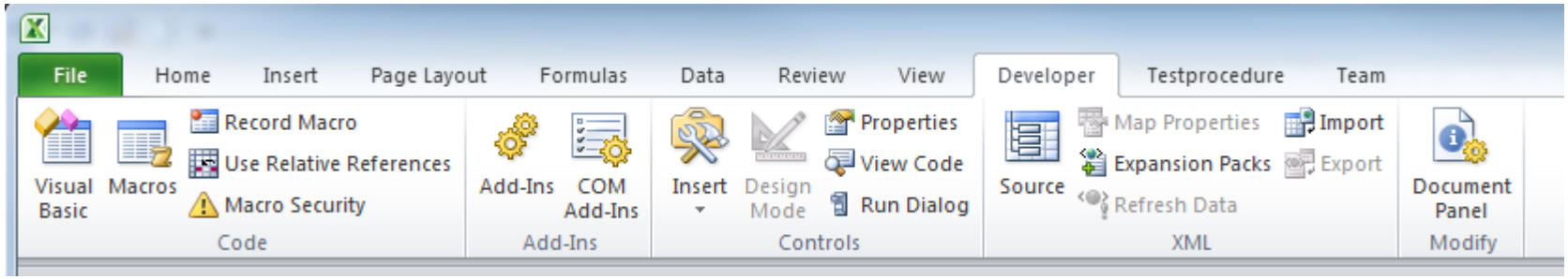
Doe het zelf tooling

Met VBA

Een goede voorbereiding:

- Start je laptop
- Start Excel
- Zet het tabblad “Ontwikkelaars” aan (Bestand, Opties, Lint aanpassen)



