[Steve McDonald, Gillian Werner „Microsoft Excel VBA for Beginners Learn VBA Step by Step” [Udemy] (2022) 3](#_Toc117777026)

[Boris Paskhaver „Excel VBA Programming The Complete Guide” [Packt] (2022) 8](#_Toc117777027)

[Richard Mansfield „Mastering VBA for Microsoft® Office 365® (2019 Edition)” [Sybex] (2019).epub 9](#_Toc117777028)

[Partea 1. ÎNREGISTRAREA MACRO-URILOR ŞI ÎNCEPEREA UTILIZĂRII VBA 9](#_Toc117777029)

[Cap. 1. Înregistrarea şi rularea macro-urilor în aplicaţiile Office 9](#_Toc117777030)

[Cap. 2. Visual Basic Editor 11](#_Toc117777031)

[Cap. 3. Editarea maacro-urilor înregistrate 14](#_Toc117777032)

[Cap. 4. Crearea codului de la zero în Visual Basic Editor 14](#_Toc117777033)

[Partea 2. ÎNVĂŢAREA UTILIZĂRII VBA 14](#_Toc117777034)

[Cap. 5. Înţelegerea lucrurilor esenţiale ale sintaxei VBA 14](#_Toc117777035)

[Cap. 6. Lucrul cu variabile, constante şi enumerări 15](#_Toc117777036)

[Cap. 7. Utilizarea variabilelor tablou (Array) 16](#_Toc117777037)

[Cap. 8. Găsirea obiectelor, metodelor şi proprietăţilor necesare 17](#_Toc117777038)

[Partea 3. LUAREA DECIZIILOR ŞI UTILIZAREA CICLURILOR ŞI FUNCŢIILOR 19](#_Toc117777039)

[Cap. 9. Utilizarea funcţiilor încorporate 19](#_Toc117777040)

[Formate predefinite 20](#_Toc117777041)

[Crearea unui format numeric: 21](#_Toc117777042)

[Crearea unui format de dată calendaristică sau de timp: 22](#_Toc117777043)

[Crearea unui format şir de caractere: 23](#_Toc117777044)

[Utilizarea funcţiei Chr şi a constantelor pentru a introduce caractere speciale într-un şir de caractere: 24](#_Toc117777045)

[Utilizarea funcţiilor pentru a manipula şiruri de caractere: 25](#_Toc117777046)

[Utilizarea funcţiilor matematice VBA: 26](#_Toc117777047)

[Utilizarea funcţiilor de dată calendaristică şi de timp VBA: 27](#_Toc117777048)

[Utilizarea funcţiilor de management de fişiere: 29](#_Toc117777049)

[Cap. 10. Crearea funcţiilor 29](#_Toc117777050)

[Componentele unei funcţii 29](#_Toc117777051)

[Crearea unei funcţii pentru Word 29](#_Toc117777052)

[Crearea unei funcţii pentru Excel 30](#_Toc117777053)

[Crearea unei funcţii pentru PowerPoint 30](#_Toc117777054)

[Crearea unei funcţii pentru Access 31](#_Toc117777055)

[Cap. 11. Luarea deciziilor în cod 31](#_Toc117777056)

[Blocuri if 32](#_Toc117777057)

[Blocuri Select Case: 33](#_Toc117777058)

[Cap. 12. Utilizarea ciclurilor pentru acţiuni repetitive 33](#_Toc117777059)

[Cicluri For… 33](#_Toc117777060)

[Cicluri For… Next 33](#_Toc117777061)

[Cicluri For Each… Next 35](#_Toc117777062)

[Cicluri Do 35](#_Toc117777063)

[Cicluri While.. Wend 35](#_Toc117777064)

[Partea 4. UTILIZAREA MESSAGE BOXES, INPUT BOXES ŞI DIALOG BOXES 36](#_Toc117777065)

[Cap. 13. Preluarea datelor de la utilizator cu Message Boxes şi Input Boxes 36](#_Toc117777066)

[Afişarea mesajelor barei de stare în Word şi Excel 36](#_Toc117777067)

[Message Boxes 36](#_Toc117777068)

[Input Boxes 37](#_Toc117777069)

[Cap. 14. Crearea de căsuţe de dialog (Dialog Boxes) personalizate simple 37](#_Toc117777070)

[Crearea unei căsuţe de dialog 37](#_Toc117777071)

[Proprietăţi comune ale controalelor din Toolbox: 39](#_Toc117777072)

[Label 40](#_Toc117777073)

[Textbox 40](#_Toc117777074)

[ComboBox şi ListBox 42](#_Toc117777075)

[CheckBox 42](#_Toc117777076)

[OptionButton 42](#_Toc117777077)

[ToggleButton 42](#_Toc117777078)

[Frame 43](#_Toc117777079)

[CommandButton 43](#_Toc117777080)

[TabStrip şi MultiPage 44](#_Toc117777081)

[ScrollBar şi SpinButton 44](#_Toc117777082)

[Image 44](#_Toc117777083)

[Page 44](#_Toc117777084)

[Tab 44](#_Toc117777085)

# Steve McDonald, Gillian Werner „Microsoft Excel VBA for Beginners Learn VBA Step by Step” [Udemy] (2022)

* Activarea tab-ului Developer în Microsoft Excel 2019: meniul File > Options > Customize Ribbon > Choose Commands > All Tabs > Main Tabs: selectarea tab-ului Developer > OK
* Inserarea tab-ului Developer în Ribbon
* Activarea tab-ului Developer în Microsoft Excel 2007: Excel Options > Popular: selectarea căsuţei Show Developer tab in the Ribbon
* Salvarea fişierului Excel ca un fişier Excel cu macro-uri: meniul File > Save As > Save as type > Excel Macro-Enabled Workbook
* Iconiţă diferită – cu un semn suplimentar în dreapta jos
* Fişierul nu este partajabil (shareable)
* Tab-ul Developer are patru secţiuni:

1. Code
2. Add-ins
3. Controls
4. XML

* Deschiderea ferestrei Visual Basic: tab-ul Developer > secţiunea Code > Visual Basic (Alt + F11)
* Fereastră cu trei zone:

1. Project Explorer
2. Properties
3. Zona pentru cod sau interfeţe grafice (Forms)

* Crearea unei subroutine: tab-ul Developer > secţiunea Code > Visual Basic (Alt + F11) > meniul Insert > Module
* Public Sub Example()

End Sub

* Private Sub PrivateExample()

End Sub

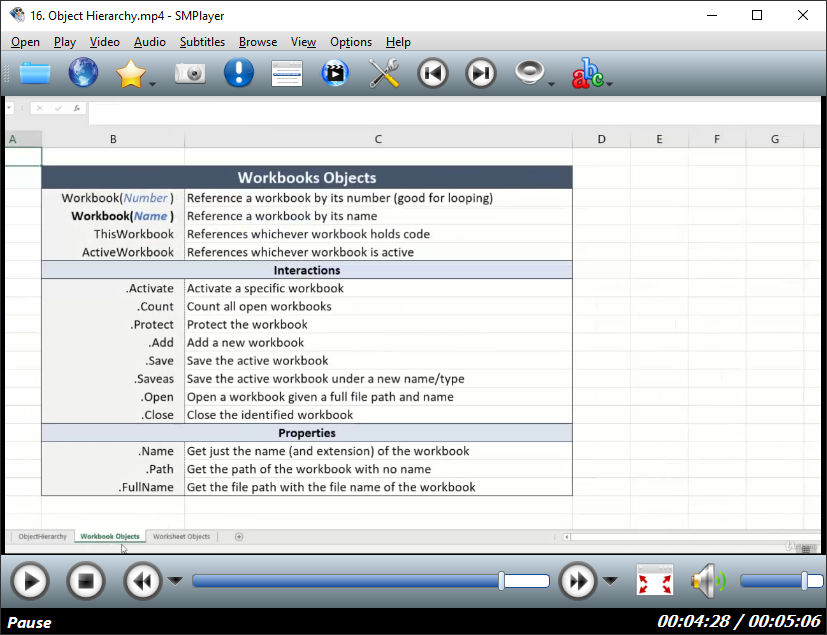
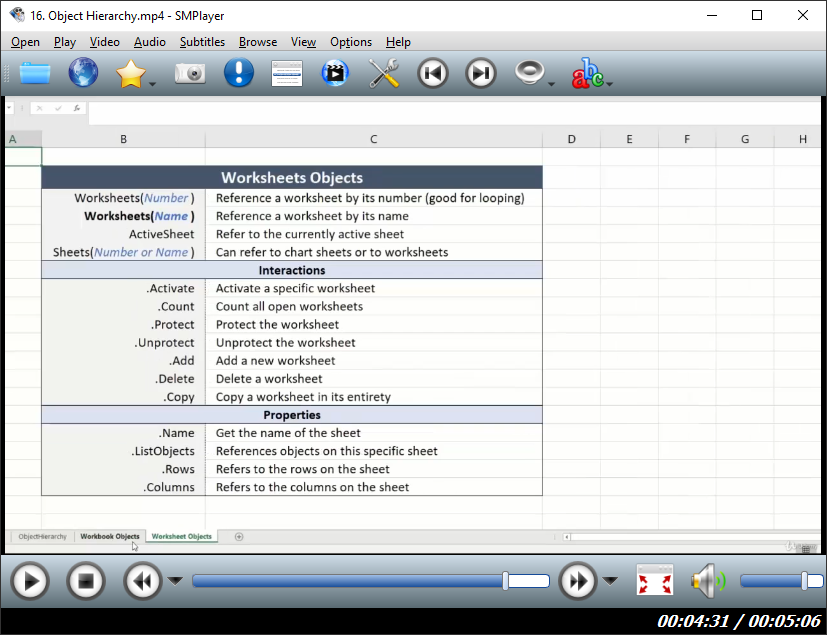
* Subrutinele publice pot fi apelate din alte subrutine sau din alte fişiere Excel. Sunt vizibile prin tab-ul Developer > secţiunea Code > Macros.
* Subrutinele private pot fi apelate numai din subrutine din modulul propriu.
* Cuvintele cheie din cod sunt colorate în albastru, comentariile în verde, iar celelalte în negru.
* Comentariile încep cu un apostrof.
* VBA nu doreşte scrierea de cod atunci când o celulă din Excel este activă (are cursorul de editare în interior) – trebuie deselectată.
* Ieşirea forţată dintr-o subrutină: Exit Sub
* Tipuri de variabile în VBA:
* numele de variabile încep cu o literă, pot conţine litere, cifre şi liniuţa de subliniere, dar nu şi spaţiiu
* numerice: Byte, Integer, Long, Single, Double, Currency
* nenumerice: String, Boolean, Date, Object, Variant

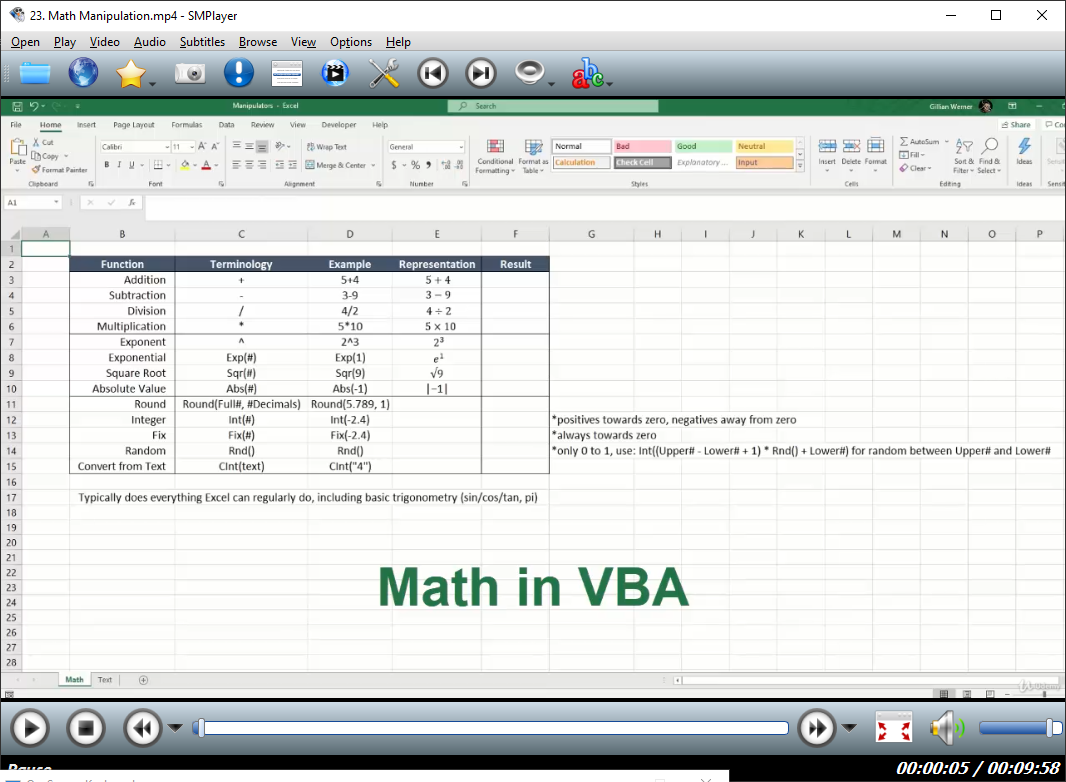
* Declararea unei variabile:
* explicită: Dim ExVar As Integer – recomandat
* implicită: ImVar = 2
* variabila devine de tipul Variant – ineficient
* Impunerea declarării explicite a unei variabile: meniul Tools > Options > Editor > selectarea căsuţei Require Variable Declaration
* Afişarea valorii unei variabile într-o celulă:
* Cells(2, 1).Value = ExVar
* Cells(2, 2).Value = ImVar
* Metode de declarare:
* globale: în afara unei subrutine - Public GlobalVar As Integer
* locale: într-o subrutină - Dim ExVar As Integer
* Ferestre de dialog:
  + MsgBox "alfa"
  + MsgBox "alfa", , "Example"
  + MsgBox "alfa", vbYesNo, "Example"
* Ferestre de preluare date: InputBox("Care este culoarea ta preferată?", "Culoare")
* Dezactivarea mesajelor de alertă: Application.DisplayAlerts = False
* Activarea mesajelor de alertă: Application.DisplayAlerts = True
* Concatenare: &
* Introducere linie nouă într-un text afişat: vbCrLf
* Tablouri în VBA:
* statice: Dim students(0 To 4) As String
* dinamice: Dim students() As String
  + - ReDim students(0 To 4) => se reiniţializează toate valorile atribuite iniţial la valori nule
    - ReDim Preserve students(0 To 4) => se păstrează toate valorile atribuite iniţial
* atribuire valori: students(0) = “Joe”
* limita superioară a tabloului (indicele maxim) : UBound(students)
* Controale pentru o interfaţă grafică (GUI: Graphical User Interface): tab-ul Developer > secţiunea Controls > Insert > Form Controls
* butonul se ataşează unei subrutine (macro)
* Controale pentru o interfaţă grafică (GUI: Graphical User Interface): tab-ul Developer > secţiunea Controls > Insert > ActiveX Controls
* Inserarea unei forme grafice (UI – User Form): tab-ul Developer > secţiunea Code > Visual Basic > meniul Insert > UserForm
* Obiecte şi ierarhia obiectelor:

