```
numbers.insert(7.3, at: 1)
```

الطباعه

```
[12.3, 7.3, 10.5]
```

الان المفهوم من append اضافه في اخر خانه و insert ادخل العنصر في المكان المحدد

```
numbers.remove(at: 1)
```

دالة الحذف مع تحديد العنصر المراد حذفه

Example 1

```
let intArray = [7, 21, 25, 13, 1]
var sum = 0
for value in intArray {
    sum += value
}
sum
```

في هذا المثال اضفنا مصفوفه عرفنا متغير واعطنا قيمة المتغير صفر
انشأنا دالة for مهمتها تجمع الاعداد داخل المصفوفه
الطباعه 67

القاموس في المصفوفات Dictionary
نوع من أنواع تخزين القيم بقيمة أخرى فعلى سبيل المثال نستطيع تخزين كلمة انجليزية بمعناها العربي

Example 2

```
var dic = ["yes" :" نعم ", "no" : "لا"]

dic["yes"]
```

استدعينا الكلمه yes المفترض الطباعه الان كلمة نعم

Example 3

```
var phoneNumbers = ["Abdulwahab": 1234, "Abdullah": 05555,
"wahbe": 99999]

phoneNumbers["Abdulwahab"]
```

الطباعه الان 1234

إضافة عنصر جديد

```
phoneNumbers["Jamal"] = 1111
```

الان شكل المصفوفه بعد الاضافه

```
["Abdulwahab": 1234, "Abdullah": 05555, "wahbe": 99999,
"Jamal" : 1111 ]
```

كيف يتم حذف قيمة معينه

```
phoneNumbers.removeValue(forKey: "Abdulwahab")
```

الان المصفوفه بهذا الشكل

```
[ "Abdullah": 05555, "wahbe": 99999, "Jamal" : 1111 ]
```

@hjd959

Function

ماهي الدالة : الدالة عباره عن مجموعة أوامر بداخلها لها وظيفه محدده ولها اسم ،، يمكن استدعاءها من خلال مناداتها باسمها دون الحاجه لكتابة الاكواد الموجوده داخلها مره أخرى

كتابة الفنكشن بالشكل هذا

In Swift, functions are defined using the following syntax:

```
func nameOfFunction() {
    // code
}
```

Example 1

Defining the “sayHello” function
مثال فنكشن باسم
SayHello

```
func sayHello() {
    print("Hello")
}
```

Calling “ sayHello”
طريقة استدعاء الفنكشن فقط بأسمها و هي تنفذ ما بداخلها

```
sayHello()
```

الطباعه

```
Hello
```

Example 2

```
var n1 = 10
var n2 = 20

func nameOfFunction() {

    print("\(n1 * n2)")
}
nameOfFunction()
```

الطباعة 200

@hjd959

Function with Parameters

هي عباره عن المتغيرات التي تستقبلها الدالة ان كانت الداله بحاجه الى متغير او اكثر من متغير

Example 1

```
func sayHelloToStudent(student: String) {
    print("Hello \(student)")
}

sayHelloToStudent(student: "Abdulwahab")
```

الطباعة

Hello Abdulwahab

لا حظ استدعينا الداله وكتبنا المتغير اثناء الاستدعاء

Function Scope

اكثر من راجع

Example :

```
func averageScore(firstScore: Double, secondScore: Double,
thirdScore: Double) {
    let totalScore = firstScore + secondScore + thirdScore
    print(totalScore / 3)
}

averageScore(firstScore: 7.7, secondScore: 6.7, thirdScore:
8.6)
```

الـطبـاعـة الان

7.66666

Omitting External Parameter Names

معناه اختصار

Example 1

```
func aver(first:Double ,Second: Double){

    print("Hello \(first) \(Second)")
}

aver(first: 30.5, Second: 40.5)
```

Example 2

```
func aver(_ first:Double ,_ Second: Double){

    print("Hello \(first) \(Second)")
}

aver(30.5,40.5)
```

الفرق بين المثال الأول والثاني هو إضافة(_)هذه الشرطه قبل اسم المتغير اذا اضفتها عند الاستدعاء لا تحتاج ان تكتب اسم المتغير فقط اضف القيمه

function that returns a value

هذا دالة فنكشن ولاكن لها قيمه راجعه

Example

```
func calculateTip(priceOfMeal: Double) -> Double {
    return priceOfMeal * 0.15
}
```

```
-> Value  اذا اضفت هذه الجمله لابد من راجع
```

أي لازم تكتب قبل اغلاق الفنكشن كلمة return
هذه الكلمه تعني ارجع وهي التي تدل على القيم المراد إرجاعها للمستخدم

```
return value
```