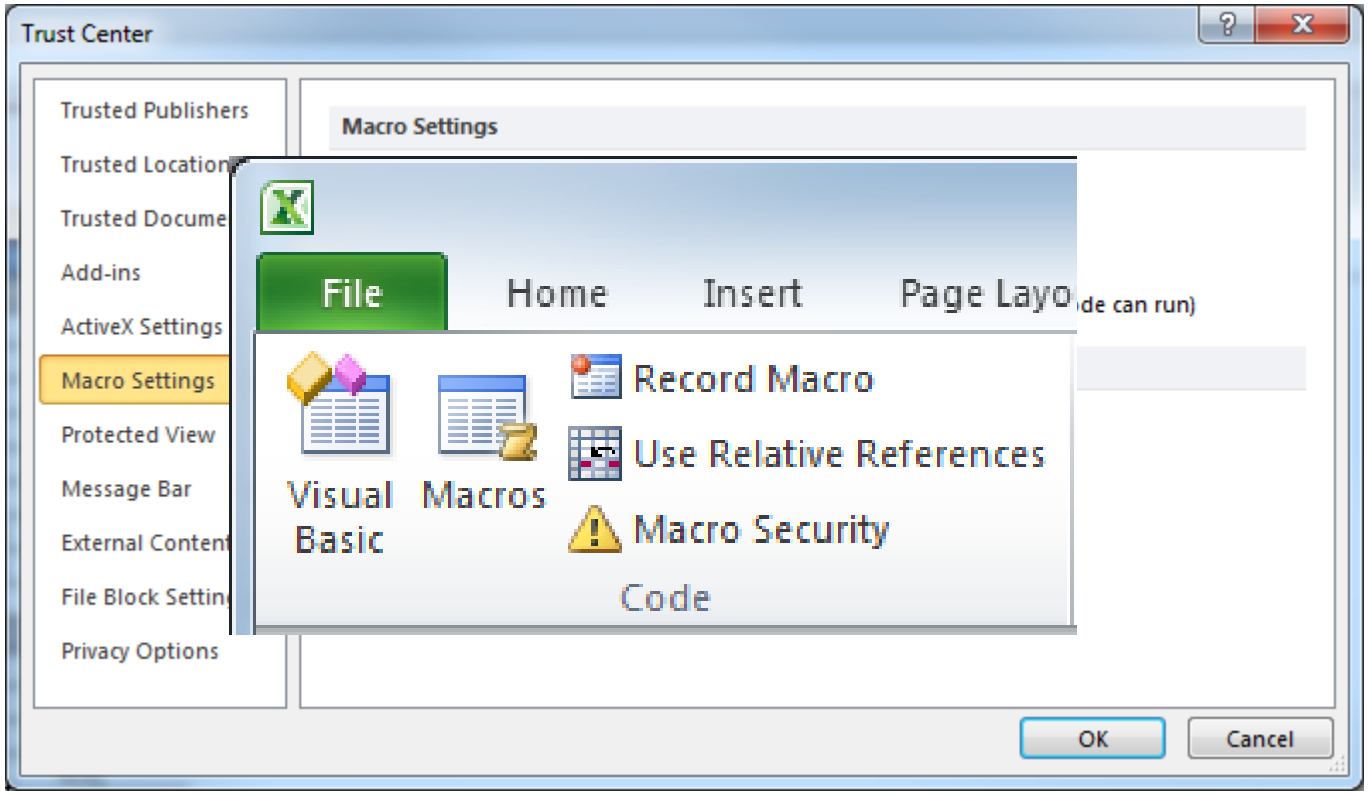
Stéphanie Heidstra

Testmanager bij Quadrant Software

15 jaar in het testvak

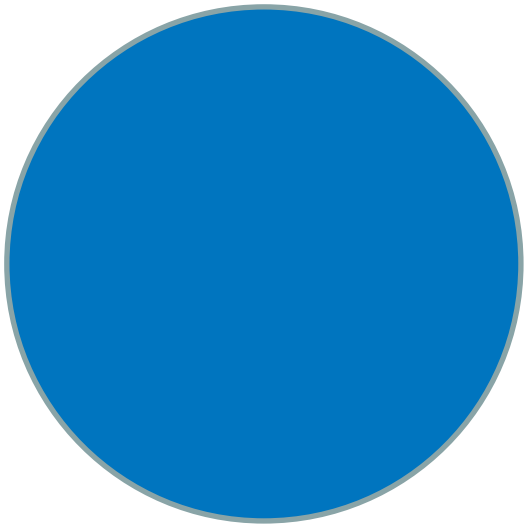
ISTQB Practitioner

Certified Ethical Hacker


Trust Center
Trusted Publishers
Trusted Location
Trusted Docume
Add-ins
ActiveX Settings
Macro Settings
Protected View
Message Bar
External Content
File Block Setting
Privacy Options
Macro Settings
File
Home
Insert
Page Layo
de can run)
Visual Basic
Macros
Record Macro
Use Relative References
Macro Security
Code
OK
Cancel

Oefening 1:

Neem een macro op die “Hello world” in een willekeurige cell zet.



Oefening 1

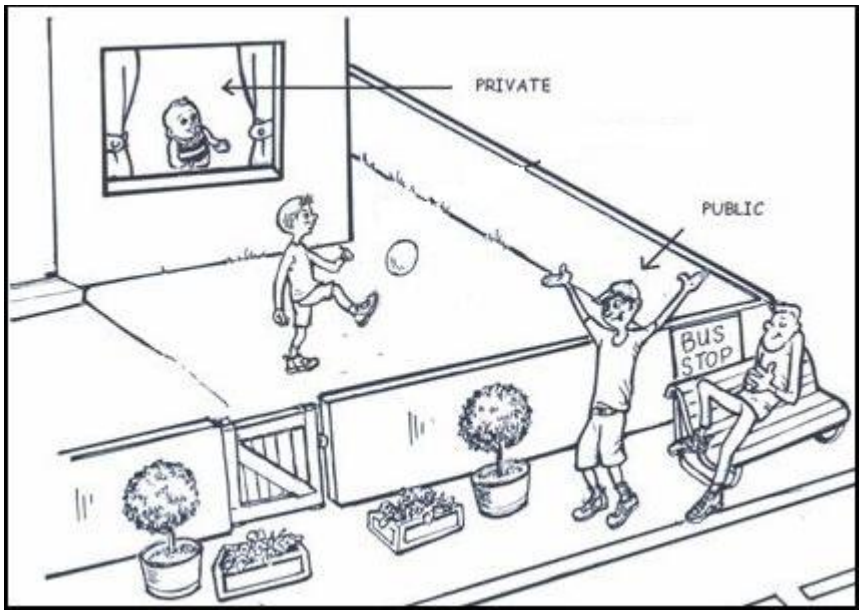
Sub
Procedure

Function
Functie

“Ga koffie zetten”

