

VBA ΣΤΟ MS EXCEL 2010

MS OFFICE 2010

(ΜΕΡΟΣ ΠΡΩΤΟ)

Με το **Excel VBA** μπορούμε να αυτοματοποιήσουμε πολλές εργασίες με το **Excel** κάνοντας χρήση των μακροεντολών. Στην ενότητα αυτή θα μάθουμε πώς να δημιουργούμε μια απλή μακροεντολή η οποία θα εκτελείται πατώντας ένα πλήκτρο εντολής.

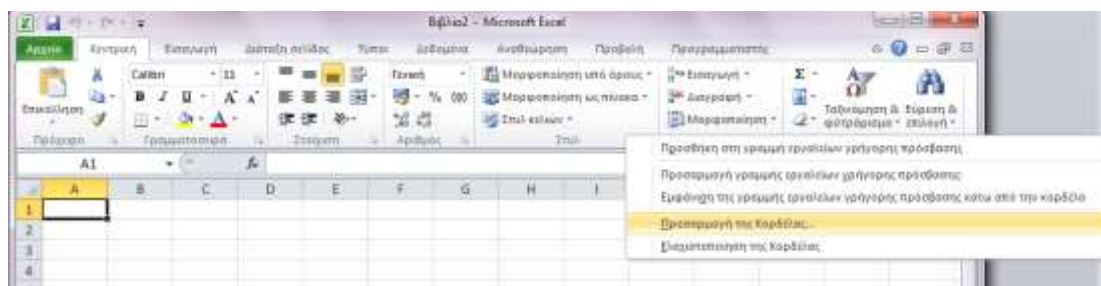
Για το λόγο αυτό αρχικά, ενεργοποιούμε την **καρτέλα Προγραμματιστής**.

With **Excel VBA** you can automate tasks in Excel by writing so called macros. In this chapter, learn how to create a simple **macro** which will be executed after clicking on a command button. First, turn on the Developer tab.

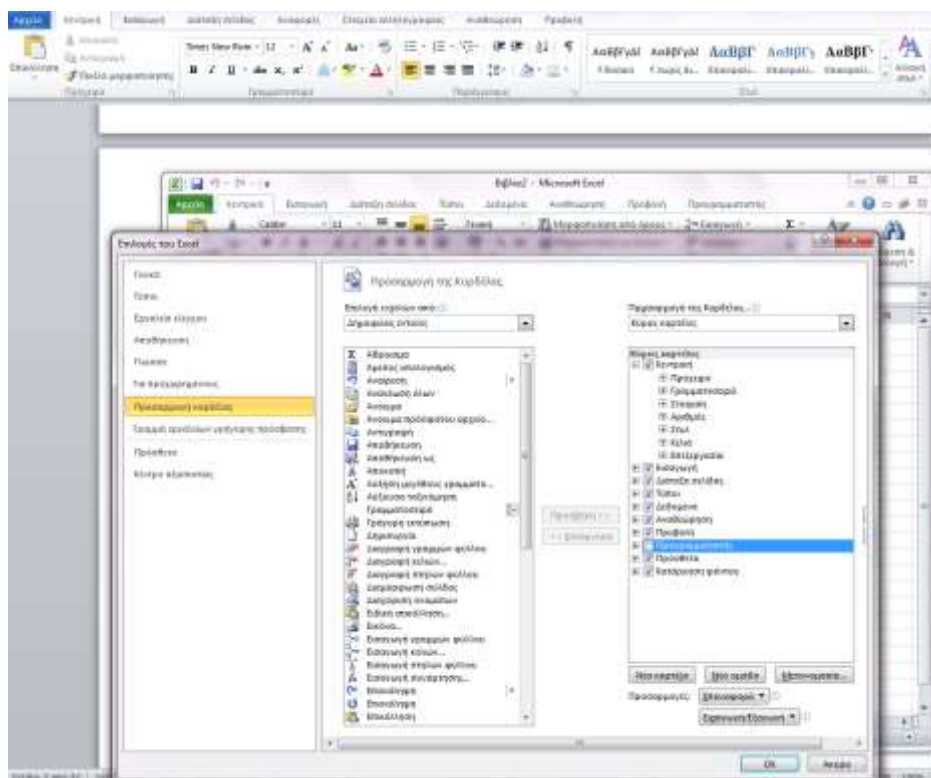
Developer Tab (Καρτέλα Προγραμματιστής)

Για να ενεργοποιήσουμε την καρτέλα Προγραμματιστής, εκτελούμε τα ακόλουθα βήματα.