@hjd959

Example with out Function : مثال على كود بدون فنكشن

```
var name = "Abdulwahab"
var age = 19
var onGuestList = true
var knowsTheOwner = true

if onGuestList && age >= 21 {
    print("\(name), come party with us!")
} else if knowsTheOwner {
    print("\(name), we'll have to take you to the owner.")
} else {
    print("Sorry, \(name), maybe you can go play Bingo with the Android team.")
}

name = "Gabrielle"
age = 29
onGuestList = true
knowsTheOwner = true

if onGuestList && age >= 21 {
    print("\(name), come party with us!")
} else if knowsTheOwner {
    print("\(name), we'll have to take you to the owner.")
} else {
    print("Sorry, \(name), maybe you can go play Bingo with the Android team.")
}

name = "Chris"
age = 32
onGuestList = false
knowsTheOwner = false

if onGuestList && age >= 21 {
    print("\(name), come party with us!")
} else if knowsTheOwner {
    print("\(name), we'll have to take you to the owner.")
} else {
    print("Sorry, \(name), maybe you can go play Bingo with the Android team.")
}
```

لاحظ كل مره كتبنا المتغيرات
الان نفس المثال ولاكن بستخدام الفنكشن

```
func clup(name : String , age : Int ,onGuestList: Bool ,knowsTheOwner:Bool ) {

    if onGuestList && age >= 21 {
        print("\(name), come party with us!")
    } else if knowsTheOwner {
        print("\(name), we'll have to take you to the owner.")
    } else {
        print("Sorry, \(name), maybe you can go play Bingo with the Android team.")
    }
}
```

اثناء الاستدعاء اضف القيم التي تريدها

```
clup(name: "Abdulwahab", age: 19, onGuestList: true, knowsTheOwner: true)
clup(name: "Gabrielle", age: 29, onGuestList: true, knowsTheOwner: true)
clup(name: "Chris", age: 32, onGuestList: false, knowsTheOwner: false)
```

هذي طريقة كتابة الفنكشن (لاحظ الفرق)

@hjd959

Enumes(Enumeration)

هو عبارة عن نوع من القيم تكون لها علاقه مع بعضها في شكل مجموعه ،، كما يستخدم ايضاً في عملية الترتيب للمعلومات

طريقة كتابته

```
enum nameOfenum {
    case name
    case name
    case name
}
```

او بالشكل هذا

```
enum nameOfenum {
    case name, name, name, name, name
}
```

Example

بدون قيم

```
enum PrimaryColor {
    case red
    case blue
    case yellow
}
```

مع قيم

```
enum AmericanLeagueWest: String {
    case athletics = "Oakland"
    case astros = "Houston"
    case angels = "Los Angeles"
    case mariners = "Seattle"
    case rangers = "Arlington"
}
```

@hjd959

Struct

البناء او دالة البنى هو عباره عن قالب بداخلها تحوي على خصائص وافعال ويمكنك استدعائه متى ما شئت

طريقة كتابته :

```
struct NameOfStruct {

methods And properties

}
```

Example

```
struct Student {
    let name: String
    var age: Int
    var school: String
}
```

بالشكل هذا تعرف لك داله وكل مره تستدعيه

```
var st = Student(name: "Abdulwahab", age: 24, school:
"University of Hail")

var ay = Student(name: "As", age: 19, school: "Udacity")
```

بالشكل هذا الاستدعاء

طريقة عرض الاسم

St.name
Abdulwahab يطبع لك

تغيير القيمه

St.age = 25

@hjd959

Classes

الكلاس هو عبارة عن صندوق بداخله مجموعة من الصفات والدوال متعددة المهام أي انه لا يشترط كل داله تعمل ذات العمل
الفرق بين الكلاس و دالة البناء ستركت هو الكلاس شمولية عمليات التعدد والوراثة وكثير من هذه الأمور لا يحويها دالة البناء ستركت ،، ستركت في الاستخدامات البسيطه

طريقة كتابة الكلاس :

```
class nameOfClass {

properties and methods

}
```

الان مثال عملنا كلاس باسم اب وله متغيرات ومن ثم عملنا داخل كلاس الاب فنكشن ،، بعد ذالك عملنا كلاس باسم ابن يرث من الاب و له متغير واحد دخل كلاس الابن استدعينا الفنكشن الموجوده عند الاب وعدلنا عليها ،، بعد ذالك عملنا متغيرات خارج الكلاسات واستدعيناهم وعدلنا على المتغيرات

Example

```
class Father {
    let name : String
    let lastName : String
    let colorEye : String
    var treasure : Double
```