“Breng mij koffie”

Public - Private



Oefening 2:

Zet de macro die we net opgenomen hebben om van een Sub naar een Function.

Zorg dat hij van buiten af aan te roepen is en roep hem aan vanaf het worksheet.

Oefening 2

Syntax:

- If ….. Then
- Else
- EndIf

```
If sVandaag = sZondag Then
   Call Doelets
Else
   Call DoeletsAnders
End If
```

Oefening 3:

Pas de code van net aan zodat hij het volgende doet:

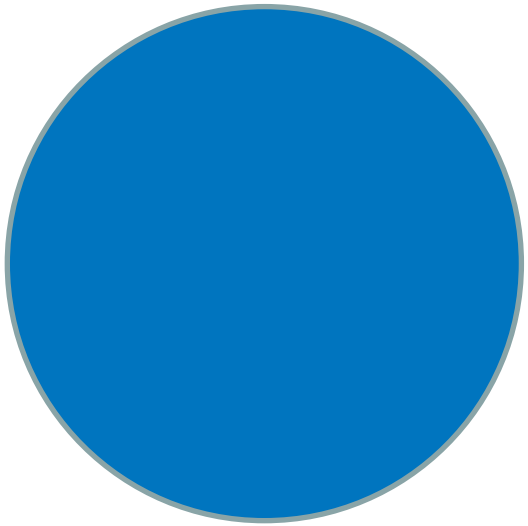
Op 13 mei tussen 9 en 1 zitten we op TestNet, dus moet de melding “Hallo TestNet” zijn.

Op diezelfde dag is het tussen 11:00 en 11:30 is het koffietijd en moet de melding “Tijd voor koffie” zijn.

Op 13 mei:

- Tussen 9:00 en 13:00: “Hallo TestNet”
- Tussen 11:00 en 11:30 “Tijd voor koffie”
- Anders “Hallo Wereld”

- Date()
- Now()
- Hour()
- Minute()
- DateSerial(yyyy,mm,dd)



Oefening 3

- Debugging
- Breaks
- Watches
- MsgBox

Oefening 3b

Syntax:

- Dim … As …
- Set

```
Dim iTeller As Integer
Dim sNaam As String

sNaam = “Joe”
iTeller = 1
```

```
Dim oSheet As WorkSheet

Set oSheet = ActiveWorkbook.Sheets(1)
```

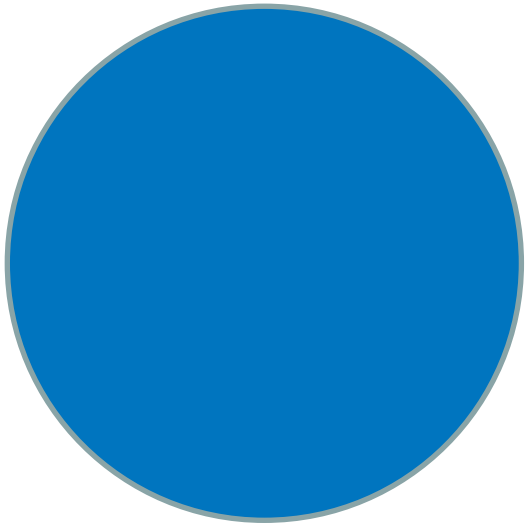
Syntax:

- For Each … In …
- Next

```
Dim oWorksheet as Worksheet

For Each oWorksheet In ActiveWorkbook.Sheets
    MsgBox(oWorksheet.Name)
Next
```

Oefening 4:

Maak op tabblad Management-info een functie in het urenveld, die op basis van de gegevens erboven de juiste uren bepaald.