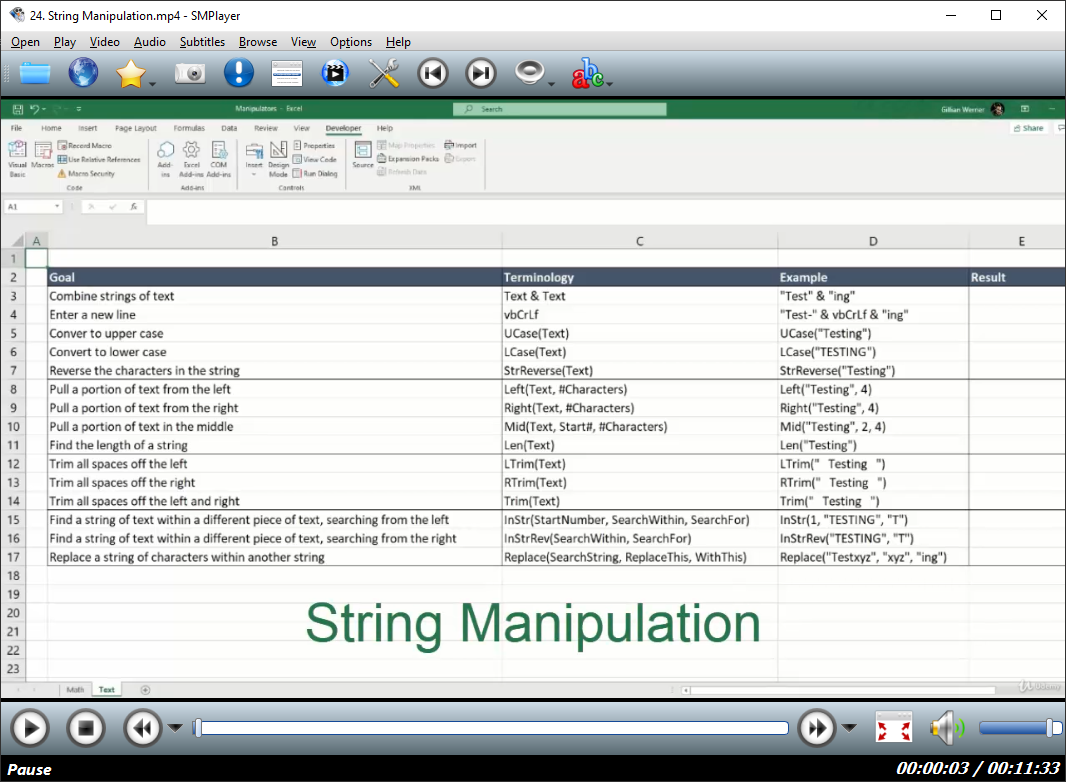
1. Application
2. Workbooks(Numele registrului de lucru)
3. Worksheets(Nume sau număr)
4. Ranges, Cells, Tables

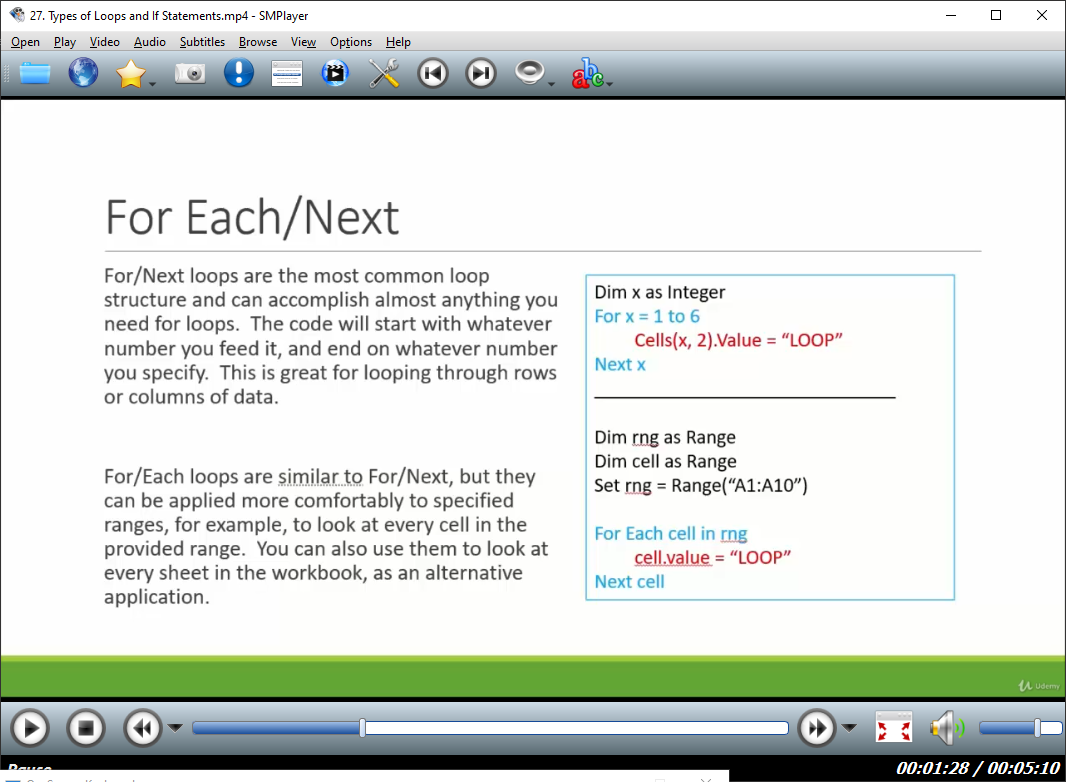
Ex: referinţa la celula H2, din foaia de lucru Sheet1, din registrul de lucru Macros

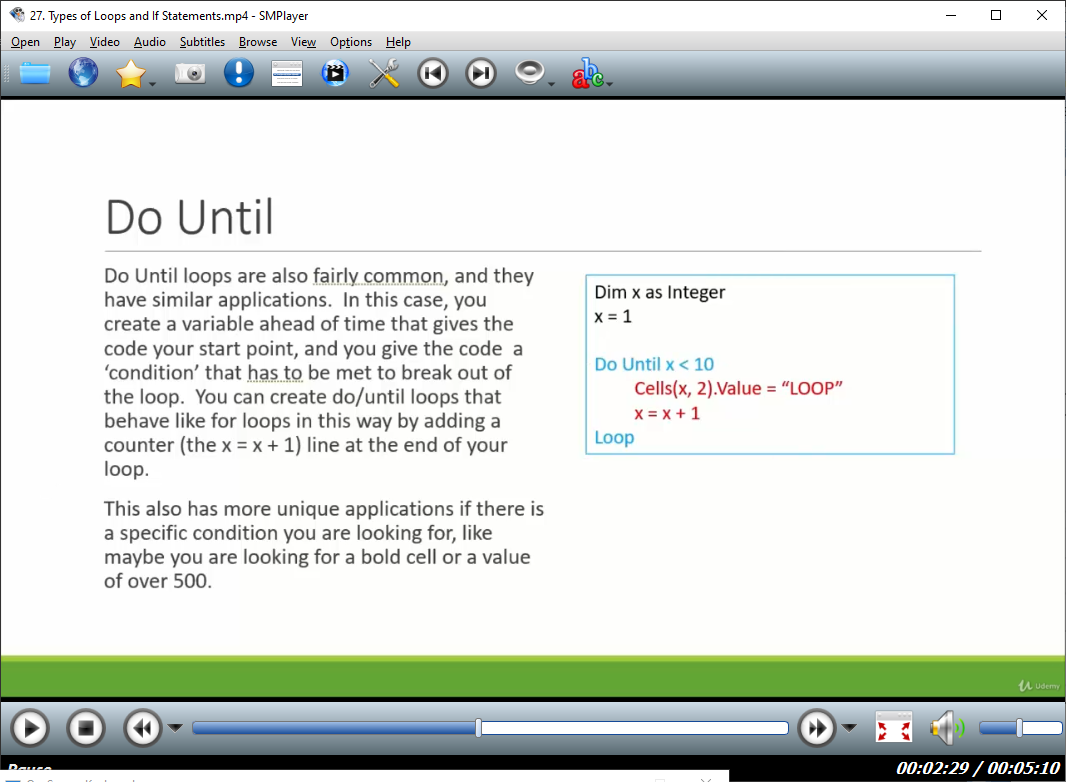
Application.Workbooks(“Macros”).Worksheets(“Sheet1”).Cells(2, 8)

