import UIKit  
//: # Enumerations  
//: ### An enumeration defines a common type for a group of related values and enables you to work with those values in a type-safe way within your code  
/\*:  
 ### Syntax:  
 \*enum SomeEnumeration {\*  
   
 \*\/\/ enumeration definition goes here\*  
   
 \*}\*  
\*/  
enum CompassPoint {  
 case north  
 case south  
 case east  
 case west  
}  
//: ### Multiple cases can appear on a single line, separated by commas:  
enum Planet {  
 case mercury, venus, earth, mars, jupiter, saturn, uranus, neptune  
}  
  
var directionToHead = CompassPoint.west  
//: set it to a different CompassPoint value using a shorter dot syntax:  
directionToHead = .east  
//: ### Matching Enumeration Values with a Switch Statement  
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"  
//: ### provide a default case when it is not appropriate to provide a case for every enumeration case  
let somePlanet = Planet.earth  
switch somePlanet {  
case .earth:  
 print("Mostly harmless")  
default:  
 print("Not a safe place for humans")  
}  
// Prints "Mostly harmless"  
//: ## Raw Values  
//: ### Raw values can be strings, characters, or any of the integer or floating-point number types  
enum ASCIIControlCharacter: Character {  
 case tab = "\t"  
 case lineFeed = "\n"  
}  
//: ### Implicitly Assigned Raw Values  
//: \* when integers are used for raw values, the implicit value for each case is one more than the previous case. If the first case doesn’t have a value set, its value is 0  
//: \* a refinement of the earlier Planet enumeration  
enum Planet2: Int {  
 case mercury = 1, venus, earth, mars, jupiter, saturn, uranus, neptune  
}  
//: ### When strings are used for raw values, the implicit value for each case is the text of that case’s name  
//: ### a refinement of the earlier CompassPoint enumeration  
enum CompassPoint2: String {  
 case north, south, east, west  
}  
//: ### access the raw value of an enumeration case  
//: \* rawValue property  
let earthsOrder = Planet2.earth.rawValue  
// earthsOrder is 3  
  
let sunsetDirection = CompassPoint2.west.rawValue  
// sunsetDirection is "west"  
//: ## Initializing from a Raw Value  
//: ### If you define an enumeration with a raw-value type, the enumeration automatically receives an initializer that takes a value of the raw value’s type (as a parameter called rawValue) and returns either an enumeration case or nil  
let possiblePlanet = Planet2(rawValue: 7)  
// possiblePlanet is of type Planet? and equals Planet.uranu  
  
let positionToFind = 11  
if let somePlanet = Planet2(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"