1. Δεξί click οπουδήποτε επάνω anywhere στη κορδέλα (ribbon), και τότε πατάμε Προσαρμογή της κορδέλας.

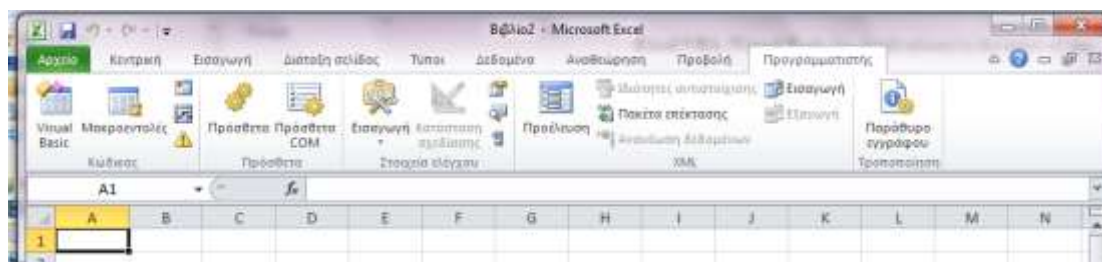


2. Κάτω από τη προσαρμογή της κορδέλας, στη δεξιά πλευρά στο κουμπί του διαλόγου επιλέγουμε κύριες καρτέλες.
3. Επιλέγουμε Προγραμματιστής στο Πλαίσιο ελέγχου.



4. Πατάμε OK.

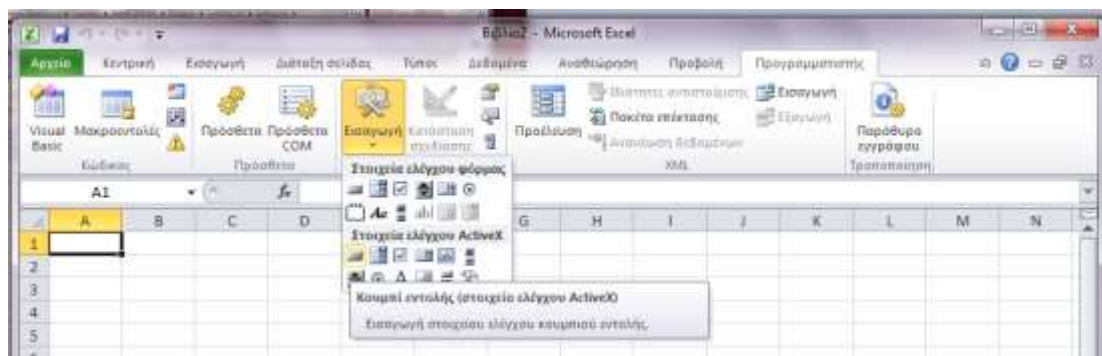
5. Μπορούμε να βρούμε την καρτέλα Προγραμματιστής ακολούθως στην καρτέλα που βρίσκεται δίπλα στην Προβολή.



Κουμπί εντολής

Για να τοποθετήσουμε ένα κουμπί εντολής στο φύλλο εργασίας, εκτελούμε τα πιο κάτω βήματα.

1. Στη καρτέλα Προγραμματιστής, πατάμε εισαγωγή.
2. Στην ομάδα πλήκτρων: στοιχεία ελέγχου ActiveX πατάμε κουμπί εντολής.