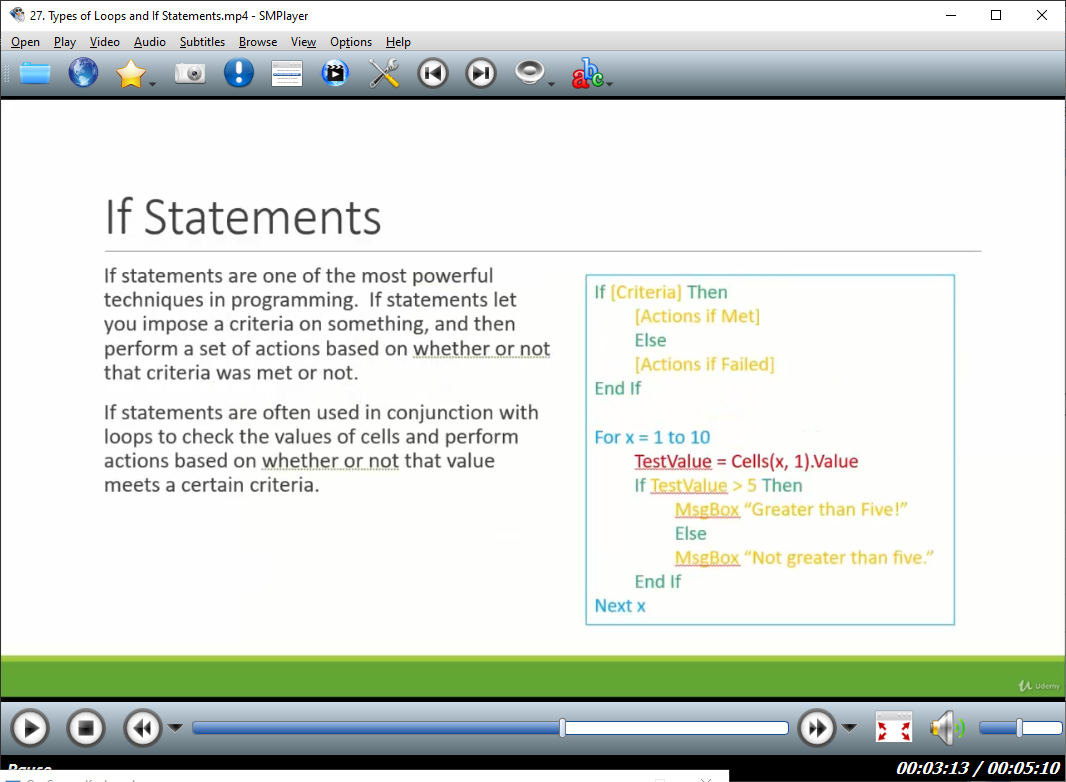
 

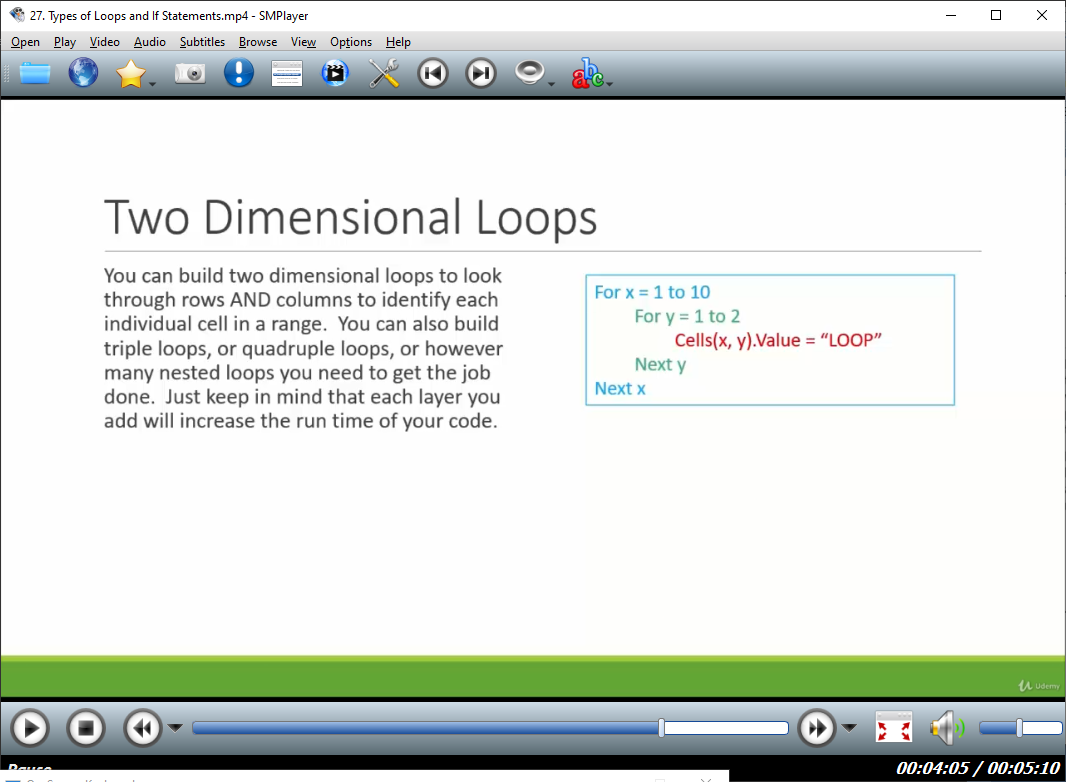


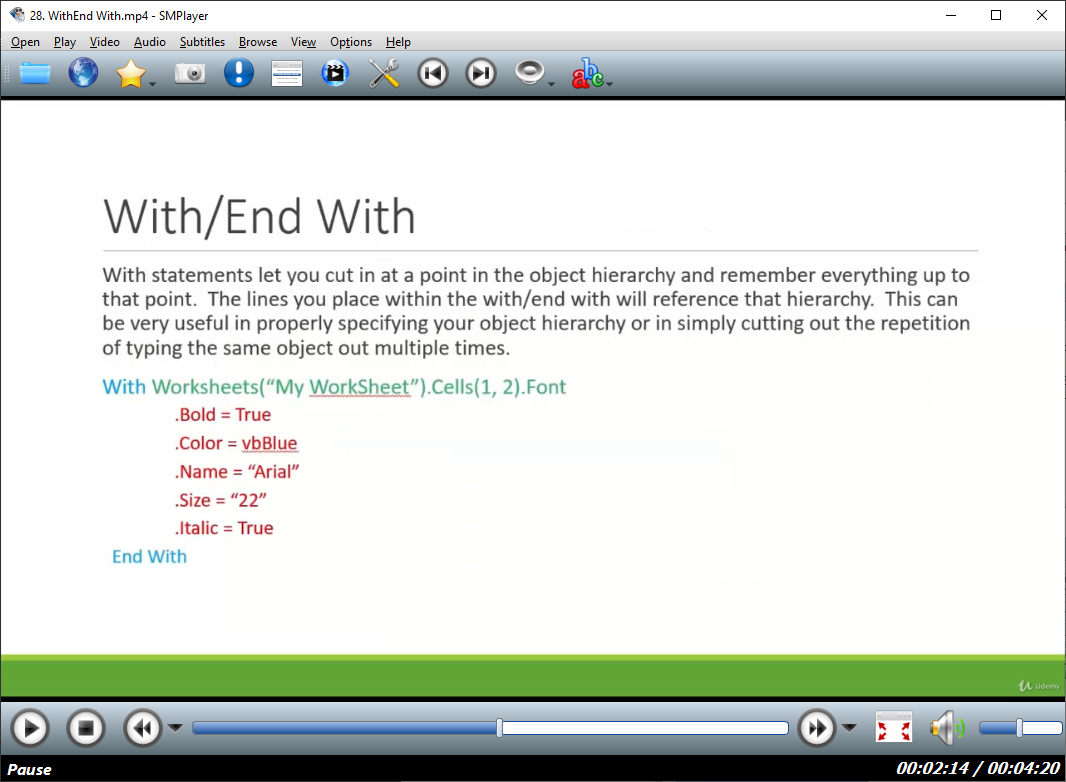












* Recomandări pentru creşterea vitezei de execuție:
  + Oprirea calculării automate în timpul rulării:
    - Application.Calculating = False – la începutul codului
    - Application.Calculating = True – înainte de terminarea codului
  + Oprirea actualizării automate a ecranului:
    - Application.ScreenUpdating = False – la începutul codului
    - Application.ScreenUpdating = True – înainte de terminarea codului
  + Dezactivarea barei de stare:
    - Application.DisplayStatusBar = False – la începutul codului Application.DisplayStatusBar = True – înainte de terminarea codului
  + Dezactivarea evenimentelor:
    - Application.EnableEvents = False – la începutul codului
    - Application.EnableEvents = True – înainte de terminarea codului
  + Minimizarea interacţiunii cu fişierul Excel şi încercarea de a muta sau citi date în blocuri de date cu funcţii Range pentru a referi întregul domeniu de date sau un tabel întreg de date – se pot converti într-un tablou, tablou care se află în cod şi minimizează interacţiunea cu fişierul.
  + Utilizarea inteligentă a ierarhiei de obiecte şi folosirea avantajului with/end with pentru a evita parcurgerea prin ierarhie şi activarea foilor de lucru în mod repetat.
  + Evitarea folosirii “WorksheetFunction” pentru a referi funcţiile foilor de lucru interne din Excel – este mult mai rapid de a codifica în VBA, a.î. nu trebuie accesat fişierul Excel.
  + Evitarea variabilelor de tip Variant.
  + Evitarea copy-paste, înlocuirea cu setarea egalităţii unui domeniu cu altul.
* Funcţii particularizate
  + Similare cu subrutinele private, cu deosebirea principală că funcţiile returnează o valoare.
  + Se pot apela dintr-o celulă din Excel.
* Evenimente particularizate
  + BeforeDoubleClick

# Boris Paskhaver „Excel VBA Programming The Complete Guide” [Packt] (2022)

# Richard Mansfield „Mastering VBA for Microsoft® Office 365® (2019 Edition)” [Sybex] (2019).epub

## Partea 1. ÎNREGISTRAREA MACRO-URILOR ŞI ÎNCEPEREA UTILIZĂRII VBA

### Cap. 1. Înregistrarea şi rularea macro-urilor în aplicaţiile Office

VBA lucrează întotdeauna în interiorul unei aplicaţii Office – nu se pot construi aplicații de sine-stătătoare cu VBA.

Excel: ActiveWorkbook

Word: ActiveDocument

PowerPoint: ActivePresentation

Visual Basic Express: o versione gratuită de VB .NET:

https://www.visualstudio.com/en-us/products/visual-studio-express-vs.aspx

Manipularea SO Windows:

https://www.autohotkey.com/

Tab-ul Developer: în Word, Excel, PowerPoint şiu Outlook.

Word, dar nu şi Excel, permite oprirea temporară a lui Macro Recorder pentru a face ceva ce nu se doreşte înregistrat.

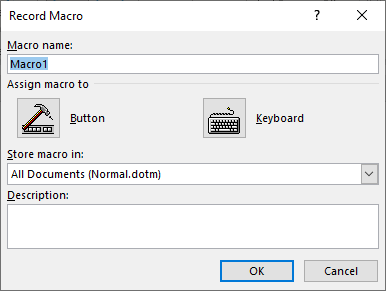
Trust Center: tab-ul Developer > Macro Security > Macro Settings > alegere Disable All Macros With Notification şi selectarea Trust access to the VBA project object model.

Securitatea macro-urilor în Office 2019:

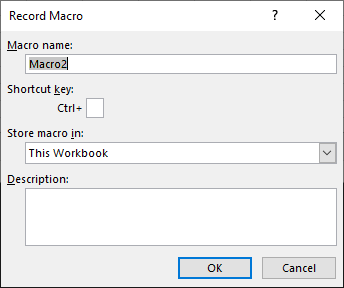
https://support.office.com/en-us/article/enable-or-disable-macros-in-office-files-12b036fd-d140-4e74-b45e-16fed1a7e5c6

Înregistrare macro-uri: tab-ul View > Macros sau tab-ul Developer > Record < Macro

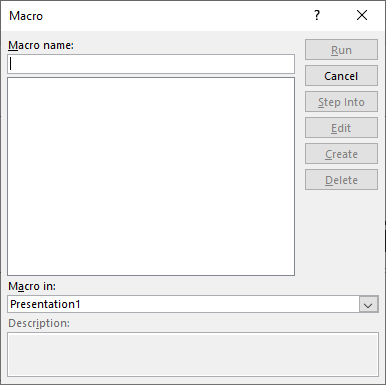
- Word:



- Excel:



- PowerPoint – nu are tab-ul Developer:



Salvare macro-uri:

- Word:

- pentru uz global, cu toate fişierele Word: se stochează într-un fişier „Normal.dotm”;

- pentru uz în fişierul activ curent: fişier cu extensia „.dotm”.

- Excel:

- pentru uz global, cu toate fişierele Excel: în Excel Personal Macro Workbook, un fişier cu numele „Personal.xlsb”, un fişier ascuns;

- în registrul de lucru curent;

- într-un registru de lucru nou.

În versiunea PowerPoint 2019 nu se pot înregistra macro-uri, dar se pot crea prin scrierea codului de programare în Visual Basic Editor. Apoi, macro-urile se pot stoca în prezentarea activă curentă sau în orice altă prezentarea sau şablon.

PowerPoint furnizează un container de stocare a unui macro global, similar cu „Normal.dotm” – se găseşte prin tab-ul Developer > Macros > Macro in > All Open Presentations.

Nume macro-uri:

- să înceapă cu o literă;

- maxim 80 caractere;

- pot conţine litere, cifre şi liniuţa de subliniere;

- nu pot conţine spaţii, semne de punctuaţie sau caractere speciale.

Adăugarea unui macro la Ribbon: click dreapta pe Ribbon > Customize the Ribbon:

- Choose commands from: Macros > selectarea macro-ului

- selectarea unui tab în partea dreaptă şi crearea unui grup nou în care va fi adăugat macro-ul.

- butonul Add >>.

Adăugarea unui macro la bara de instrumente Quick Access: click dreapta pe bara de instrumente Quick Access > Customize Quick Access Toolbar:

- Choose commands from: Macros > selectarea macro-ului

- butonul Add >>.

Asignarea unei combinaţii de taste pentru un macro (Wiord): click dreapta pe Ribbon > Customize the Ribbon:

- Choose commands from: Macros > selectarea macro-ului

- Keyboard Shortcus > Customize.

Comentarii: încep cu un apostrof: ‘.

Arhitectura Word pe patru nivele:

- aplicaţia însăşi – aplicaţia Word, nivel activ întotdeauna;

- şablonul global (Normal.dotm) - nivel activ întotdeauna;

- şablonul documentului activ – nivelul este activ numai dacă documentul curent sau activ este ataşat la alt şablon decât cel global;

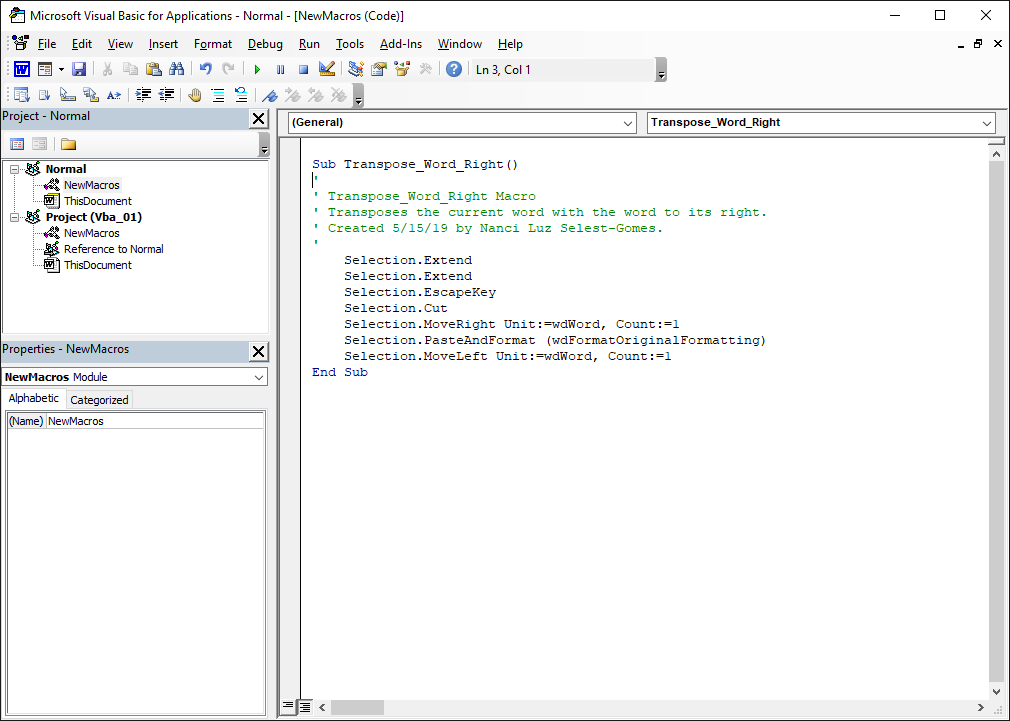
- documentul activ el însuşi – este prezent întotdeauna când este deschis documentul.

Comanda „winword /n” previne ca macro-urile din şablonul global să fie active.

### Cap. 2. Visual Basic Editor

Deschidere:

- selectarea unui macro existent şi editarea lui:



- deschiderea directă a lui Visual Basic Editor: Alt + F11 sau tab-ul Developer > Visual Basic.

Vizualizarea ferestrei Properties: F4.

