iOS行動程式基礎開發上架

同學，歡迎你參加本課程

- 請關閉你的FB 、Line等溝通工具，以免影響你上課。
- 考量頻寬、雜音，請預設關閉攝影機、麥克風，若有需要再打開。
- 隨時準備好，老師會呼叫你的名字進行互動，鼓勵用麥克風提問。
- 如果有緊急事情，你必需離開線上教室，請用聊天室私訊給老師，以免老師癡癡呼喚你的名字。
- 軟體安裝請在上課前安裝完成，未完成的同學，請盡快進行安裝。

課程檔案下載


巨匠電腦線上真人
開課查詢
免費體驗專區
課程總覽
專業師資
學員專區
講師專區
最新消息
您好!
登出
點數卡產品兌換
APCS檢測專區
公告專區
我的課表
IT真人課程劃位
電腦分校課程劃位
外語真人課程劃
美語分校課程劃
取消劃位
課程檔案下載
上課權益查詢
教學平台測試
學習諮詢
常見問題
個資維護
忘記密碼
登出
課程檔案下載
程式語言好難學?
那是因為
你還沒學過Python!
線上老師LIVE直播教學．搶先看
巨匠電腦真人課程

ZOOM 學員操作說明

5 查看選項/共同註記/筆（連連看）

2 共享螢幕（指導演練；點評作品）
老師須先停止共享螢幕
才能請學生共享螢幕

1 聊天

4 解除靜音

3 與會者/舉手

本課程各堂教學主題

swift基礎6堂課

- 第一堂：swift基本概念
- 第二堂：基本運算子、字串和字元
- 第三堂：集合物件
- 第四堂：流程控制
- 第五堂：函式和閉鎖
- 第六堂：列舉

本課程各堂教學主題

iOS行動程式基礎開發上架20堂課

- 第一堂：使用swift建立第一個App
- 第二堂：AutoLayout
- 第三堂：使用Stack Views設計UI
- 第四堂：建立以表格為基礎的App
- 第五堂：使用原型儲存格建立自訂的TableView
- 第六堂：用UIAlertController和使用者互動
- 第七堂：儲存格的刪除和自訂功能按鈕

本課程各堂教學主題

iOS行動程式基礎開發上架20堂課

- 第八堂：使用導覽控制
- 第九堂：自訂細節頁面
- 第十堂：自動調整高度的儲存格
- 第十一堂：使用地圖
- 第十二堂：展示圖片控制項
- 第十三堂：使用CoreData
- 第十四堂：搜尋控制項

本課程各堂教學主題

iOS行動程式基礎開發上架20堂課

- 第十五堂：TabBarController
- 第十六堂：內建瀏覽器
- 第十七堂：自多國語言
- 第十八堂：使用實機測試
- 第十九堂：上架說明1
- 第二十堂：上架說明2

巨匠線上真人

iOS行動程式基礎開發上架

第一堂：swift基本概念

本堂教學重點

1. 常數和變數
2. 資料類型註解
3. 輸出常數和變數
4. 註解
5. 整數
   - 整數的範圍
   - Int
   - UInt
6. 浮點數
7. 資料型別安全和型別推測
8. 數值表示法
9. 數值類型的轉換
   - 整數類型轉換
   - 整數和浮點數間的轉換
10. 資料類型的小名
11. 布林值
12. Tuple
13. Optionals
14. 錯誤處理

1常數和變數

```
let maximumNumberOfLoginAttempts = 10
var currentLoginAttempt = 0
```

```
var x = 0.0, y = 0.0, z = 0.0
```

資料類型註解

```
var welcomeMessage: String

welcomeMessage = “Hello"
```

輸出常數和變數

```
let π = 3.14159
let 你好 = "你好世界"
let 🐶🦊 = "dogcow"
```

```
let languageName = "Swift"
languageName = "Swift++"
// this is a compile-time error -
languageName cannot be changed
```

```
var friendlyWelcome = "Hello!"
friendlyWelcome = "Bonjour!"
// friendlyWelcome 现在是 "Bonjour!"
```