3. Σύρουμε ένα κουμπί εντολής πάνω στο φύλλο εργασίας.

Εκχώρηση μακροεντολής (macro)

Για να εκχωρήσουμε μια μακροεντολή στο κουμπί εντολής, εκτελούμε τα ακόλουθα βήματα.

1. Δεξί πάτημα στο CommandButton1 (Προσοχή: βεβαιωνόμαστε ότι βρισκόμαστε σε κατάσταση σχεδίασης (είναι επιλεγμένη)).

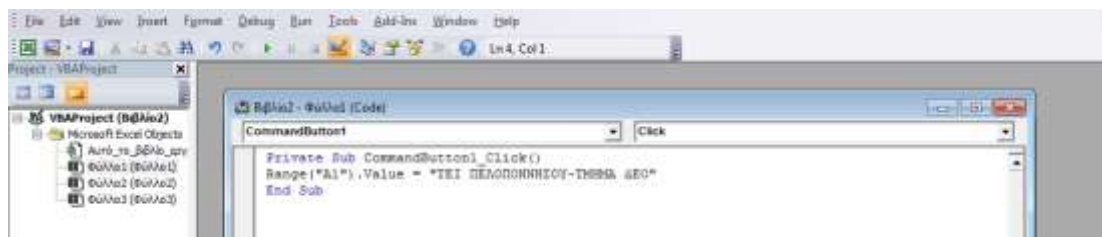
2. Πατάμε προβολή κώδικα.



Εμφανίζεται ο Visual Basic συντάκτης

3. Τοποθετούμε τον δρομέα μεταξύ της εντολής Private Sub CommandButton1_Click() και End Sub.

4. Προσθέτουμε τη γραμμή κώδικα όπως φαίνεται στο πιο κάτω σχήμα.



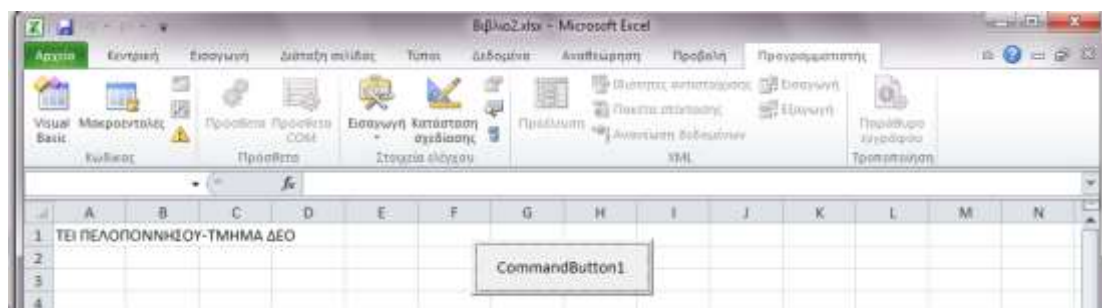
Σημείωση: Το παράθυρο αριστερά με τα ονόματα Φύλλο1, Φύλλο2, Φύλλο3 αποκαλείται Project Explorer. Αν ο Project Explorer δεν είναι ορατός, πατάμε View, Project Explorer.



5. Κλείνουμε τον συντάκτη Visual Basic.

6. Πατάμε το κουμπί εντολής στο φύλλο εργασίας (βεβαιωνόμαστε ότι η κατάσταση σχεδίασης είναι ανενεργή).