Oefening 4

Syntax:

- …Find(What:=…)

```
Dim oWorksheet As Worksheet
Dim ofoundRange As Range

Set oWorksheet = ActiveWorkbook.Sheets(1)
Set ofoundRange = oWorksheet.UsedRange.Find(What:="Cem")
```

Syntax:

- …Find(What:=…)

- LookIn
- LookAt
- SearchOrder
- MatchCase

- xlValues, xlFormulas
- xlWhole, xlPart
- xlByRows, xlByColumns
- True, False

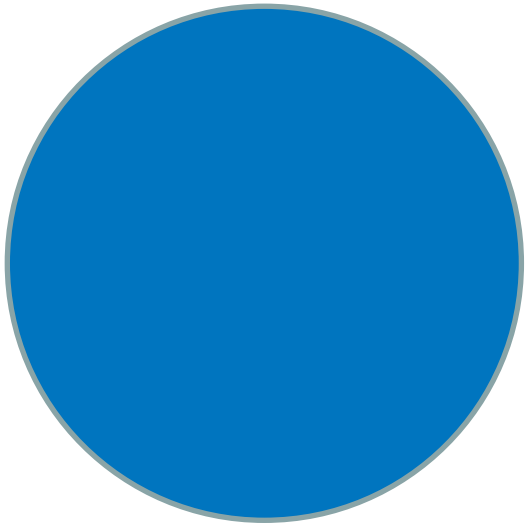
Syntax:

- …Find(What:= …)

```
Dim oWorksheet As Worksheet
Dim ofoundRange As Range

Set oWorksheet = ActiveWorkbook.Sheets(1)
Set ofoundRange = oWorksheet.UsedRange.Find(What:="Cem", _
                                            LookIn:=xlValues, _
                                            LookAt:=xlWhole, _
                                            SearchOrder:=xlByRows, _
                                            MatchCase:=False)
```

Oefening 4b:

Bouw de For Each uit oefening 4 om naar een .Find constructie.



Oefening 4b

Syntax:

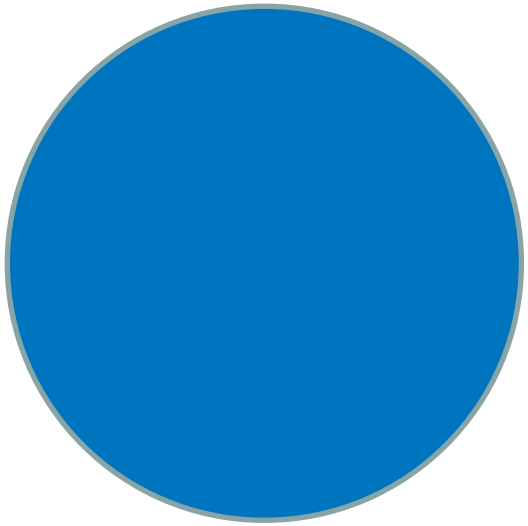
- With …
- End Wit

```
Dim oCell As Range

    Set oCell = ActiveWorkbook.Sheets(1).Cells(2, 2)

    oCell.Borders.LineStyle = xlContinuous
    oCell.Borders.Color = 255
    oCell.Borders.Weight = xlMedium
```

```
 Set oCell = ActiveWorkbook.Sheets(1).Cells(2, 2)

    With oCell.Borders
          .LineStyle = xlContinuous
          .Color = 255
          .Weight = xlMedium
    End With
```

Oefening 5:

Maak een macro die met een rode kleur in de “Totaal”-rij aangeeft in welke oneven weken Erik minder dan 40 uur gewerkt heeft.