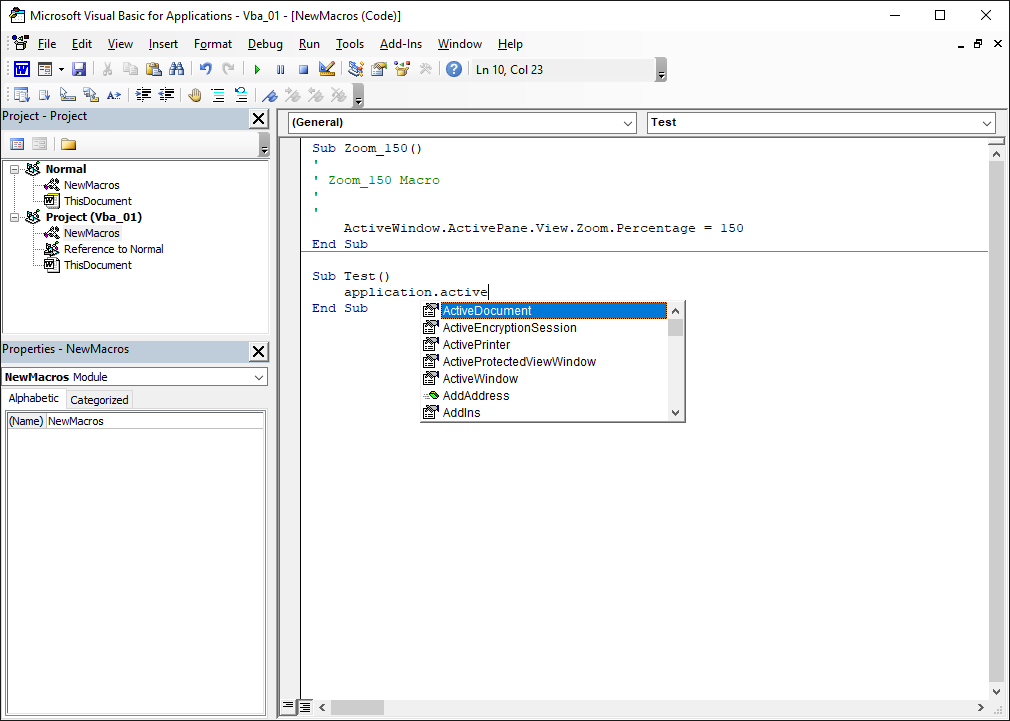
Macro-urile din şablonul Normal, Normal.dotm, se stochează în modulul „NewMacros”.

Se pot utiliza simultan mai multe instanţe ale lui Visual Basic Editor – una pentru Word, alta pentru Excel, alta pentru Access.

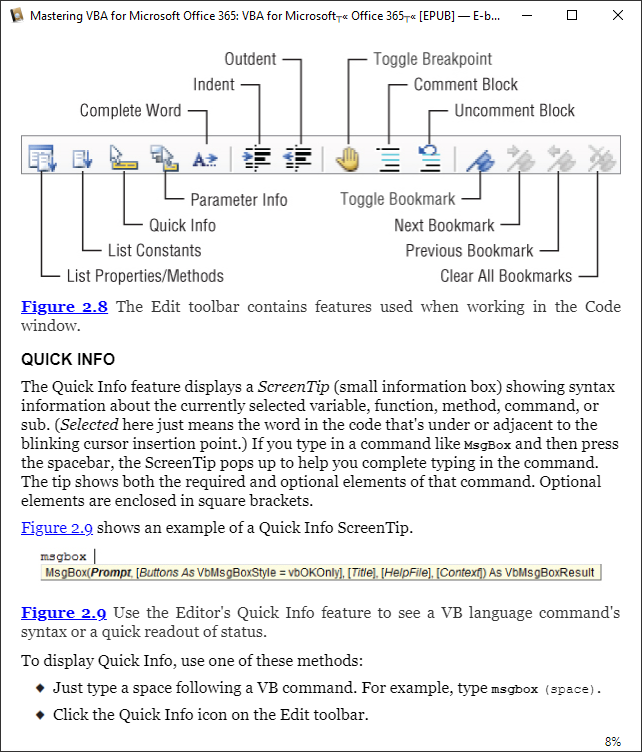
Caracteristica Complete Word poate completa un cuvânt care se tipăreşte în fereastra Code odată ce s-au tipărit destule litere pentru a distinge acel cuvânt de altul.

Cel mai uşor mod de a activa Complete Word în timpul tipăririi este Ctrl + bara de spaţiu.

Altfel, tab-ul Edit > Complete Word.

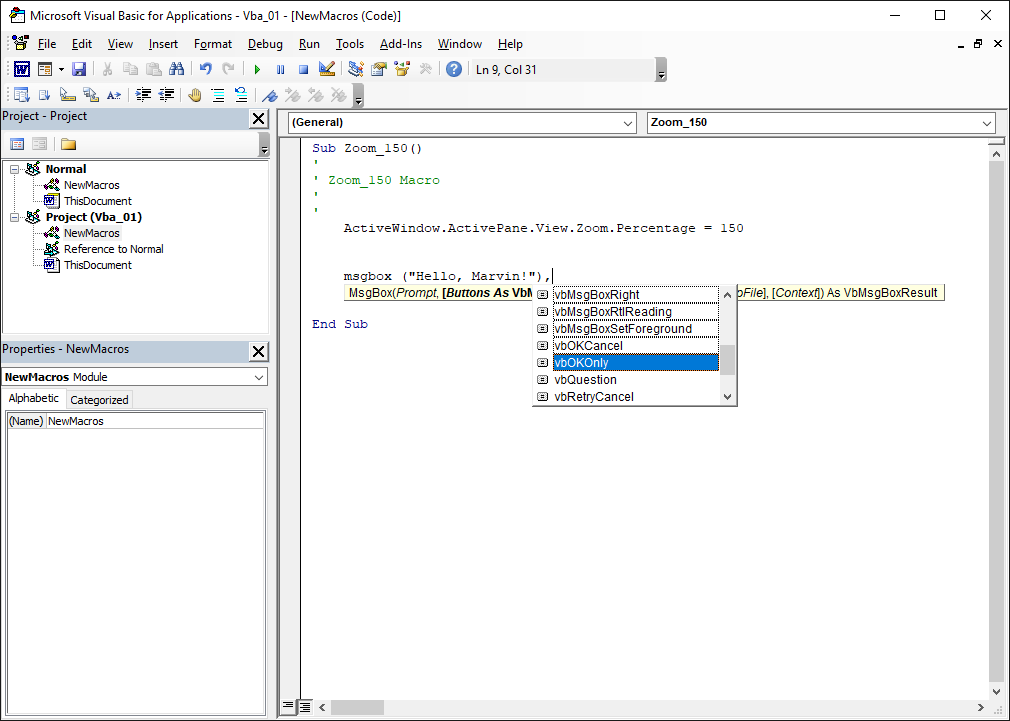


Afişarea barei de instrumente Edit: tab-ul View > Toolbars > Edit.



VBA este case-insensitiv.

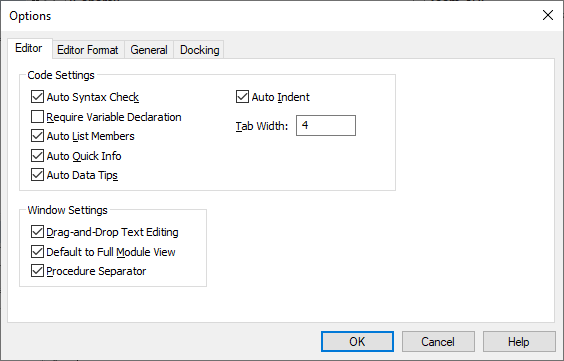
Lista Auto List Memnbers permite completarea rapidă a liniei de cod. Este comutată implicit şi este afişată automat când se tipăreşte un punct în descrierea unui obiect sau o virgulă, paranteză sau alt semn de punctuaţie într-o linie de cod.



Caracteristica List Constants afişează o casetă listă pop-up conţinând constantele pentru o proprietate care a fost tipărită. Este comutată automat sau prin butonul List Constants din bara de instrumente Edit.



Caracteristica Data Tips afişează un text ScreenTip care conţine valoarea unei variabile în timp ce pointer-ul de mouse se deplasează peste ea atunci când Visual Basic Editor se află în modul Break. Caracteristica este comutată automat sau prin Tools > Options.



Moduri Visual Basic Editor:

- Design mode: lucrul în codul sursă;

- Run mode (runtime): rularea codului – F5;

- Break mode: codul rulează, dar execuţia este suspendată temporar şi se poate executa pas cu pas.

Ferestrele interioare (panouri – panes) sunt fixate implicit, dar se pot elibera sau fixa la loc prin dublu click pe bara de titlu.

Fereastra Immediate:

- se poate folosi ca o zonă virtuală pentru a introduce cod de testare fără a-l introduce în codul macro-ului actual;

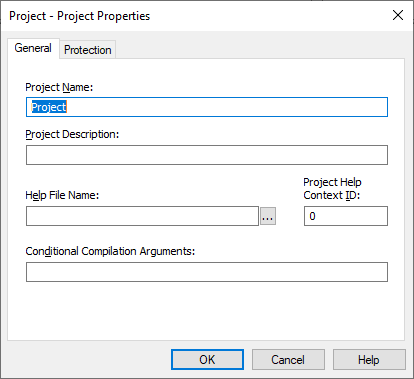
- meniul View > Immediate Window;

- Ctrl + G;

- afişează rezultatul comenzii Debug.Print folosită în cod la rularea lui.

Biblioteca de tip (type library) descrie obiectele pe care le conţine proiectul (module, forme utilizator).

Numele proiectului este numele bibliotecii de tip – poate conţine liniuţe de subliniere, dar nu şi spaţii.



Orice personalizare în Visual Basic Editor se răsfrânge în editor la toate instanţele găzduite de Word, Excel, PowerPoint, Outlook.

### Cap. 3. Editarea maacro-urilor înregistrate

### Cap. 4. Crearea codului de la zero în Visual Basic Editor

Setarea Visual Basic Editor pentru a crea macro-uri:

- afişarea Project Explorer: View > Project Explorer sau Ctrl + R;

- afişarea ferestrei Properties: View > Properties Windows aues Windows sau F4;

- forţarea cerinţei ca variabilele să fie declarate explicit: Tools > Options > Editor > selectarea căsuţei Require Variable Declaration.

Word:

- deschidere fişier nou;

- deschidere Visual Basic Editor;

- în Project Explorer, click dreapta în itemul Normal şi alegere Insert > Module;

- Visual Basic Editor adaugă un modul nou în şablonul global Normal.dotm şi deschide fereastra Code pentrui el;

- tastarea tastei F4 pentru a activa fereastra Properties pentru noul modul;

- Visual Basic Editor selectează proprietatea (Name), singura proprietate disponibilă pentru acest nou modul, şi redenumirea lui;

- tastarea tastei F7 pentru activarea ferestrei Code;

- crearea unei proceduri Sub;

Excel:

- deschidere fişier nou;

- tab-ul View > Unhide;

- deschidere Personal.xlsb (C:\Users\marius\AppData\Roaming\Microsoft\Excel\XLSTART);

- numele Auto\_Open pentru un macro este special – VBA ştie că acţiunile din codul macro-ului sunt executate atunci când Excel porneşte rularea – evenimentul Open.

PowerPoint:

Access:

- nu are tab-ul Developer;

- Visual Basic Editor se lansează din tab-ul Database Tools.

## Partea 2. ÎNVĂŢAREA UTILIZĂRII VBA

### Cap. 5. Înţelegerea lucrurilor esenţiale ale sintaxei VBA

Proceduri:

- între Sub şi End Sub;

- o colecţie de proceduri este conţinută într-un modul.

Funcţii:

- între Function şi End Function;

- returnează o valoare.

Declaraţii (statements):

- de obicei, una pe linie;

- mai multe pe linie, separate prin două puncte „:”;

- una pe mai multe linii, separate prin liniuţa de subliniere, „\_”, precedată de un spaţiu;

- şirurile de caractere cuprinse între ghilimele nu pot fi despărţite prin caracterul de continuare a liniei \_.

Obiecte Word:

- ActiveDocument: documentul activ curent, cel care este vizibil;

- colecţia Documents: reprezintă toate documentele deschise;

- Selection: reprezintă selecţia curentă din interiorul unui document sau doar locaţia cursorului de inserare, dacă nu este selectat nimic.

Ex:

Documents.Open "C:\Users\marius\Documents\work\vba\vba\_01.docm"

MsgBox ActiveDocument.Name

ActiveDocument.Close SaveChanges:=wdDoNotSaveChanges

Documents.Add

Application.Quit

Cuvinte cheie

Expresii

Operatori

Variabile

Constante

Argumente

Literali

Obiecte

Colecţii

Proprietăţi

Metode

Evenimente

### Cap. 6. Lucrul cu variabile, constante şi enumerări

**1. Variabile:**

- notaţia Camel:



- dacă numele unei variabile începe cu un număr, VBA consideră că se doreşte notarea rândurilor în Visual Basic Editor şi că acel număr reprezintă numărul rândului:

20alfa=”papa” => 20 alfa = “papa”

10msgbox “Alfa” => 10 MsgBox “Alfa”

- declarare implicită: printr-o simplă atribuire a unei valori, variabila având implicit tipul Variant

- declarare de tip caracter: se adaugă un caracter la sfârşitul numelui unei variabile într-o declaraţie implicită pentru a comunica VBA ce tip de dată să utilizeze pentru variabilă:

-- % Integer

-- & Long

-- @ Currency

-- ! Single

-- # Double

-- $ String

- declarare explicită:

-- Dim | Private | Public | Static nume\_variabilă => tip Variant

-- Dim | Private | Public | Static nume\_variabilă as Integer | Long | Currency | Single | Double | String | …

-- se pot declara mai multe variabile pe aceeaşi linie prin separarea declaraţiilor cu o virgulă.