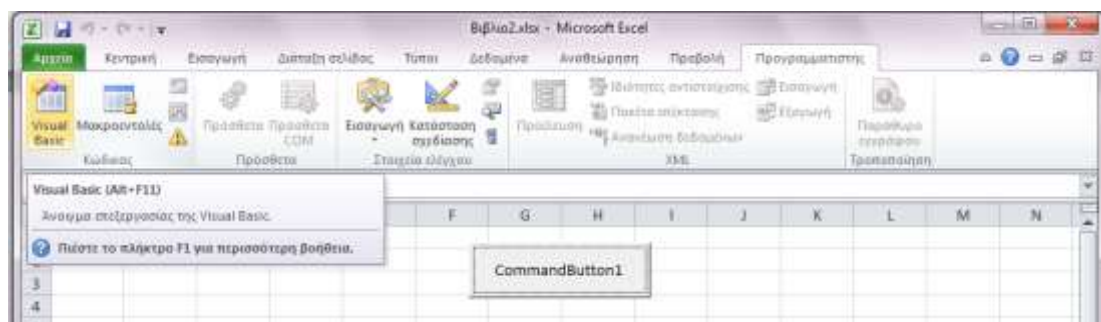
Το αποτέλεσμα εμφανίζεται πιο κάτω.



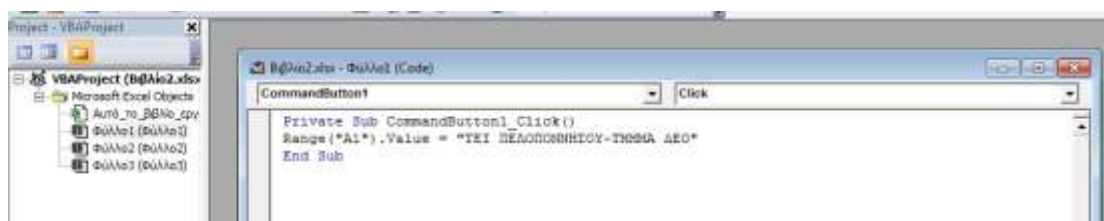
Συγχαρητήρια. Μόλις δημιουργήσατε τη πρώτη σας μακροεντολή στο Excel!

Συντάκτης Visual Basic

Για να ανοίξουμε τον Visual Basic συντάκτη, στη κορδέλα του προγραμματιστή, πατάμε, Visual Basic.



Ο συντάκτης Visual Basic εμφανίζεται.



Μπορούμε έτσι να συνεχίσουμε δημιουργώντας τις δικές μας εφαρμογές με τη χρήση των μακροεντολών.

Ας δούμε τώρα αναλυτικά μερικές βασικές έννοιες που σχετίζονται με τον προγραμματισμό.

Α. Τι είναι οι μεταβλητές VBA

Στη γλώσσα προγραμματισμού Visual Basic programming χρησιμοποιούνται δυο βασικές κατηγορίες μεταβλητών, οι συγκεκριμένου τύπου και οι γενικού τύπου **Variants**.

Οι ορισμένου τύπου μεταβλητές ορίζονται με βάση την μορφή: **Dim [Variable] As [Type]** για παράδειγμα: **Dim MyNumber As Integer**

Εκτός από ακέραιους μπορούμε να ορίσουμε πραγματικούς **As Single** ή πραγματικούς διπλής ακρίβειας **As Double**. Ενώ τύπους που επιδέχονται αλφαριθμητικά ορίζονται με το γενικό τύπο **variant** ή **As String**.

Οι τύπου γενικού τύπου μεταβλητές (**Variant**) δεν χρειάζεται να οριστούν προηγουμένως. Για να δώσουμε μια τιμή στη μεταβλητή οποιουδήποτε τύπου αρκεί να γράψουμε το σύμβολο της ισότητας, όπως στο παράδειγμα: **MyNumber = 10**

Τα ονόματα μεταβλητών πρέπει να είναι γραμμένα με λατινικούς χαρακτήρες και να ακολουθούν μερικούς απλούς κανόνες:

- Μια μεταβλητή δεν μπορεί να ξεκινάει από κάποιον αριθμό.

