

Instance Methods

בשיעור זה נכיר

פונקציות הכתובות על עצם - או ליתר דיוק

”מתודות עצם”

Instance Methods

פונקציה המוגדרת על אובייקט נקראת מתודה.

ניתן להגדיר מתודה לעצמים הנוצרים ממחלקה.

ניתן להגדיר מתודה על עצמים הנוצרים ממבנה.

וכן ניתן להגדיר מתודה על עצמי enum.

בעת קריאה למתודה, יש לציין ראשית מי הוא העצם, ואז לשלוח אליו\עליו את הקריאה.

```
class Robot{
```

```
    func turnLeft(){  
        //turning left  
    }
```

```
    func turnRight(){  
        for _ in 0...2{  
            turnLeft()  
        }  
    }
```

```
    func stepForward() -> Bool{  
        //stepping forward, if succeeded - return true  
        return true  
    }
```

```
    func walkSteps(steps : UInt) -> Bool{  
        for _ in 0...steps{  
            if stepForward() == false{  
                return false  
            }  
        }  
        return true  
    }
```

```
}
```

Instance Methods

דוגמא למתודה על עצם הנוצר ממחלקה

Instance Methods

שימוש בקוד דלעיל

```
var basicRobotBill = Robot()  
basicRobotBill.turnLeft()  
basicRobotBill.turnRight()  
  
basicRobotBill.stepForward()  
basicRobotBill.walkSteps(10)
```

Instance Methods

```
struct Circle{  
    var radius : Double  
    var center : Point  
  
    init(radius : Double, center : Point){  
        self.radius = radius  
        self.center = center  
    }  
  
    func getScope() -> Double{  
        return M_PI * self.getDiameter()  
    }  
  
    func getDiameter() -> Double{  
        return 2 * radius  
    }  
}
```

דוגמא למתודה על עצם הנוצר ממבנה

```
var circle = Circle(radius: 5, center: Point(v: 10, h: 10))  
var scope = circle.getScope()
```

Instance Methods

```
enum Card : UInt8{  
    case Ace = 1  
    case Two,Three,Four,Five,Six,Seven,Height,Nine,Ten  
    case Jack,Queen,King
```

דוגמא למתודה על עצם הנוצר מ enum

```
func isGreaterThan(otherCard : Card) -> Bool{  
    switch self{  
        case .Ace:  
            return !(otherCard == .Ace)  
  
        default:  
            return toRaw() > otherCard.toRaw()  
    }  
}
```

```
func canPairWith(otherCard : Card) -> Bool{  
    return self.toRaw() == otherCard.toRaw()  
}
```

```
}
```

Instance Methods

```
var card1 = Card.Ace  
var card2 : Card = .Two
```

דוגמת שימוש בקוד הנ"ל

```
var isBigger = card1.isGreaterThan(card2)  
var canPair = card2.canPairWith(card1)
```

Instance Methods

שם חיצוני לארגומנטים.

החל מהארגומנט השני, שם הארגומנט הוא גם השם החיצוני לארגומנט.

דוגמא:

```
class Person{  
    func say(something : String, toPerson : Person){  
  
    }  
}
```

```
var aharon = Person()  
var moshe = Person()
```

```
moshe.say("Git Morgen", toPerson: aharon)
```

המטרה היא כמובן
לאפשר לנו לכתוב שמות למתודות
בצורה נוחה ואינטואיטיבית

Instance Methods

```
class Person{  
    func give(#amount : Int, _toPerson : Person){  
    }  
}
```

ניתן להגדיר גם את
הארגומנט הראשון
להיות בעל שם חיצוני

```
}  
  
var aharon = Person()  
var moshe = Person()  
  
aharon.give(amount: 4, _toPerson: moshe)
```

Instance Methods

וכן אפשר לתת שם חיצוני שונה מאשר שם הארגומנט

```
class Person{  
    func get(amount : Int, fromFriend p : Person){  
  
    }  
}
```

```
var aharon = Person()  
var moshe = Person()
```

```
moshe.get(5, fromFriend: aharon)
```

Instance Methods

```
class Person{  
    func meetWith(p1 : Person, _ p2:Person){  
    }  
}
```

```
var aharon = Person()  
var moshe = Person()  
var david = Person()
```

```
david.meetWith(moshe, aharon)
```

במידה ונרצה כי לא יהיה
שם חיצוני כלל לארגומנט

נשתמש ב_

Mutating Value-Type Method

לא ניתן לשנות property של מבנה

אלא באמצעות מתודה המזוהה כ mutating

```
5 struct Square{
6     var size : Double
7
8     init(size : Double){
9         self.size = size
10    }
11
12    func setScope(scope : Double){
13        size = sqrt(scope)
14    }
15
16 }
```

```
mutating func setScope(scope : Double){
    self = Square(size: sqrt(scope))
}
```

```
5 struct Square{
6     var size : Double
7
8     init(size : Double){
9         self.size = size
10    }
11
12    mutating func setScope(scope : Double){
13        size = sqrt(scope)
14    }
15
16 }
```

Instance Methods

שאלות?