註解

```
// This is a comment.

/* This is also a comment
but is written over multiple lines.
*/

/* This is the start of the first multiline comment.
 /* This is the second, nested multiline comment. */
This is the end of the first multiline comment. */
```

整數

```
let minValue = UInt8.min  // minValue is equal to 0, and is of type UInt8
let maxValue = UInt8.max  // maxValue is equal to 255, and is of type UInt8
```

浮點數

```
let pi = 3.14159
```

資料型別安全和型別推測

```
let meaningOfLife = 42
// meaningOfLife is inferred to be of type Int

let pi = 3.14159
// pi is inferred to be of type Double

let anotherPi = 3 + 0.14159
// anotherPi is also inferred to be of type Double
```

數值表示法

```
let decimalInteger = 17
let binaryInteger = 0b10001       // 17 in binary notation
let octalInteger = 0o21           // 17 in octal notation
let hexadecimalInteger = 0x11     // 17 in hexadecimal notation


let paddedDouble = 000123.456
let oneMillion = 1_000_000
let justOverOneMillion = 1_000_000.000_000_1
```

數值類型的轉換

整數和浮點數間的轉換

- let twoThousand: UInt16 = 2_000
- let one: UInt8 = 1
- let twoThousandAndOne = twoThousand + UInt16(one)

整數和浮點數間的轉換

- let three = 3
- let pointOneFourOneFiveNine = 0.14159
- let pi = Double(three) + pointOneFourOneFiveNine
- // pi equals 3.14159, and is inferred to be of type Double

- let integerPi = Int(pi)

資料類型的小名

```
typealias AudioSample = UInt16
```

- `var maxAmplitudeFound = AudioSample.min`
- `// maxAmplitudeFound is now 0`

布林值

```
let orangesAreOrange = true
let turnipsAreDelicious = false

if turnipsAreDelicious {
    print("Mmm, tasty turnips!")
} else {
    print("Eww, turnips are horrible.")
}
// Prints "Eww, turnips are horrible."
```

Tuples

```
let http404Error = (404, "Not Found")
// http404Error is of type (Int, String), and equals (404, "Not Found")

let (statusCode, statusMessage) = http404Error
print("The status code is \(statusCode)")
// Prints "The status code is 404"
print("The status message is \(statusMessage)")
// Prints "The status message is Not Found"

let (justTheStatusCode, _) = http404Error
print("The status code is \(justTheStatusCode)")
// Prints "The status code is 404"
```

Tuples

```
print("The status code is \(http404Error.0)")
// Prints "The status code is 404"
print("The status message is \(http404Error.1)")
// Prints "The status message is Not Found"

let http200Status = (statusCode: 200, description: "OK")

print("The status code is \(http200Status.statusCode)")
// Prints "The status code is 200"
print("The status message is \(http200Status.description)")
// Prints "The status message is OK"
```

Optionals

```
let possibleNumber = "123"
let convertedNumber = Int(possibleNumber)
// convertedNumber is inferred to be of type "Int?", or "optional Int"
```

```
var serverResponseCode: Int? = 404
// serverResponseCode contains an actual Int value of 404
serverResponseCode = nil
// serverResponseCode now contains no value
```

```
var surveyAnswer: String?
// surveyAnswer is automatically set to nil
```

Optionals

if判斷式和強制打開

```
if convertedNumber != nil {
    print("convertedNumber contains some integer value.")
}
// Prints "convertedNumber contains some integer value."


if convertedNumber != nil {
    print("convertedNumber has an integer value of \(convertedNumber!).")
}
// Prints "convertedNumber has an integer value of 123."
```

Optionals

Optional Binding

```
if let actualNumber = Int(possibleNumber) {
    print("The string \"\(possibleNumber)\" has an integer value of \(actualNumber)")
} else {
    print("The string \"\(possibleNumber)\" could not be converted to an integer")
}
// Prints "The string "123" has an integer value of 123"
```

Optionals

Implicitly Optional Binding

```
if let definiteString = assumedString {
    print(definiteString)
}
// Prints "An implicitly unwrapped optional string."
```