يلزم كتابته اذا كانت القيم فارغه init

```
init(name : String , lastName : String , colorEye : String
,treasure : Double ) {
        self.name = name
        self.lastName = lastName
        self.colorEye = colorEye
        self.treasure = treasure

    }


    func Addname() {

        print("Hello \(name) \(lastName)")

    }


}
```

```
class Son : Father {

    var Myage : Int

    init(name: String, lastName: String, colorEye: String,
treasure: Double , Myage : Int) {
        self.Myage = Myage
        super.init(name: name, lastName: lastName, colorEye:
colorEye, treasure: treasure)

    }
    // تعديل على الفنكشن

    override func Addname() {
        print("Hello \(name) \(lastName) And my age is
\(Myage)")
    }

    }

let sonAb = Son(name: "Abdulwahab", lastName: "Alenezi",
colorEye: "black", treasure: 200.0, Myage: 25)

let fatherAb = Father(name: "Abdullah", lastName: "Alenezi",
colorEye: "black", treasure: 10000000.0)



fatherAb.Addname()
sonAb.Addname()
```

الان الطباعه

```
Hello Abdullah Alenezi
Hello Abdulwahab Alenezi And my Age is 25
```

@hjd959

Protocol

هي فنكشن ولاكن بدون تعريف ويقدر أي شخص يورثهم ويعمل لهم تعريف جديد هذا المفهوم المبدئي

طريقة كتابة البروتوكول

```
protocol nameOfprotocol {
    func functionName()

}
```

Example 1

```
protocol abdulwahab {
    func sayHello()
}
```

عرفنا بروتوكول باسم abdulwhab ودخال البروتوكول فنكشن باسم sayHello

```
class  TestPro : abdulwahab {

    var name : String

    init(name :String) {

        self.name = name
    }

    func sayHello() {

        print(" Hello \(name)")

    }
}
```

عملت كلاس باسم TestPro وهذا الكلاس يرث البروتوكول (أي فنكشن داخل البروتوكول لازم تكتب داخل الكلاس) ،،
اضفنا متغير من نوع String باسم name اضفنا الفنكشن الموروثه من البروتوكول وعدلنا عليها

```
var obj1 = TestPro.init(name:"ABDULWAHAB ABDULLH")
```

الان اضفنا متغير واعطيناه اسم واستدعينا المتغير الموجود داخل كلاس TestPro واعطيناه قيمه

```
obj1.sayHello()
```

الان الطباعه

```
Hello ABDULWAHAB ABDULLH
```

Example 2

نفس المثال السابق ولاكن بأضافة اكثر من كلاس

```
protocol abdulwahab {
    func sayHello()

}

class  TestPro : abdulwahab {

    var name : String
    init(name :String) {
        self.name = name
    }
    func sayHello() {
        print(" Hello \(name)")
    }
}

class Abdullah: abdulwahab {

        var titel = "Eng Abdulwahab"

        func sayHello() {

            print("Hello \(titel)")
        }
    }


var obj1 = TestPro.init(name:"ABDULWAHAB ABDULLH")

obj1.sayHello()

var obj2 = Abdullah.init()
```

هنا ما عرفنا المتغير لانه اساساً معطى قيمه من داخل الكلاس

```
obj2.sayHello()
```

الطباعه

```
Hello ABDULWAHAB ABDULLH
Hello Eng Abdulwahab
```

Extensions

مثال عندي كلاس تبي تورث ١٠ كلاسات وتبي تعمل ٢٠ بروتوكول صعبه كتابتهم مع بعض لذالك اضافو extensions لترتيب الكود

```
protocol abdulwahab {
    func sayHello()
}
protocol abdulwahab1 {
    func sayHello1()
}
protocol abdulwahab2 {
    func sayHello2()
}
protocol abdulwahab3 {
    func sayHello3()
}

class  TestPro : abdulwahab ,abdulwahab1 , abdulwahab2{
   مثال بدال ما اصفهم عند الكلاس بالشكل هذا

    var name = "Hello"
    func sayHello() {
        print("\(name) 0")
    }
    func sayHello1() {
        print("\(name) 1")
    }
    func sayHello2() {
        print("\(name) 2")
    }

       }

extension TestPro : abdulwahab3 {
     اكتب اكس تن شن وخلاص ارتب للكود
    func sayHello3() {
        print("\(name) 3")
    }
}
```

خاتمه

في الواقع لا توجد خاتمه .. فالكل نهاية امتداد يبدأ به كل شي ..

ببساطه هذا ملخص لبرمجة تطبيقات الأيفون لغة سويفت
وأتمنى أن يكون قد شمل جميع الأفكار والعناصر الأساسية
متمنياً من الله عز وجل أن أكون قد قمت بتقديم شيء نافع، وأسأل الله أن يوفقنا وأياكم وصلى الله وسلم على نبينا محمد

للتواصل

Email : Eng.abdulwahab7@gmail.com

Twitter : @hjd959