Domeniul de valabilitate pentru o variabilă:

- Procedure (local) => spaţiu Small

- Private (modul) => spaţiu Medium

- Public (proiect) => spaţiu Large

Toate variabilele declarate implicit sunt locale.

Variabilele declarate cu Static într-o procedură au domeniul de valabilitate local, dar durata de viaţă ca cele private sau publice. De obicei, la comutarea (toggle) între două valori într-o procedură.

Comanda „Not” comută o variabilă logică (booleană).

Lista de prefixe uzuale:

<http://en.wikipedia.org/wiki/Leszynski_naming_convention>

Se recomandă folosirea numerelor în virgulă fixă în loc de cele în virgulă flotantă.

Datele de tip Date pot fiu introduse ca literali între două caractere #.

La datele calendaristice se specifică secolul întotdeauna (1999 sau 2009).

Ex:

#June 30, 1936# => 6/30/36

#10:15PM# => 10:15:00 PM

Variabilele şir de caractere de lungime fixă se declară cu un \*:

Ex:

Dim strStateAbbreviation As String \* 2

Variabilele Variant pot conţine patru valori speciale:

- Empty;

- Error;

- Nothing;

- Null.

Dacă nu se ştie sigur ce tip de variabilă va conţine cel mai bine informaţia, se poate porni prin declararea iniţială a variabilei ca Variant şi vizualizarea procedurii în modul Break cu Locals Window afişată (View > Locals Window).

**2. Constante: unele sunt încorporate în VBA.**

[Public | Private] Const nume\_constantă [As tip] = expresie

**3. Enumerare: listă numerotată de itemi, cu numere întregi care au nume individuale.**

Numele care corespund întregilor din enumerare se numesc „constante enumerate”.

VBA include multe enumerări.

AutoExec este numele unei proceduri încorporate în VBA care rulează automat atunci când porneşte Word.

### Cap. 7. Utilizarea variabilelor tablou (Array)

Tablouri cu indexul de la 1: Option Base 1 în secţiunea General Declarations, laolaltă cui Option Explicit.

Declarare: cu paranteze rotunde după numele tabloului.

Dacă în paranteze este un număr, el reprezintă ultimul index, dacă sunt două numere, ele sunt indecşii inferior şi superior.

Ex:

Dim alfa/()

Dim beta(10)

Dim gama(1 to 10)

Referirea unui element din tablou: tot cu paranteze rotunde.

Dacă parantezele sunt goale, este un tablou dinamic.

Redimensionarea unui tablou: ReDim.

Redimensionarea unui tablou cu păstrarea valorilor iniţiale: ReDim Preserve.

Ştergerea unui tablou: Erase.

Verificarea dacă o variabilă este un tablou: IsArray.

Aflarea marginilor unui tablou: LBound şi UBound.

### Cap. 8. Găsirea obiectelor, metodelor şi proprietăţilor necesare

Toate aplicaţiile gazdă VBA, inclusiv toate aplicaţiile Office majore, au un obiect „Application”, care reprezintă aplicaţia ca un întreg.

Obiectul Application însuşi are proprietăţi şi metode pentru lucruri care se aplică aplicaţiei ca un întreg. Multe aplicaţii au metoda „Quit” care iese din aplicaţie şi o proprietate „Visible” care controlează dacă aplicaţia este vizibilă sau nu.

Într-un model obiect obişnuit, obiectul Application în mod esenţial conţine toate celelalte obiecte (şi colecţii de obiecte) care alcătuiesc aplicaţia.

Ex:

- Excel: are un obiect Application care reprezintă aplicaţia Excel, un obiect Workbook, grupat în colecţia Workbooks, care reprezintă un registru de lucru, un obiect Worksheet, grupat în colecţia Sheets, care reprezintă o foaie de lucru, obiecte Row şi Column şi obiecte Range, care reprezintă domenii de celule.

Selectarea celulei A1 din prima foaie de lucru, Sheets(1), din primul registru de lucru, Workbooks(1), din aplicaţie:

Application. Workbooks(1). Sheets(1). Range(“A1”).Select

Majoritatea aplicaţiilor pun la dispoziţie un număr de obiecte care pot fi create, obiecte care sunt, de obicei, cele mai utilizate obiecte pentru aplicaţie şi, mergând prin ele, se poate accesa majoritatea celorlalte obiecte fără a mai fi necesară referirea la obiectul Application.

Oricum, obiectul Application este opţional şi, de obicei, este lăsat înafara codului, deoarece se presupune existenţa lui.

=> Workbooks(1). Sheets(1). Range(“A1”).Select

Colecţii:

- Word: Documents

- Excel: Workbooks

- PowerPoint: Presentations

Proprietăţi principale ale colecţiilor:

- Application

- Count

- Creator

- Item

- Parent

Ex. în fereastra Immediate Window pentru deschiderea fişierului Word VBA.docx:

?Documents.Application => Microsoft Word

?Documents.Count => 1

?Documents.Creator => 1297307460

?Documents(1).Name => VBA.docx

?Documents.Item(1) => VBA.docx

?Documents.Item(1).Name => VBA.docx

?Documents.Parent => Microsoft Word

Adăugarea unui obiect la o colecţie: add

?Documents.Add => deschiderea unui fişier Word nou.

Object Browser:

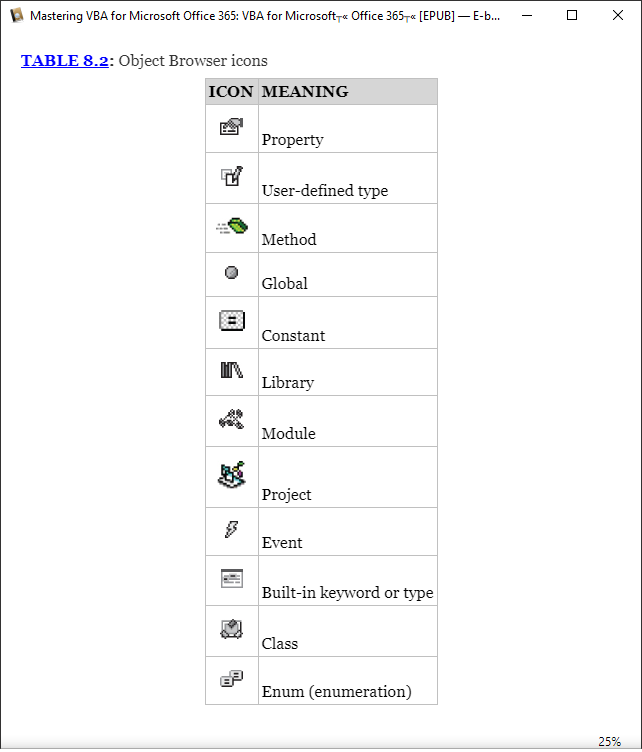
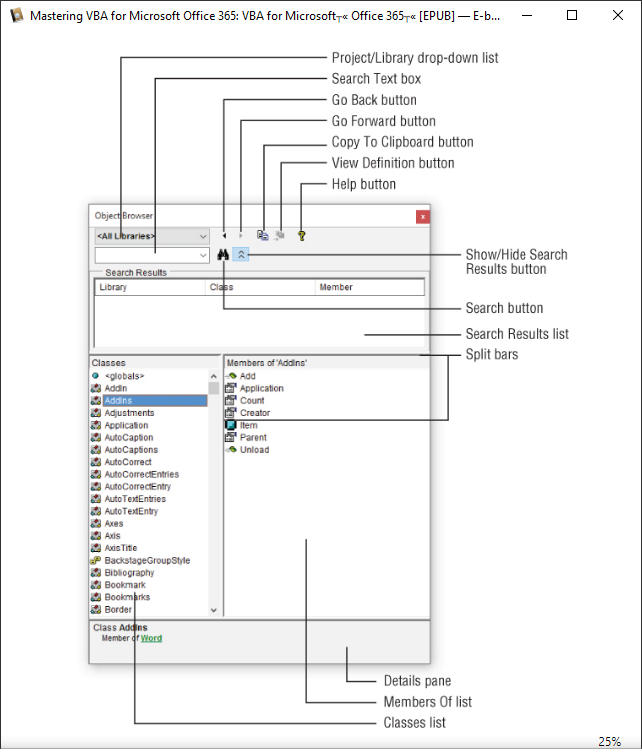
- clase;

- proprietăţi;

- metode;

- evenimente;

- constante.



Un punct albastru în colţul stânga-sus al iconiţei unei proprietăţi sau a unei metode indică faptul că acea proprietate sau metodă este implicită.

Adăugare şi ştergere biblioteci de obiecte în Visual Basic Editor: Tools > References



Asistenţă:

- tasta F1 cu cursorul într-o linie goală de cod va afişa portalul VBA.

- tasta F1 cu cursorul pe un cuvânt cheie din cod va afişa o pagină de Help cui informaţii despre acel cuvânt;

- suplimentar

<https://stackoverflow.com/questions/tagged/vba>

Utilizarea caracteristicii Auto List Members: porneşte de la un obiect.

Utilizarea variabilelor Obiect pentru a reprezenta obiecte:

- pentru accesul sau manipularea obiectului pe care îl reprezintă;

- face codul mai uşor de citit;

- sunt necesare atunci când se lucrează cu colecţii de obiecte;

- pot primi nume descriptive uşor de înţeles;

- se recomandă ca, atunci când se creează obiecte, acestea să fie numite, deoarece numele variabilelor Obiect nu se schimbă, dar numerele indecşilor se pot schimba.

Ex:

Dim objMyObject as Object

Set objMyObject = ActiveWorkbook - creare explicită

Set objMyObject = {[New] expresie|Nothing} - creare implicită

Set objMyObject = Nothing - eliberează memoria

## Partea 3. LUAREA DECIZIILOR ŞI UTILIZAREA CICLURILOR ŞI FUNCŢIILOR

### Cap. 9. Utilizarea funcţiilor încorporate

În limbajul BASIC, al cărui descendent este VBA, o funcţie diferă de o subrutină prin faptul că o funcţie returnează întotdeauna o valoare, pe când o subrutină, nu.

Utilizarea funcţiilor:

[Call] nume\_funcţie [listă\_argumente]

Lista de argumente este, de obicei, în paranteze.

Furnizarea mai multor argumente:

- prin valorile argumentelor, fără numele lor, poziţional;

- prin argumente cu tot cu numele lor, în orice ordine: nume:=valoare.

Omiterea unui argument opţional: printr-o virgulă, într-o listă de valori.

Conversii tipuri de valori:

- CBool(număr) => Boolean

- CByte(expresie) => Byte

- CCur(expresie) => Currency

- CDate(expresie) => Date

- CDec|CDbl|CInt|CLng|CSng(expresie) => Decimal|Double|Integer|Long|Single

- CStr(expresie) => String

- CVar(expresie) => Variant

- Hex(număr) => String cu valoarea hexazecimală a numărului

- Oct(număr) => String cu valoarea octală a numărului

- RGB(număr1, număr2, număr3) => Un întreg Long reprezentând valoarea specificată de cele 3 numere

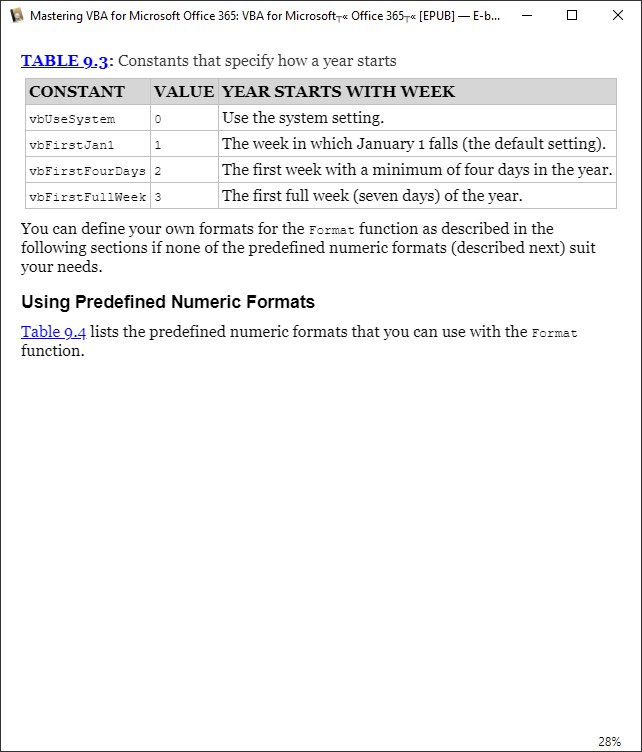
- QBColor(număr) => Un întreg Long conţinând valoarea RGB pentru culoarea specificată

- Str(număr) => Un Variant/String conţinând o valoare şir de caractere a numărului – rec. CStr

- Val(şir\_de\_caractere) => Porţiunea numerică a şirului; dacă nu are cifre, returnează 0 – rec. CInt

**Utilizarea funcţiei Format pentru a formata o expresie:**

Format(*expression*[, *format*[, *firstdayofweek*[, *firstweekofyear*]]])

****

Ex:

Format(*expression*[, *format*[, *firstdayofweek*[, *firstweekofyear*]]])

?Format(Now) => 10/23/2022 2:16:49 PM

Formate predefinite: General Number, Currency, Fixed, Standard, Percent, Scientific, Yes/No, True/False, On/Off.