- - Μια μεταβλητή δεν μπορεί να έχει κενούς χαρακτήρες ή τελείες.
- - Μια μεταβλητή δεν μπορεί να περιέχει οποιοδήποτε από τα παρακάτω σύμβολα: !, %, 1/2, #, \$

B. Τι είναι οι Πίνακες (Arrays)

Οι Arrays απεικονίζουν διανύσματα, δηλαδή μονοδιάστατους πίνακες. Ένας τύπος Array συντάσσεται ως εξής: **Dim [ArrayName]([from] To [to]) As [type]** Dim [Variable] As [Type]

πχ: **Dim MyArray(1 To 4) As Integer**, όπου έχουμε ορίσει έναν πίνακα 4 θέσεων που η πρώτη θέση ξεκινάει από το 1 και η τελευταία είναι η 4.

Για να δώσουμε μια τιμή στην array θα πρέπει να δώσουμε τιμές σε κάθε έναν όρο:

MyArray[1] = 13

MyArray[2] = 7

MyArray[3] = 8

MyArray[4] = 9

Έτσι η μεταβλητή MyArray ορίζει το διάνυσμα (13, 7, 8, 9).

Οι πίνακες ή πολυδιάστατα arrays ορίζονται ως: **Dim [ArrayName]([Dimension 1], [Dimension 2], ..., [Dimension n]) As [type]**

ή

Dim [ArrayName]([dimension 1 from] To [dimension 1 to], ..., [dimension n from] To [dimension n to]) As [type]

πχ: **Dim MyArray(2, 3) As Integer** ή **Dim MyArray(1 To 3, 1 To 4) As Integer**

Γ. Σχόλια & Κώδικας

Ο κώδικας που γράφουμε πρέπει να είναι

- ευανάγνωστος,
- χωρίς περιττές πράξεις και
- να έχει σχόλια όπου αυτό κρίνεται απαραίτητο.

Υπάρχουν κάποιοι απλοί κανόνες για τη γραφή κώδικα που αν τους ακολουθούμε μπορούμε να γράφουμε και να μπορούμε εύκολα να ξαναδιαβάσουμε τη δουλειά που έχουμε κάνει, αλλά και να συνεργαστούμε σε ομάδα ή να παραδώσουμε τη δουλειά αυτή σε τρίτους.

Απλοί κανόνες γραφής

- - Οι μεταβλητές μας πρέπει να δέχονται απλά ονόματα που να έχουν νόημα σε εμάς, π.χ. **cost**, **patient**, **product**, **x**, κλπ
- - Οι μεταβλητές μας πρέπει να περιέχουν μέχρι δυο ή τρεις λέξεις ώστε να μην είναι αρκετά μεγάλες. Κάθε μεταβλητή ξεκινάει από πεζό γράμμα. Οι λέξεις μεταξύ τους ξεχωρίζουν διότι η δεύτερη και κάθε νέα λέξη ξεκινάει από κεφαλαίο γράμμα. Για παράδειγμα: **averageSummary** ή **subTotal**, ή **medianValue**
- Οι ρουτίνες και οι functions ακολουθούν τον ίδιο τρόπο που δίνουμε ονόματα στις μεταβλητές, μόνον που το όνομα ξεκινάει πάντα από κεφαλαίο πχ **Average**.
- Πάνω από κάθε ρουτίνα θα πρέπει να σημειώνουμε μια δυο γραμμές σχόλια που θα περιγράφουν τι κάνει η ρουτίνα και τι σκοπό έχουν οι μεταβλητές. Το στάδιο αυτό μπορεί να παραληφθεί αν το όνομα της ρουτίνας παραπέμπει από μόνο του σε όλη τη διαδικασία ώστε να γνωρίζουμε τι κάνει η κάθε ρουτίνα.

Σχόλια

Τα σχόλια είναι γραμμές στον κώδικα που δεν εκτελούνται και ο μόνος λόγος ύπαρξής τους είναι να μας υπενθυμίζουν την εργασία που έχουμε κάνει. Είναι βασικό να έχουμε κρατήσει καλές σημειώσεις στον κώδικα μας ώστε να είναι ευανάγνωστος. Τα σχόλια στην VBA ορίζονται με το σύμβολο `'''` πριν από το σχόλιο.