Oefening 5

Best practices / Coding guidelines

Uitlijning

```
Dim sMelding As String
sMelding = "Hallo wereld"
If Date = DateSerial(2013, 5, 13) Then
If Hour(Now()) >= 9 And Hour(Now()) < 13 Then
If Hour(Now()) = 11 And Minute(Now()) <= 30 Then
sMelding = "Tijd voor koffie"
Else
sMelding = "Hallo TestNet!"
End If
End If
End If
Oefening3 = sMelding
```

Naamgeving

- 1 letterige variabelen
- Nietszeggende namen
- Volgnummers

- i, j, k
- Data, value, myString, iIdx
- Data2, index2

Naamgeving

```
Public Sub MakeMT940()
Dim iLine As Long
Dim lAmount As Double
Dim dMaxDate As Date

    iLine = 2
    Do While Cells(iLine, 1).Value <> "-" And Cells(iLine, 1).Value <> ""
        Call WriteRecord(iLine)
        lAmount = lAmount + Cells(iLine, 1).Value
        If dMaxDate < Cells(iLine, 3).Value And Cells(iLine, 3) <> "" Then
            dMaxDate = Cells(iLine, 3).Value
        End If
        iLine = iLine + 1
    Loop
End Sub
```

Variabele definitie

- Option Explicit
- Type declaratie

```
Dim iTeller As Integer
Dim sNaam As String
```

Naamgeving - Prefix

- String
- Integer
- Object

- strNaam, sNaam
- intTeller, iTeller
- objCell, oCell

Commentaar

```
Public Sub MakeMT940()
Dim iLine As Long
Dim lAmount As Double
Dim dMaxDate As Date

    'Beginnen bij regel 2 want regel 1 bevat kolomkoppen
    iLine = 2
    'Alle niet lege regels worden langs gelopen
    Do While Cells(iLine, 1).Value <> "-" And Cells(iLine, 1).Value <> ""
        Call WriteRecord(iLine)
        'Het totaalbedrag van de regels wordt bijgehouden
        lAmount = lAmount + Cells(iLine, 1).Value
        'De hoogste datum in de regels wordt bepaald
        If dMaxDate < Cells(iLine, 3).Value And Cells(iLine, 3) <> "" Then
            dMaxDate = Cells(iLine, 3).Value
        End If
        iLine = iLine + 1
    Loop
End Sub
```

Constanten

```
Private Const Red = 255

Sub MakeRedBorder()
    Set oCell = ActiveWorkbook.Sheets(1).Cells(2, 2)

   With oCell.Borders
         .LineStyle = xlContinuous
         .Color = Red
         .Weight = xlMedium
   End With
End Sub
```

Procedures of Functies splitsen

- Te veel regels in 1 procedure
- Te diepe nesting
- 1 taak per procedure

Syntax:

- Call …

```
Sub Hallo()

   Msgbox (“Hallo wereld!”)

End Sub
```

```
Sub OokHallo()

     Call Hallo ()

End Sub
```

Syntax:

- Input parameters

```
Sub Hallo()

   Msgbox (“Hallo wereld!”)

End Sub
```

```
Sub HalloTestNet()

     Call Hallo (“Hallo Testnet!”)

End Sub
```

```
S Sub OokHallo()

       Call Hallo ()

E End Sub
```

Complexiteit

```
Private Const Red = 255

Sub MakeRedBorder()
    Set oCell = ActiveWorkbook.Sheets(1).Cells(2, 2)
    Call SetBorderColor(oCell, red)
End Sub

Sub SetBorderColor(rRange As Range, iColor As Integer)
    With rRange.Borders
            .LineStyle = xlContinuous
            .Color = iColor
            .Weight = xlMedium
    End With
End Sub
```

