Swift Enum

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


var direction = CompassPoint.north
direction = .south
```

Associated Values
Enums can store associated values of different types:

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

var productBarcode = Barcode.upc(8, 85909, 51226, 3)
productBarcode = .qrCode("ABCDEFGHIJKLMNOP")

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).")
}
```

Raw Values
Enums can also be created with raw values, which are of a specific type (string, integer, or floating-point):

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

let earthOrder = Planet.earth.rawValue // 3

if let planet = Planet(rawValue: 3) {
   print("Planet \(planet) found.")
```

```
}
```

Methods
Enums can have instance methods:

```
enum CompassPoint {
    case north, south, east, west

    func description() -> String {
        switch self {
        case .north:
            return "North"
        case .south:
            return "South"
        case .east:
            return "East"
        case .west:
            return "West"
        }
    }
}

let direction = CompassPoint.east
print(direction.description()) // "East"
```