Ex:

?Format(“12345”, “Currency”) => $12,345.00

?Format("12345", "Fixed") => 12345.00

?Format("12345", "Standard") => 12,345.00

?Format("12345", "Percent") => 1234500.00%

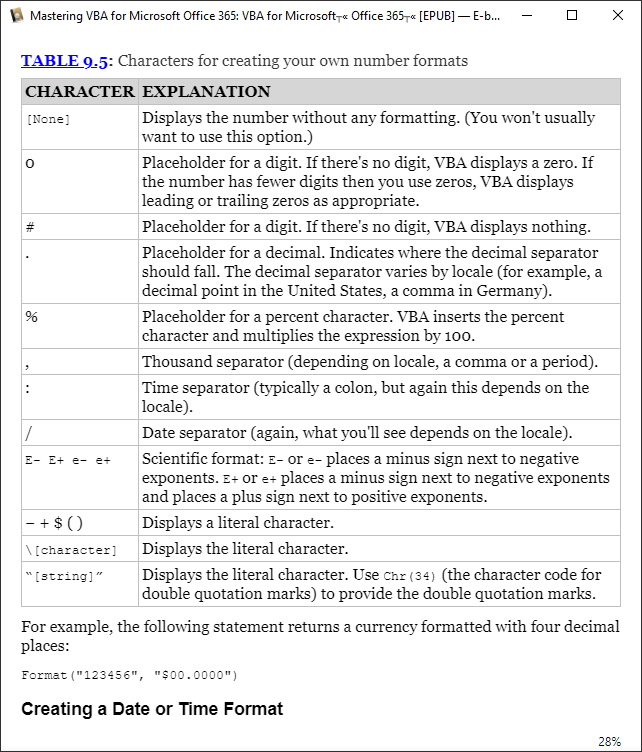
?Format("12345", "Scientific") => 1.23E+04

?Format("12345", "Yes/No") => Yes

?Format("12345", "True/False") => True

?Format("12345", "On/Off") => On

#### Crearea unui format numeric:



Ex:

?Format("123456") => 123456

?Format("123456", "00.0000") => 123456.0000

?Format("123456", "##.####") => 123456.

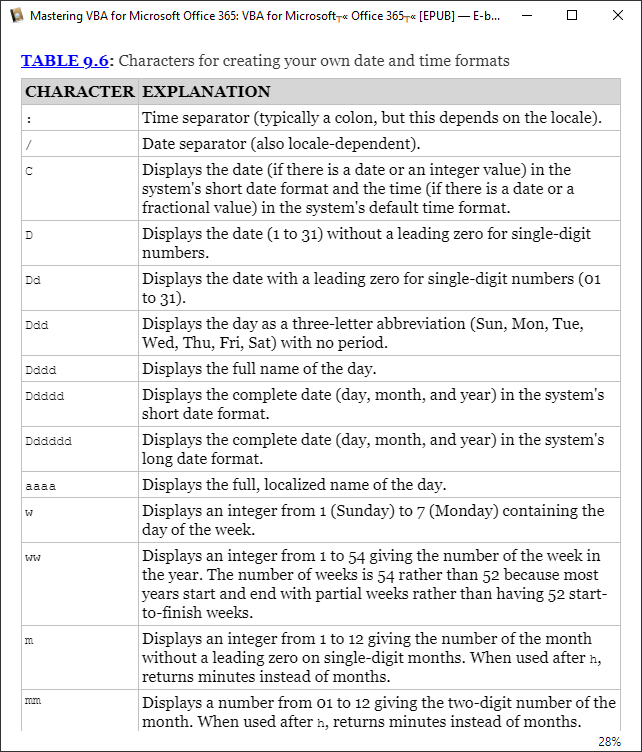
?Format("123456", "0%") => 12345600%

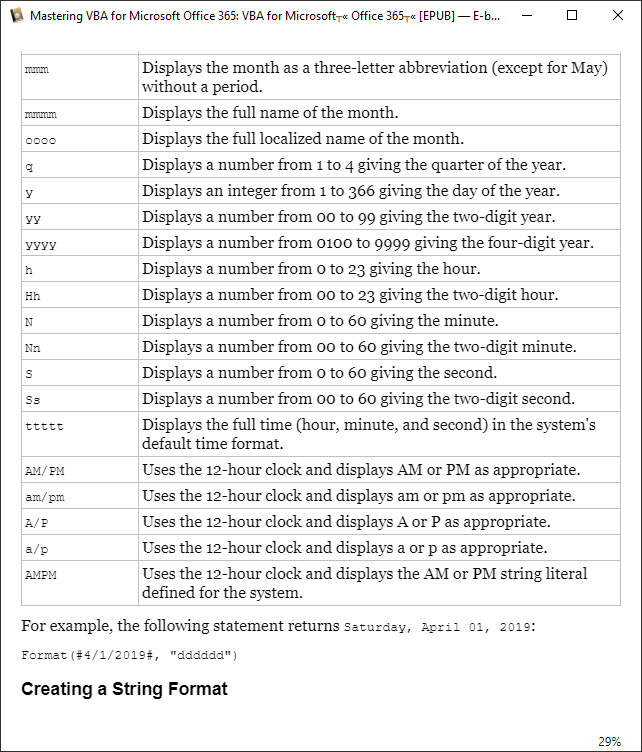
?Format("123456", "0,0") => 123,456

?Format("123456", "0.000E-") => 1.235E5

?Format("123456", "\N\=0,0") => N=123,456

#### Crearea unui format de dată calendaristică sau de timp:





Ex:

?Format(Now, "dddddd") => Sunday, October 23, 2022

#### Crearea unui format şir de caractere:

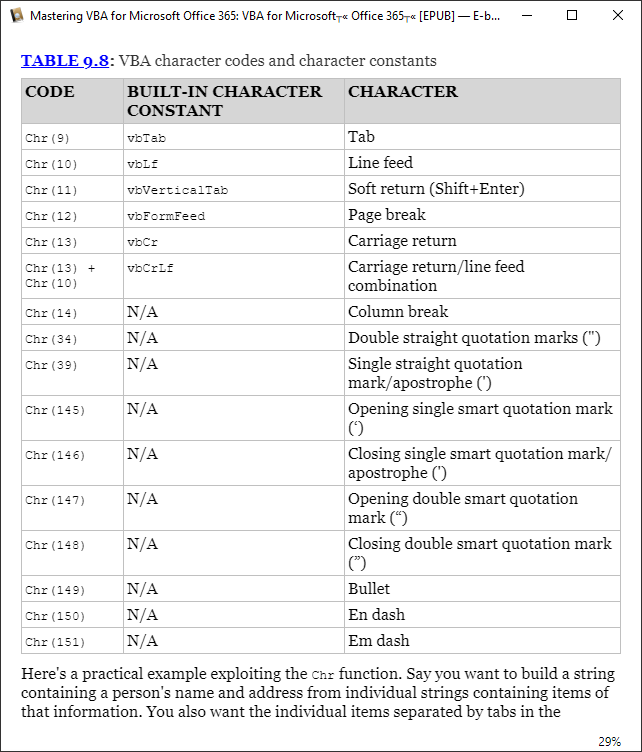


Ex.

?Format("PAPA", "<") => papa

?Format("alfa", ">") => ALFA

#### Utilizarea funcţiei Chr şi a constantelor pentru a introduce caractere speciale într-un şir de caractere:



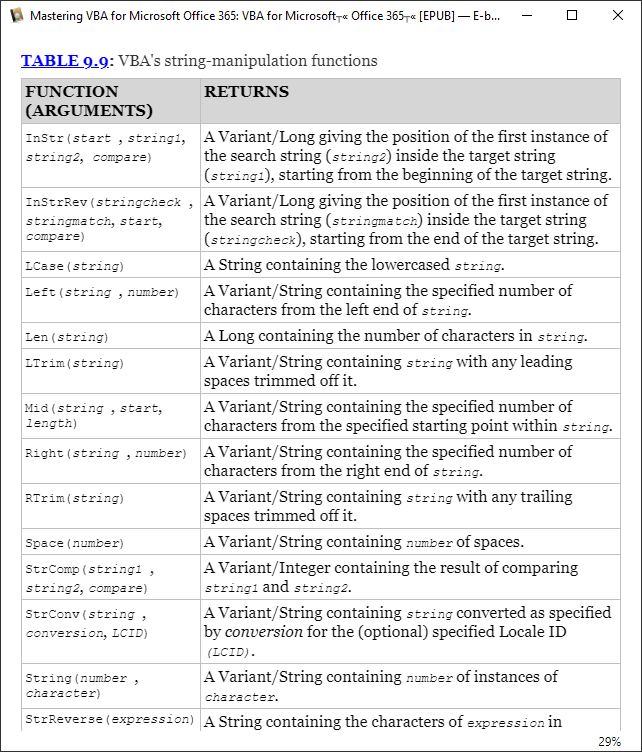
Ex:

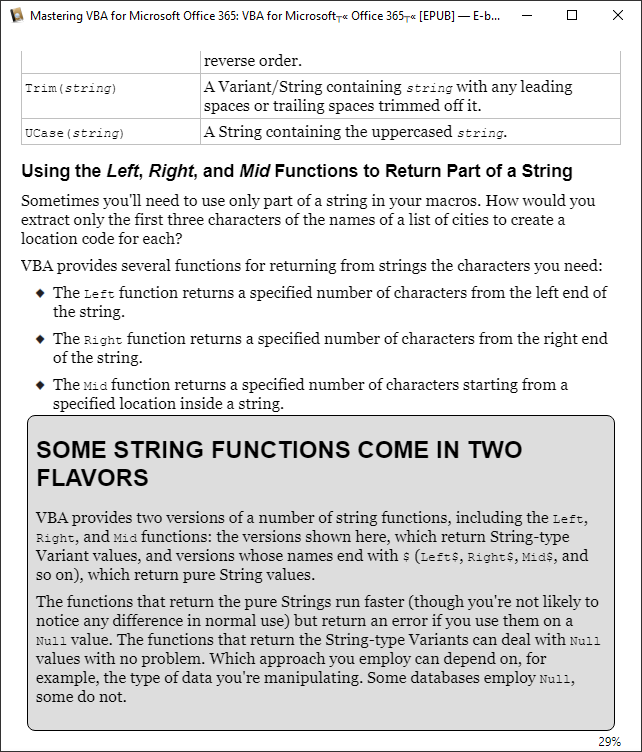
?"alfa" & vbTab & "beta" & vbTab & "gama" & vbCr & "End"

=> alfa beta gama

End

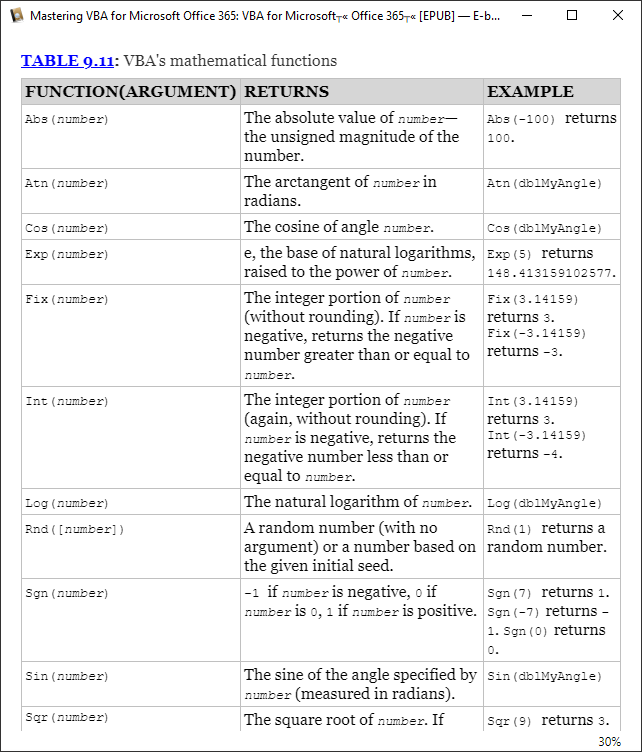
#### Utilizarea funcţiilor pentru a manipula şiruri de caractere:

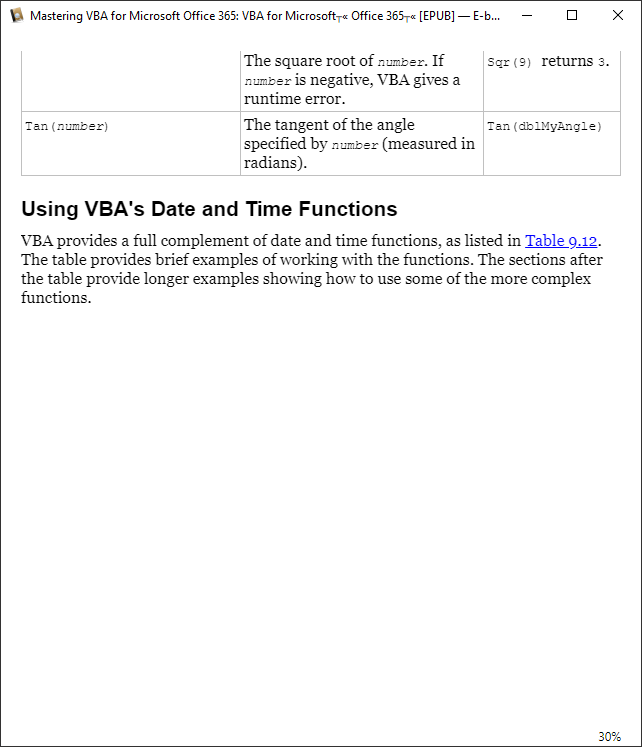




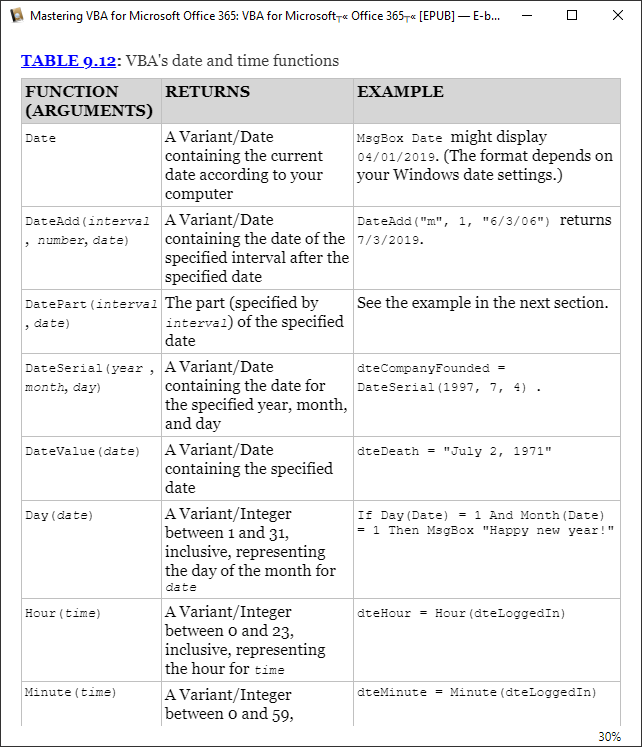
Unele funcţii au şi varianta care se termină cu un $ - returnează numai valori pure String, dar returnează eroare când se utilizează cu valori Null.

