Swift Optional Value

בשיעור זה נכיר

מהו ערך אופציונלי
כיצד להצהיר
וכיצד להשתמש
בערך אופציונלי

Swift Optional Value - למה?

ישנו עניין שמאז ומעולם אתגר את נושא מדעי המחשב
מה היא הדרך הטובה ביותר להגדיר "כלום", (ברגע שמגדירים כלום, הוא אינו כלום יותר, אז כיצד נגדיר אותו מבלי לפגוע מהיותו כלום?)

הפתרון נמצא בהגדרת ה "אפס", בגרמנית - נול.
null, NULL, nil, Nil, 0
כל שפה נתנה פתרון משלה בהתאם למבנה שלה

בשפה המשתמש במצביעים כמו Objective C , הגדירו nil כמצביע לכתובת ה 0 בזיכרון, מה שהשאיר בעיה לערכים שאינם מצביעים, כדוגמת מספר שלם (מה תתנו כתשובה לשאלה, מהו האינדקס של התו a במחרוזת שאינו מכילה את התו a ?)

בשפות OOP נהגו להגדיר null ככלום, אבל אז נוצרה בעיה של קריסות על כל שטות, ולא תמיד זה אסון שמשהו הוא null.

היום נראה את הגישה של Swift

הגדרה - Swift Optional Value

בהצהרתנו על עצם כעל אופציונלי, אנו אומרים בעצם (תרגום חופשי מהמקור)

- לעצם יש ערך כלשהו

  או
- לעצם אין ערך בכלל

להבדיל מעצמים אחרים שאינם אופציונליים שבהצהרתם אנו מכריזים רק כי

- לעצם יש ערך כלשהו

ודואגים להצבה מיידית.

בהצהרה על עצם שהוא אופציונלי, נצרף להצהרה את התו ? לאחר שם המחלקה.

נראה דוגמאות

דוגמאות - Swift Optional Value

קצת דוגמאות קוד

```
var myString : String? = "some string"
myString = nil

var num : Int?
num = 4
```

Swift Optional Value - ערך חוזר

כאשר אנחנו מקבלים ערך אופציונלי, איך נתייחס אליו?

```
var numberString = "369"
var number : Int = numberString.toInt()
```

```
Int? toInt()
```

If the string represents an integer that fits into an Int, returns the corresponding integer.

דרך אחת, להצהיר עליו כאופציונלי

```
var numberString = "369"
var number : Int? = numberString.toInt()
```

דרך שניה, במידה ואנחנו בטוחים ב 100% שלא יחזור nil

```
var numberString = "369"
var number : Int = numberString.toInt()!
```

```
var testString : String! = "some string"
println(testString)
```

בדיקה - Swift Optional Value

בהינתן ערך אופציונלי, סביר ונרצה לבדוק האם הוא מכיל ערך או שאינו מכיל ערך.
נראה דוגמאות

```
if number{
    println("\(numberString) is a string that represent \(number)")
} else {
    println("\(numberString) isn't a valid string")
}
```

```
if let n = numberString.toInt(){
    //n is availbe only here
    println("\(numberString) is a string that represent \(n)")
} else {
    println("\(numberString) isn't a valid string")
}
```

```
if number != nil{

}

if number == nil{

}
```

עבודה - Swift Optional Value

אז הבנו איך יוצרים ואיך מקבלים ערכים אופציונלים שיתכן והם nil
אבל איך עובדים איתם?
מה יקרה אם נעבוד עם אחד שהוא nil??

אם אנו חוששים כי ערך הוא nil, ואנו רוצים לבצע קריאה רק אם הוא nil, אחרת שלא יבצע דבר,
עלנו להוסיף את התו ?
דוגמא:

```
var array : NSString[]?
array?.append("string")
```

לחילופין אם אנחנו בטוחים בוודאות (המהדיר אולי קצת יעזור לוודאות שלכם)
כי הערך אינו nil, עלינו להוסיף !

```
var array : String[]? = []
var count =  array!.count
```

סיכום

למדנו עבודה עם ערכים אופציונליים

לצערכם הרב, גם אם לא אהבתם את הנושא, לא תהיה לכם ברירה
לכל אורך הדרך אנחנו נפגוש ערכים אופציונליים במחלקות המובנות של אפל.

אנא חזרו על הנושא והבינו אותו היטב
קראו דוגמאות והסברים ברשת ככל שתוכלו.

שאלות?