

Enum

בשיעור זה נכיר

- מספור אלקטרוני בשפת Swift

Enum

Enum נועד לזיהוי ועבודה אל מול קבוצת ערכים מוגדרת

כל ערך יכול להתאים חד חד ערכית לערך (מספרי שלם או מספרי ממשי, או אפילו מחרוזת)

בשפת Swift ערך enum הוא אובייקט.
אשר על כן, ניתן להגדיר עליו מתודות ולבצע איתו ועליו פעולות

נראה את צורות ההגדרה והעבודה של enum

Enum

```
enum ClassifieLevel{  
    case NotClassifie  
    case Saved  
    case Limited  
    case Secret  
    case TopSecret  
}
```

```
enum IntelligenceClassifieLevel{  
    case Green, Yellow, Blue, Red, Shos  
}
```

דוגמא ל enum בסיסי

Enum

```
var level1 = ClassefieLevel.NotClassefie  
level1 = .Saved
```

```
var level2 : IntelligenceClassefieLevel = .Shos
```

```
switch level2{  
case .Green:  
    println("Green")  
case .Yellow:println("Yellow")  
case .Blue:  
    println("Blue")  
case .Red:  
    println("Red")  
    fallthrough //continune to next cases  
    //if not the fallthrough the switch will end here  
case .Shos:  
    println("Shos")  
    fallthrough  
case .Shos, .Red:  
    println("Allow to enter the bunker")  
}
```

שימוש בסיסי ב enum

Enum

שימוש בסיסי ב enum - המשך

```
var level1 = ClassefieLevel.NotClassefie  
level1 = .Saved
```

```
switch level1{  
case .NotClassefie, .Saved:  
    println("Access Denied")  
default:  
    println("Access Granted")  
}
```

Enum

```
enum IsraelAreaCode : Int{  
    case Jerusalem = 2, Center, North  
    case South = 8, Lowland  
}
```

```
let code = IsraelAreaCode.South  
var rawVal = code.rawValue
```

```
let myCode = IsraelAreaCode(rawValue: 3)  
rawVal = myCode!.rawValue
```

enum
בעל ערך גולמי
מספרי

Enum

```
enum ASCIIControlCharacters : String{  
    case BreakLine = "\n"  
    case Tab = "\t"  
    case EndOfText = "\0"  
    case Backslash = "\\"  
    case DoubleQuote = "\""  
    case SingleQuote = "'"  
    case CarriageReturn = "\r"  
}
```

enum בעל ערך מחרוזתי

```
var lineFeed = ASCIIControlCharacters.BreakLine  
var string = "hello \(\lineFeed.toRaw())world"
```

Enum

enum בעל ערך תוו

```
enum SpecialCharacter : Character{  
    case SparkingHeart = "\u{1f496}"  
    case Sekelton = "\u{14840}"  
    case Bride = "\u{1f470}"  
    case Groom = "\u{1f471}"  
    case Baby = "\u{1f476}"  
    case ForkAndKnife = "\u{1f374}"  
    case Monkey = "\u{1f412}"  
}  
  
let baby = SpecialCharacter.Baby.rawValue  
let sChar = SpecialCharacter(rawValue:  
    "\u{1f496}")
```

"👶"
{(Enum Value)}

Enum

```
enum CarType{  
    case Family  
    case MiniVan  
    case SemiTrailer  
    case FullTrailer
```

```
func stringRepresentation() -> String{  
    switch self{  
    case .Family:  
        return "Family"  
    case .MiniVan:  
        return "Min Van"  
    case .SemiTrailer:  
        return "Semi Trailer"  
    case .FullTrailer:  
        return "Full Trailer"  
    }  
}
```

```
var carType = CarType.MiniVan  
var rep = carType.stringRepresentation()
```

פונקציה על ערך enum

```

func checkIfAuthorizeToDrive(#license : License) -> Bool{
  switch self{
  case .Family:
    switch license{
    case .A:
      return false
    default:
      return true
    }
  case .MiniVan:
    switch license{
    case .B, .C , .D:
      return true
    default:
      return false
    }
  case .SemiTrailer:
    switch license{
    case .A , .B:
      return false
    default:
      return true
    }
  case .FullTrailer:
    switch license{
    case .D:
      return true
    default:
      return false
    }
  }
}

```

Enum

```

enum License{
  case A //motorCycle
  case B //4 ton
  case C //12 ton
  case D //Unlimited
}

```

פונקציה על ערך enum - המשך

```

var carType = CarType.MiniVan
var rep = carType.stringRepresentation()

var l = License.B
var canDrive = carType.checkIfAuthorizeToDrive(license: l)

```

Enum

לסיכום

ראינו שימוש ב enum בשפת Swift
החל משימוש והגדרת enum בסיסי
כלה בבניית enum מתקדם

הערה: ניתן לבנות גם enum שערכיו הם tuple

Enum

שאלות?