King BUSINESS SOFTWARE

Error Handling

- On Error Resume Next
- On Error GoTo …

Error Handling

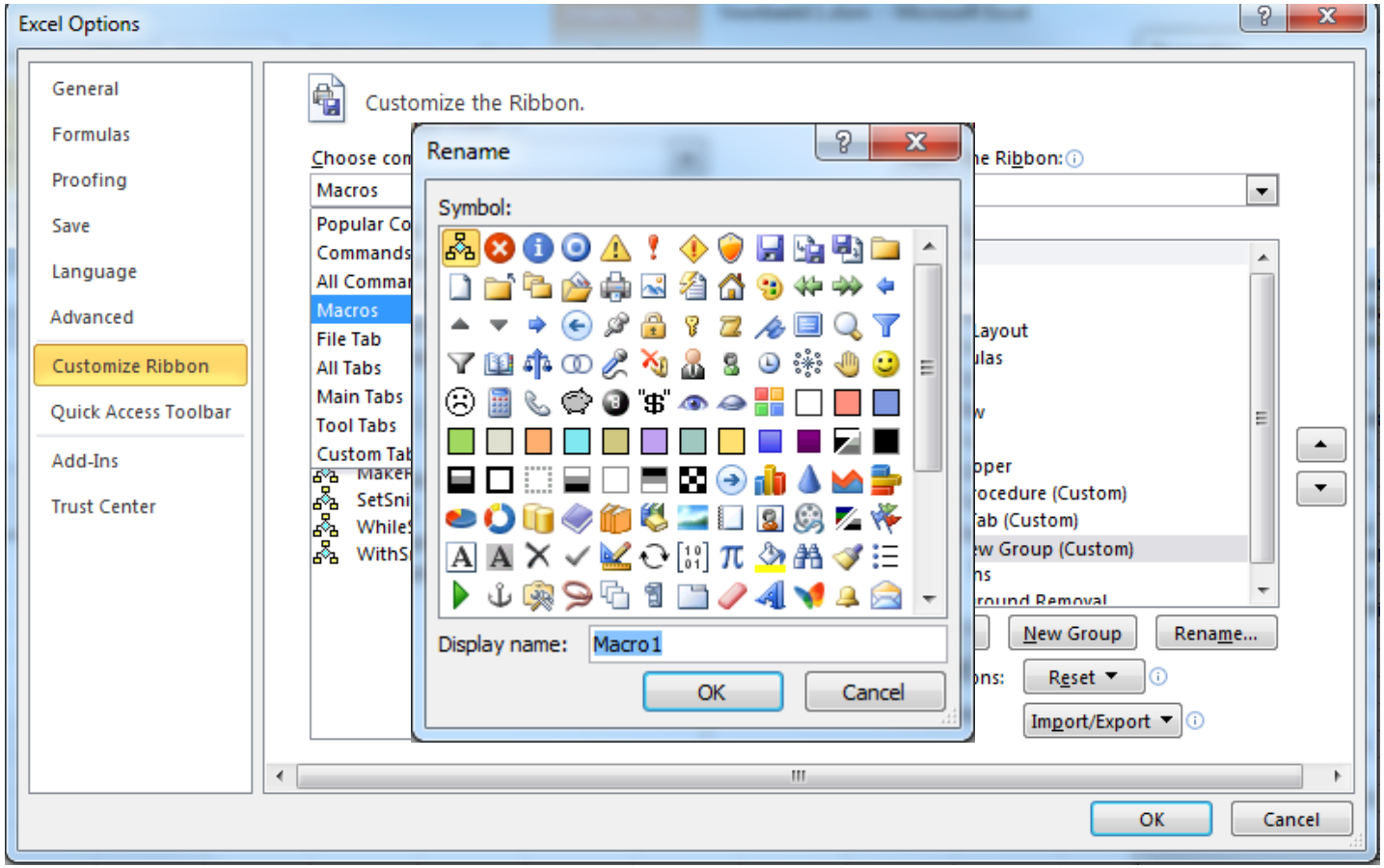
```
Sub MakeRedBorder()
    On Error GoTo ErrorHandler

    Set oCell = ActiveWorkbook.Sheets(1).Cells(2, 2)
    Call SetBorderColor(oCell, red)

Exit Sub
ErrorHandler:
   MsgBox (Err.Description)

End Sub
```

