iOS行動程式基礎開發上架

第六堂：列舉

本堂教學重點

1. 列舉語法
2. 使用switch檢查列舉值
3. 循環訪問列舉
4. 關聯值
5. 原生值

1.列舉語法

```
enum SomeEnumeration {
    // enumeration definition goes here
}

enum CompassPoint {
    case north
    case south
    case east
    case west
}

enum Planet {
    case mercury, venus, earth, mars, jupiter, saturn, uranus, neptune
}

var directionToHead = CompassPoint.west

directionToHead = .east
```

2.使用switch檢查列舉值

```
directionToHead = .south
switch directionToHead {
case .north:
    print("Lots of planets have a north")
case .south:
    print("Watch out for penguins")
case .east:
    print("Where the sun rises")
case .west:
    print("Where the skies are blue")
}
// Prints "Watch out for penguins"
```

```
let somePlanet = Planet.earth
switch somePlanet {
case .earth:
    print("Mostly harmless")
default:
    print("Not a safe place for humans")
}
// Prints "Mostly harmless"
```

3.循環訪問列舉

```
enum Beverage: CaseIterable {
    case coffee, tea, juice
}
let numberOfChoices = Beverage.allCases.count
print("\(numberOfChoices) beverages available")
// Prints "3 beverages available"

```

```
for beverage in Beverage.allCases {
    print(beverage)
}
// coffee
// tea
// juice
```

4.關聯值

```
enum Barcode {
    case upc(Int, Int, Int, Int)
    case qrCode(String)
}
```


```
var productBarcode = Barcode.upc(8, 85909, 51226, 3)
```

```
productBarcode = .qrCode("ABCDEFGHIJKLMNOP")
```

4.關聯值

```
switch productBarcode {
case .upc(let numberSystem, let manufacturer, let product, let check):
    print("UPC: \(numberSystem), \(manufacturer), \(product), \(check).")
case .qrCode(let productCode):
    print("QR code: \(productCode).")
}
// Prints "QR code: ABCDEFGHIJKLMNOP."
```

```
switch productBarcode {
case let .upc(numberSystem, manufacturer, product, check):
    print("UPC : \(numberSystem), \(manufacturer), \(product), \(check).")
case let .qrCode(productCode):
    print("QR code: \(productCode).")
}
// Prints "QR code: ABCDEFGHIJKLMNOP."
```

5.原生值

```
enum ASCIIControlCharacter: Character {
    case tab = "\t"
    case lineFeed = "\n"
    case carriageReturn = "\r"
}
```

5.原生值

暗地裏指定原生值

```
enum Planet: Int {
    case mercury = 1, venus, earth, mars, jupiter, saturn, uranus, neptune
}


enum CompassPoint: String {
    case north, south, east, west
}


let earthsOrder = Planet.earth.rawValue
// earthsOrder is 3

let sunsetDirection = CompassPoint.west.rawValue
// sunsetDirection is "west"

```

5.原生值

使用原生值初始化

```
let possiblePlanet = Planet(rawValue: 7)
// possiblePlanet is of type Planet? and equals Planet.uranus
```

```
let positionToFind = 11
if let somePlanet = Planet(rawValue: positionToFind) {
    switch somePlanet {
    case .earth:
        print("Mostly harmless")
    default:
        print("Not a safe place for humans")
    }
} else {
    print("There isn't a planet at position \(positionToFind)")
}
// Prints "There isn't a planet at position 11"
```