Παράδειγμα σχολιασμού:

Function CalculateNR *Υπολογισμός τριγωνικής ρίζας*

[κώδικας]

End Function

Δ. Ρουτίνες & Functions

Μια ρουτίνα ή ένα function είναι ένα τμήμα κώδικα που είναι γραμμένο ώστε να επιτελεί μια συγκεκριμένη εργασία. Ειδικότερα, οι ρουτίνες είναι τμήματα κώδικα που μπορούν να κληθούν αυτόνομα από κάποιο control (πχ. το πάτημα ενός button) και να εκτελέσουν το τμήμα κάποιου κώδικα. Π.χ. όταν πατάμε κάποιο button να διαβάσει τις τιμές από κάποια κελιά και να επιστρέφει τιμές σε άλλα. Μια ρουτίνα μπορεί να δεχθεί κάποιες τιμές εισόδου (input arguments) και να επιστρέψει τα αποτελέσματα σε κάποιες άλλες τιμές (output arguments). Τα function έχουν την ίδια λογική με τις ρουτίνες, δηλαδή εκτελούν ένα τμήμα ενός κώδικα, όμως τα function επιστρέφουν πάντοτε μια τιμή. Ένα function μπορεί να κληθεί από ένα φύλλο excel και να κάνουμε πράξεις πάνω σε αυτό.

Η σύνταξη μιας ρουτίνας είναι η εξής:

Sub [όνομα ρουτίνας]([argument], [argument], ..., [argument])

[Εδώ γράφουμε τον κώδικα που θέλουμε να εκτελεστεί από αυτή τη ρουτίνα]

End Sub

Παράδειγμα:

Sub JustExample(welcomeMessage)

Dim cellName As String

cellName = "A2"

Worksheets(1).Range(cellName).Value = "ΤΕΙ ΠΕΛΟΠΟΝΕΣΕ"
Worksheets(1).Range("A3").Value = welcomeMessage

End Sub

Η σύνταξη μιας function είναι η εξής:

Function [όνομα function]([argument], [argument], ..., [argument])

[Εδώ γράφουμε τον κώδικα που θέλουμε να εκτελεστεί από αυτή τη function]

[όνομα function] = [Αποτέλεσμα πράξεων]

End Function

Παράδειγμα:

Function Summary(x1, x2, x3)

Dim sum As Double sum = x1 + x2 + x3 Summary = sum

End Function

Κλήση ρουτίνας και function

Για να καλέσουμε μια ρουτίνα σε ένα τμήμα του κώδικα αρκεί να εκτελέσουμε την κλήση:

Call [όνομα ρουτίνας]

Παράδειγμα:

Dim initialX As Double

initialX = 2.51

Dim root As Double

Call Calculate(initialX, root)

Worksheets(1).Range("C2").Value = root *'type the result*

Για να καλέσουμε μια function αρκεί να την καλέσουμε με το όνομα της, για παράδειγμα:

Dim initialX As Double

initialX = 8.34

Dim root As Double

root = Calculate(initialX)

Συνθήκες if Else

Οι συνθήκες είναι τμήματα κώδικα τα οποία εκτελούνται κάτω από συγκεκριμένες προϋποθέσεις. Ο ρόλος των συνθηκών είναι να αλλάζει η ροή του προγράμματος.

Τα τμήματα αυτά στον κώδικα επισημαίνονται ως:

If [συνθήκη] Then

κώδικας που εκτελείται όταν η συνθήκη είναι αληθής (true)

Else

τμήμα κώδικα που εκτελείται όταν η συνθήκη είναι ψευδής (false)

End If

Για παράδειγμα:

If [value > 250] Then

MsgBox "Η τιμή είναι μεγαλύτερη από 250"

Else

MsgBox "Η τιμή είναι μικρότερη ή ίση με 250"

End If

Loops, βρόγχοι και επαναλήψεις

Τα loops (οι βρόγχοι) είναι τμήματα κώδικα τα οποία επαναλαμβάνονται μέχρι να ικανοποιηθεί κάποια συνθήκη. Υπάρχουν δυο βασικοί εκπρόσωποι loops,

- τα loops τύπου **For** και
- τα loops τύπου **While**.