Integratie in Excel

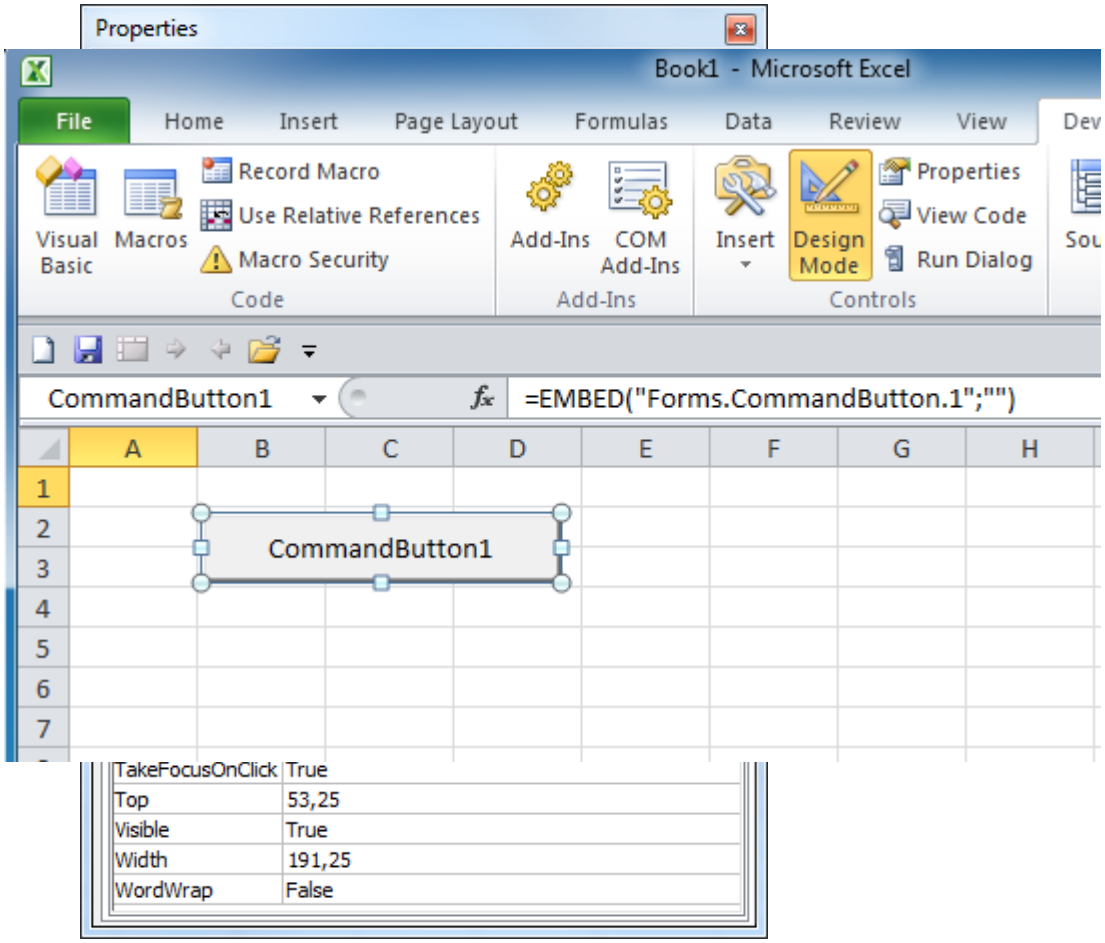
Macro starten door:

- Opnemen als functie op het werkblad
- Via de knop Macro’s op de Ontwikkelaarstab