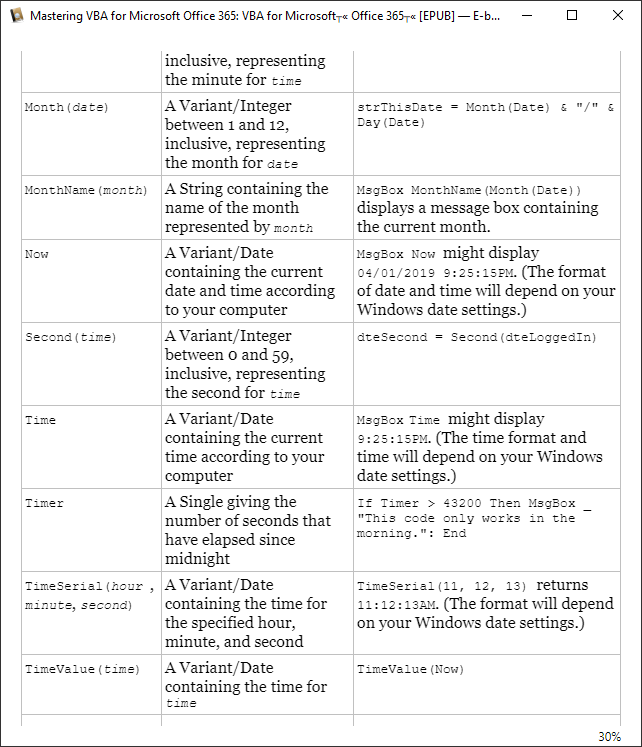
#### Utilizarea funcţiilor matematice VBA:

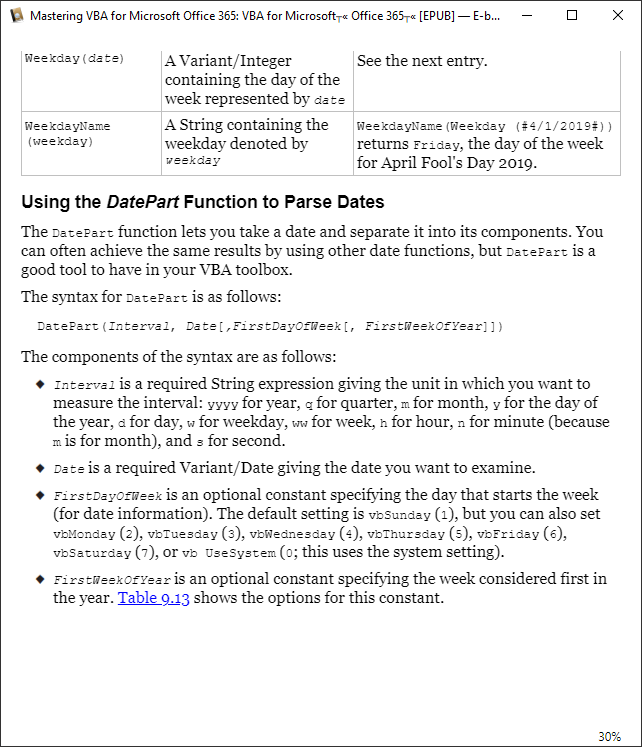




#### Utilizarea funcţiilor de dată calendaristică şi de timp VBA:







Ex:

?Date => 10/24/2022

?DateAdd("m", 3, Date) => 1/24/2023

?DateSerial(1960, 12, 20) => 12/20/1960

?DateValue("April 1, 1964") => 4/1/1964

?WeekdayName(Weekday(DateValue("April 1, 1964"))) => Wednesday

?WeekdayName(Weekday(DateSerial(1960, 12, 20))) => Tuesday

?DatePart("yyyy",Date) => 2022

?DateDiff("ww", "12/20/1960", "4/1/1964") => 171

?DateAdd("ww", 10, #5/27/2019#) => 8/5/2019

#### Utilizarea funcţiilor de management de fişiere:

Funcţia Dir verifică existenţa unui fişier.

Ex

?Dir("C:\Users\marius\Documents\work\vba\papa.txt") => papa.txt

Funcţia CurDir returnează calea curentă

Ex:

?CurDir => C:\Users\marius\AppData\Roaming\Microsoft\Templates

?CurDir("C") => C:\Users\marius\AppData\Roaming\Microsoft\Templates

### Cap. 10. Crearea funcţiilor

Funcţiile se scriu în fereastra Code, nu se pot înregistra cu Macro Recorder – acesta poate crea numai subproceduri.

Funcţiile acceptă argumente (date de intrare) de la procedura care le apelează şi returnează o valoare înapoi la procedura care le apelează.

Subprocedurile nu necesită argumente şi nu returnează niciodată nici o dată.

Beneficiul principal al funcțiilor este că facilitează comunicarea – trimiţând date înainte şi înapoi.

#### Componentele unei funcţii

Sintaxa:

[Public | Private] [Static] Function *function\_name* [(*argument\_list*)] [As *type*]

[*statements*]

[*function\_name* = *expression*]

[Exit Function]

[*statements*]

[*function\_name* = *expression*]

End Function

unde argument\_list este de forma

[Optional] [ByRef | ByVal] [ParamArray] *variable\_name*[( )] [As *type*][= *default\_value*]

Implicit: ByRef.

Marcarea unei etichete în cod: nume\_eticheta:

Salt la etichetă: GoTo nume\_eticheta.

#### Crearea unei funcţii pentru Word

Option Explicit

Function Strip\_Hyperlinks\_Bookmarks\_Fields()

Dim myLink As Hyperlink

Dim myBookmark As Bookmark

Dim myField As Field

With ActiveDocument

For Each myLink In .Hyperlinks

myLink.Delete

Next myLink

For Each myBookmark In .Bookmarks

myBookmark.Delete

Next myBookmark

For Each myField In .Fields

myField.Delete

Next myField

End With

End Function

Sub Clean\_Up\_Document\_for\_Conversion()

Call Strip\_Hyperlinks\_Bookmarks\_Fields

End Sub

#### Crearea unei funcţii pentru Excel

Option Explicit

Function Blank\_Sheets\_in\_Workbook(ByRef WorkbookToTest As Workbook) As Boolean

Dim objWorksheet As Worksheet

Blank\_Sheets\_in\_Workbook = False

For Each objWorksheet In WorkbookToTest.Worksheets

If Application.WorksheetFunction.CountBlank \_

(objWorksheet.Range("A1:IV65536")) = 16777216 Then

Blank\_Sheets\_in\_Workbook = True

Exit Function

End If

Next objWorksheet

End Function

Sub Check\_Workbook\_for\_Blank\_Worksheets()

If Blank\_Sheets\_in\_Workbook(ActiveWorkbook) = True Then

MsgBox "This workbook contains one or more blank worksheets."

End If

End Sub

#### Crearea unei funcţii pentru PowerPoint

Option Explicit

Function CheckMinFontSize(objPresentation As Presentation) As Boolean

Dim objSlide As Slide

Dim objShape As Shape

CheckMinFontSize = True

For Each objSlide In objPresentation.Slides

objSlide.Select

objSlide.Shapes.SelectAll

For Each objShape In Windows(1).Selection.ShapeRange

If objShape.Type = msoPlaceholder Then

If objShape.TextFrame.TextRange.Font.Size < 14 Then

CheckMinFontSize = False

Exit Function

End If

End If

Next objShape

Next objSlide

End Function

Sub Font\_Check()

If CheckMinFontSize(ActivePresentation) = False Then

MsgBox "Some of the fonts in the presentation are too small."

End If

End Sub

#### Crearea unei funcţii pentru Access

Access nu are acelaşi istoric precum celelalte programe Office, de aceea are căi speciale de programare.

Primul lucru de notat este obiectul de uz general „DiuCmd”, obiect care nui are proprietăţi, dar are o multitudine de metode pentru sarcini diferite.

Option Compare Database

Option Explicit

Function MoveToNew()

' DoCmd.OpenForm "Contact List"

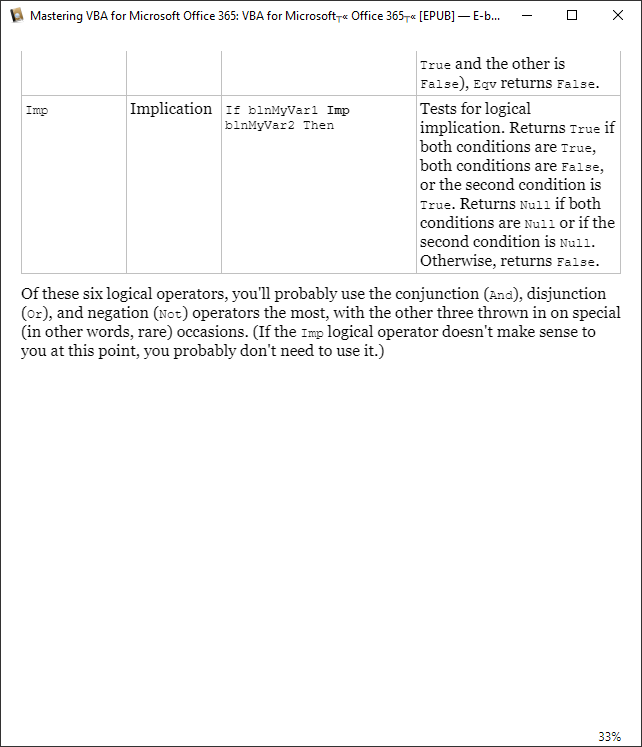
DoCmd.GoToRecord , , acNewRec

End Function

### Cap. 11. Luarea deciziilor în cod

Operatori logici:





#### Blocuri if

* If…Then
  + pe o singură linie:

If condition Then Code to be executed goes here

* + într-un bloc multilinie

If *condition* Then

Code to be executed goes here

End If

* If… Then… Else…

If *condition* Then

*statements1*

Else

*statements2*

End If

* If… Then… Elseif… Else

If *condition1* Then

*statements1*

ElseIf *condition2* Then

*statements2*

[ElseIf *condition3*

Then *statements3*]

[Else *statements4*]

End If

VBA evaluează ambele condiţii, indiferent de prima. Pentru evitarea problemelor, se recomandă imbricarea codurilor if.

Se pot include mai multe comenzi pe o linie dacă acestea sunt separate prin două puncte, :

Instrucţiunea GoTo poate produce saltul la o linie de cod marcată cu un număr sau cu o etichetă de linie, un text care începe cu o literă şi se termină cu două puncte, :

Word:

- returnarea cursorului de inserare la începutul documentului:

Selection.HomeKey Unit:=wdStory

- ştergerea formatării din comanda Fiind:

Selection.HomeKey Unit:=wdStory

- setarea căutării pentru stilul Heading 5:

Selection.Find.Style = ActiveDocument.Styles("Heading 5")

- setarea şirului de căutare ca un şir gol (“ “):

Selection.Find.Text = " "

- executarea operaţiei Find:

Selection.Find.Execute

- returnarea rezultatului căutării:

Selection.Find.Found

- setarea stilului selecţiei:

Selection.Style = "Special Note"

#### Blocuri Select Case:

Select Case *TestExpression*

Case *Expression1*

*Statements1*

[Case *Expression2*

*Statements2*]

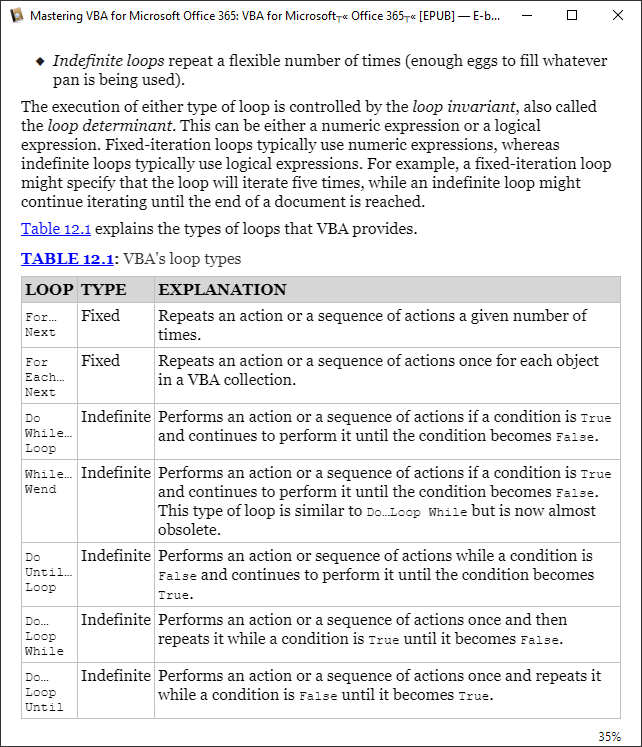
[Case Else

*StatementsElse*]

End Select

### Cap. 12. Utilizarea ciclurilor pentru acţiuni repetitive

Execuţia unui tip de ciclu este controlată de un invariant sau determinant de ciclare, o expresie numerică sau logică.