Τα loops τύπου **For** είναι τμήματα κώδικα τα οποία επαναλαμβάνονται συγκεκριμένες φορές και αυτό καθορίζεται από τις τιμές "από" - "έως" που δηλώνονται κατά τη σύνταξη:

For [Από] To [Έως]

τμήμα κώδικα που εκτελείται n φορές, όπου $n = \text{Έως} - \text{Από} + 1$

Next i

Για παράδειγμα:

Dim sum As Double

sum = 0

For i = 1 To 100

sum = sum + i

Next i

MsgBox "Η τιμή του αθροίσματος των 100 όρων αριθμητικής προόδου είναι " & sum

Τα loops τύπου **While** είναι τμήματα κώδικα τα οποία επαναλαμβάνονται όσο ισχύει μια συγκεκριμένη συνθήκη που δηλώνεται κατά τη σύνταξη:

While [Συνθήκη]

τμήμα του κώδικα που εκτελείται για όσο η παραπάνω συνθήκη είναι αληθής (true)

WEnd

Για παράδειγμα:

Dim sum As Double

Dim i As Integer

sum = 0

i = 1 While i < 101

sum = sum + i

i = i + 1

Wend

MsgBox "Η τιμή του αθροίσματος των 100 όρων αριθμητικής προόδου είναι " & sum

Ένα από τα μεγαλύτερα πλεονεκτήματα του VBA είναι η αρμονική λειτουργία του με το Excel. Έχουμε τη δυνατότητα να εισάγουμε δεδομένα από ένα φύλλο Excel, αλλά και το ανάποδο, δηλαδή να γράψουμε αποτελέσματα σε ένα φύλλο Excel. Τα δεδομένα αυτά μπορούμε να τα μορφοποιήσουμε, να σχεδιάσουμε διαγράμματα, κλπ.

Για να γράψουμε δεδομένα σε ένα ή περισσότερα κελιά μπορούμε να χρησιμοποιήσουμε τους εξής δυο τύπους εντολών ισοδύναμα:

Στην πρώτη περίπτωση αναφερόμαστε σε συγκεκριμένο κελί ή σε συγκεκριμένα κελιά ορίζοντας την ονομασία του κελιού ή των κελιών:

Worksheets([Worksheet number]).Range([Range]).Value = [Value]

για παράδειγμα: **Worksheets(1).Range("B2").Value = 10**

ή ισοδύναμα **Application.Workbooks("[WorkBook**

name]").Worksheets("[WorkSheet name]").Range([Range]).Value = [Value]

για παράδειγμα:

Application.Workbooks("Book1.xlsx").Worksheets("Sheet1").Range("B2").

Άλλα παραδείγματα:

Worksheets(1).Range("B2:B10").Value = 11 ή

Worksheets(1).Range("B2:B10").Value =

Worksheets(1).Range("C2:C10").Value

Ενώ στην δεύτερη περίπτωση αναφερόμαστε απευθείας στο κελί σαν να είναι στοιχείο ενός πίνακα:

Worksheets([Worksheet number]).Cells([Row], [Column]).Value = [Value]

για παράδειγμα: **Worksheets(1).Cells(3, 2).Value = 10**, όπου ουσιαστικά αναφερόμαστε στο κελί **B3**, δηλαδή στο κελί που βρίσκεται στην τρίτη γραμμή και τη δεύτερη κολώνα (δηλαδή στην στήλη B).

Για να διαβάσουμε δεδομένα από κάποιο κελί μπορούμε να χρησιμοποιήσουμε το δεύτερο τρόπο γραφής, όπως φαίνεται στο παρακάτω παράδειγμα:

Worksheets(1).Cells(3, 2).Value = 10, όπου ουσιαστικά αναφερόμαστε στο κελί **B3**