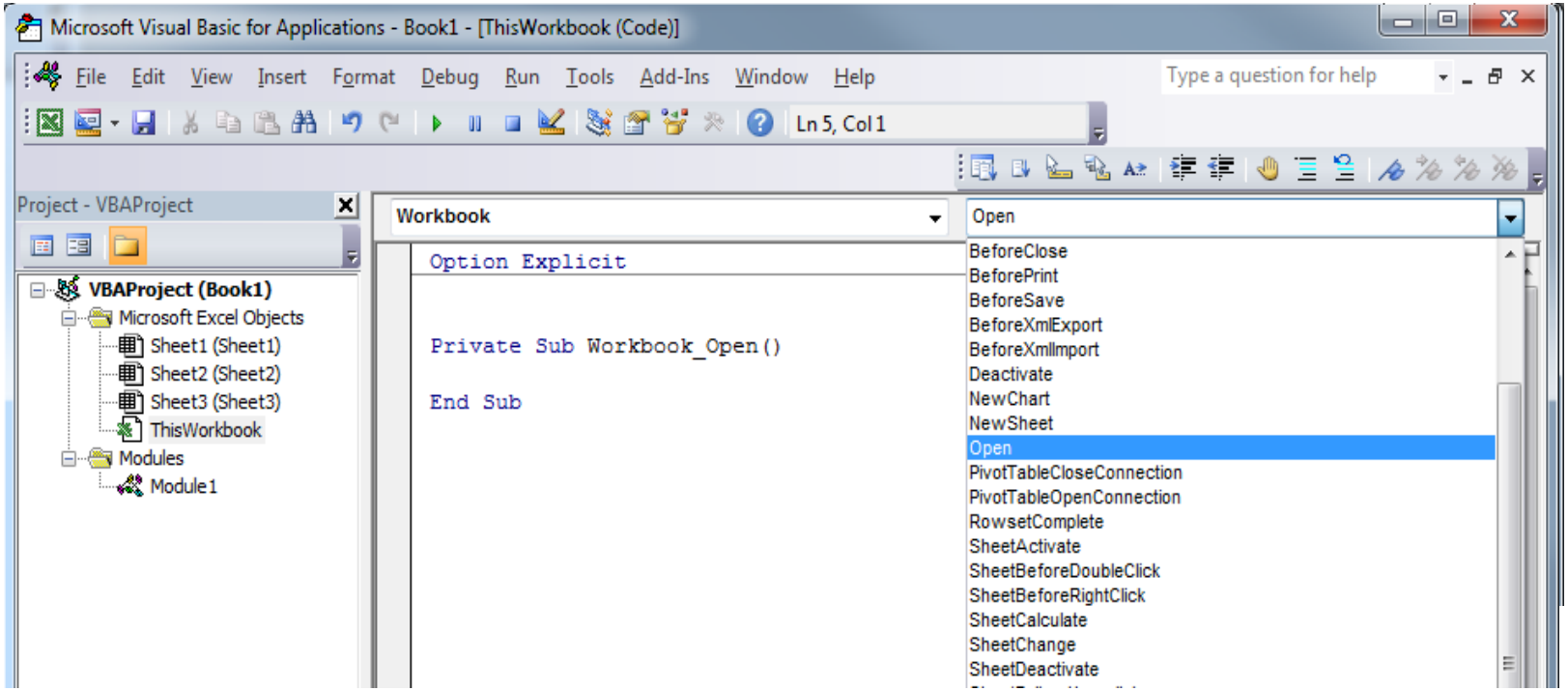
Via een knop in het Ribbon



Via een knop in het worksheet



Via Events


Microsoft Visual Basic for Applications - Book1 - [ThisWorkbook (Code)]
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Type a question for help
Ln 5, Col 1
Project - VBAProject
VBAProject (Book1)
Microsoft Excel Objects
Sheet1 (Sheet1)
Sheet2 (Sheet2)
Sheet3 (Sheet3)
ThisWorkbook
Modules
Module1
Workbook
Open
Option Explicit
Private Sub Workbook_Open()
End Sub
BeforeClose
BeforePrint
BeforeSave
BeforeXmlExport
BeforeXmlImport
Deactivate
NewChart
NewSheet
Open
PivotTableCloseConnection
PivotTableOpenConnection
RowsetComplete
SheetActivate
SheetBeforeDoubleClick
SheetBeforeRightClick
SheetCalculate
SheetChange
SheetDeactivate

Oefening 6:

Laat Excel een melding geven bij het opslaan van het workbook.

Oefening 6

Bijkomende zaken

- Ontwerpen
- Testen
- Licentieren
- Distributie
- Gebruikers
  - Support
  - Handleiding
  - Defects
- Platform-updates

King BUSINESS SOFTWARE

The end.