#### Cicluri For…

#### Cicluri For… Next

For *counter* = *start To end* [Step *stepsize*]

[*statements*]

[Exit For]

[*statements*]

Next [*counter*]

Ex. Word:

Sub Stepping\_Paragraphs()

Dim i As Integer

Dim total As Integer

total = ActiveDocument.Paragraphs.Count

For i = 1 To total

DoEvents

Application.StatusBar = "Paragragh number " & i & " out of " & total

Selection.MoveDown Unit:=wdParagraph, Count:=1, Extend:=wdMove

Next i

End Sub

Ex. Excel:

Sub Fill\_Cells\_Step()

Dim i As Integer

For i = 1 To 24 Step 5

ActiveCell.FormulaR1C1 = i & ":00"

ActiveCell.Offset(rowoffset:=1, columnoffset:=0).Select

Next i

End Sub

Ex. PowerPoint:

Sub Create\_Presentations()

Dim intPresentations As Integer

Dim i As Integer

intPresentations = InputBox \_

("Enter the number of presentations to create:", \_

"Create Presentations")

For i = 1 To intPresentations

Presentations.Add

Next i

End Sub

Funcţia MkDir creează un folder.

Funcţia ChDir schimbă directorul curent.

Căsuţe de dialog Word încorporate:

<https://msdn.microsoft.com/en-us/library/bb208857.aspx>

Ex. Word:

Sub Folders\_Working()

Dim strFolder As String

MsgBox "Current Folder: " & CurDir

strFolder = InputBox("Folder Name: ", \_

"Change Folder")

ChDir (strFolder)

MsgBox "Current Folder: " & CurDir

strFolder = InputBox("Folder Name: ", \_

"Create Folder")

MkDir strFolder

End Sub

#### Cicluri For Each… Next

Iteraţiile se bazează pe numărul de obiecte dintr-o colecţie.

Sintaxa:

For Each *object* In *collection*

[*statements*]

[Exit For]

[*statements*]

Next [*object*]

Ex. Word:

Sub Close\_All\_Documents()

Dim Doc As Document

For Each Doc In Documents

Doc.Close SaveChanges:=wdSaveChanges

Next Doc

End Sub

#### Cicluri Do

 Do While…Loop

 Do…Loop While

 Do Until…Loop

 Do…Loop Until

Sintaxa:

Do While *condition*

[*statements*]

[Exit Do]

[*statements*]

Loop

Do

[*statements*]

[Exit Do]

[*statements*]

Loop While *condition*

Do Until *condition*

*statements*

[Exit Do]

[*statements*]

Loop

Do

[*statements*]

[Exit Do]

[*statements*]

Loop Until *condition*

#### Cicluri While.. Wend

Sintaxa:

While *condition*

[*statements*]

Wend

VBA permite 16 nivele de imbricare.

Se recomandă scrierea contorului după Next.

## Partea 4. UTILIZAREA MESSAGE BOXES, INPUT BOXES ŞI DIALOG BOXES

### Cap. 13. Preluarea datelor de la utilizator cu Message Boxes şi Input Boxes

Afişarea mesajelor barei de stare în Word şi Excel:

Application.CommandBars("Status Bar").Visible = False | True

Application.StatusBar = "Excel is busy formatting. Please wait…"

Mesajele aplicaţiilor înlocuiesc mesajele create de utilizator în bara de stare.

#### Message Boxes

Sintaxa:

MsgBox(*prompt*[, *buttons*] [, *title*][, *helpfile*, *context*])

Prompt: până la 1024 caractere.

Mesaj multilinie:

- Chr(13) sau vbCr => carriage return

- Chr(10) sau vbLf => line feed

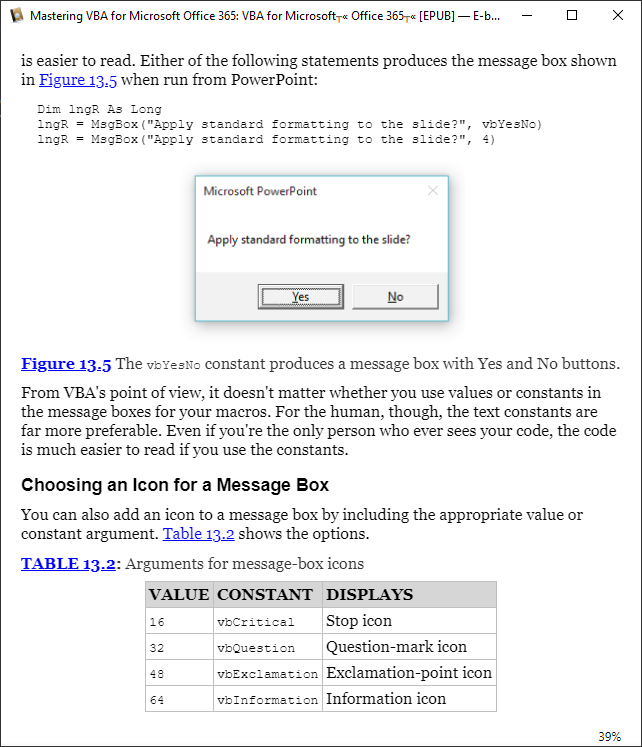
- Chr(10) + Chr(13) sau vbCrLf => line-feed/carriage-return

- Chr(9) sau vbTab => tab

- Chr(149) => bullet

Alegerea butoanelor pentru un Message Box (buttons):





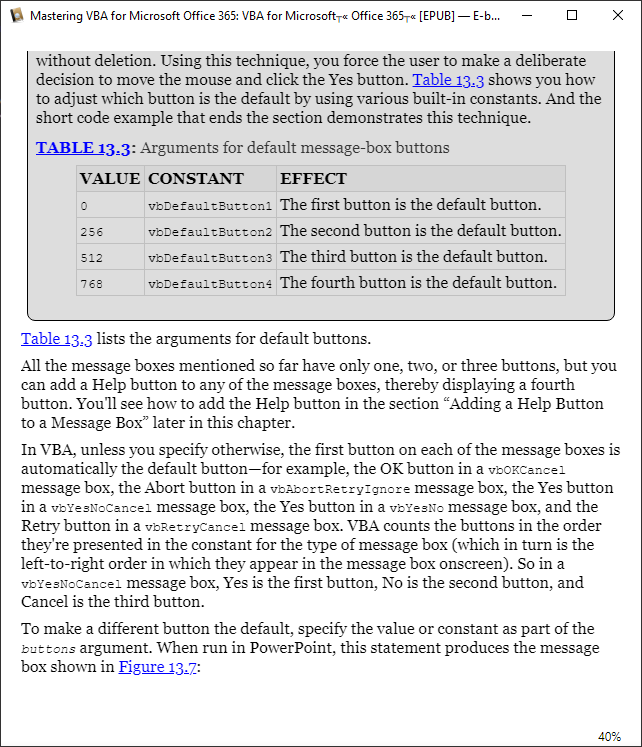
Ex:

MsgBox (“Continue?”, vbYesNo + vbQuestion)

MsgBox (“Continue?”, 4 + vbQuestion)

MsgBox (“Continue?”, 4 + 32)

VBA setează automat primul buton ca buton implicit.



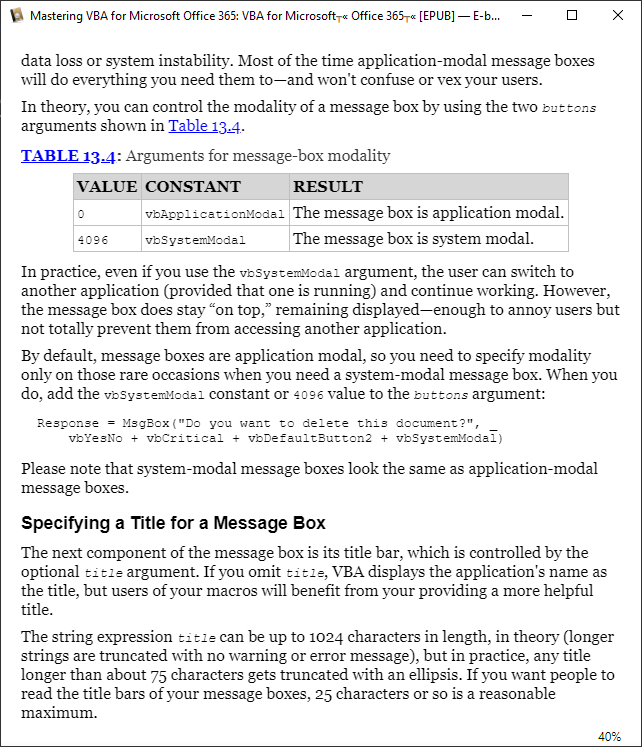
Ex:

MsgBox (“Continue?”, vbYesNo + vbQuestion + vbDefaultButton2) => No implicit

Tipuri de Message Boxes:

- modal-aplicaţie

- modal-sistem



Ex:

MsgBox (“Continue?”, vbYesNo + vbQuestion + vbDefaultButton2 + vbSystemModal)

Dacă se omite argumentul title, VBA afişează numele aplicaţiei ca titlu. Poate avea până la 1024 caractere în teorie, dar, în practică, este trunchiat după 75 caractere şi continuat cu trei puncte. Recomandat – 25 caractere.

Adăugarea unui buton Help la un Message Box: vbMsgBoxHelpButton.

Argumentul helpfile: numele şi locaţia unui fişier de Help.

Argumentul context: numărul de context Help în interiorul fişierului de Help – controlează care topic al fişierului de Help este afişat.

Ex:

MsgBox("Do you want to delete this workbook?", vbYesNo \_ + vbCritical + vbDefaultButton2 + vbMsgBoxHelpButton, \_

"Delete Workbook", "c:\Windows\Help\My\_Help.chm", 1012)

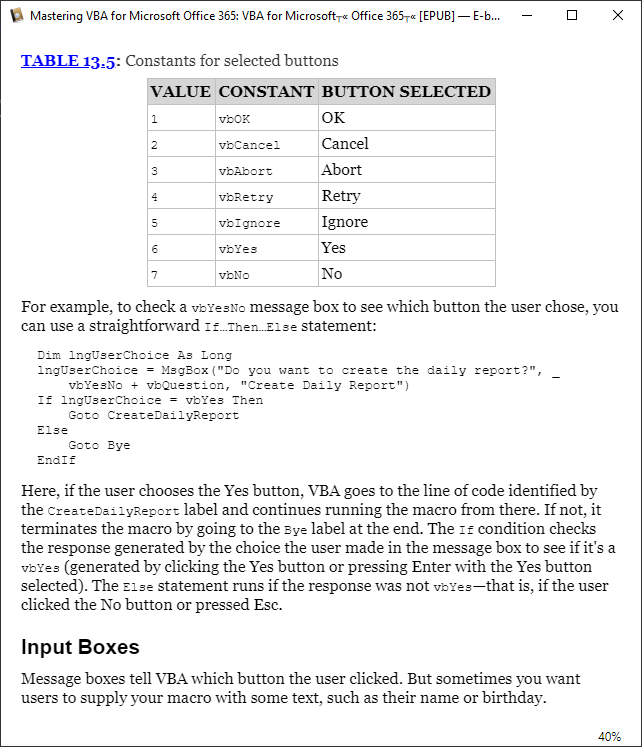
Numărul de context Help pentru deschiderea ecranului unui fişier de Help este 0 – se utilizează când este necesară afişarea unui fişier de Help, dar nu se ştie numărul de context şi informaţia se localizează manual.

Omitere argumente opţionale: prin virgule sau prin specificarea argumentelor cu nume, ceea ce le face mai uşor de citit.

Ex:

MsgBox("Do you want to format the report?", \_

HelpFile:="c:\Windows\Help\Macro Help.chm", Context:=1012)



#### Input Boxes

Sintaxa:

InputBox(*prompt*[, *title*] [, *default*] [, *xpos*] [, *ypos*] [, *helpfile*, *context*])

Valorile xpos şi ypos determină poziţia faţă de marginile ecranului, nu ale ferestrei aplicaţiei, în twips.

Implicit: la jumătate pe orizontală şi la o treime pe verticală.

1 twip = 1/1440 inch

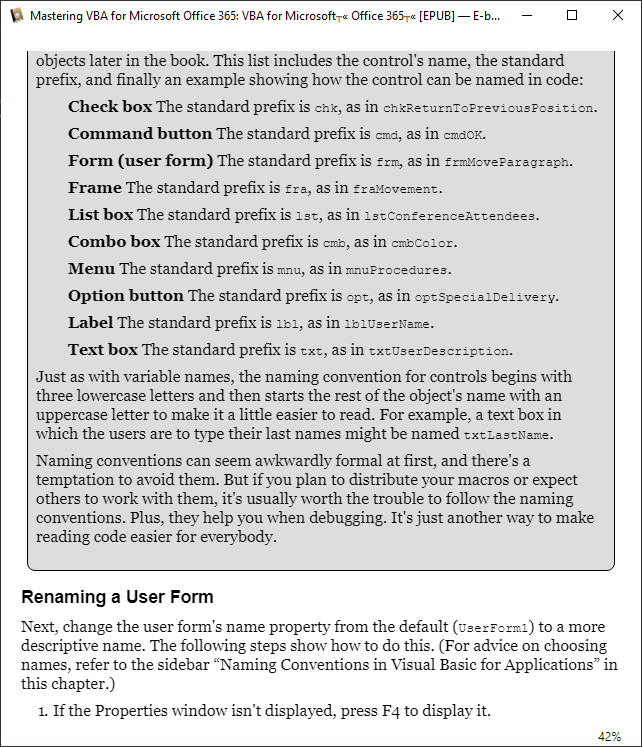
### Cap. 14. Crearea de căsuţe de dialog (Dialog Boxes) personalizate simple

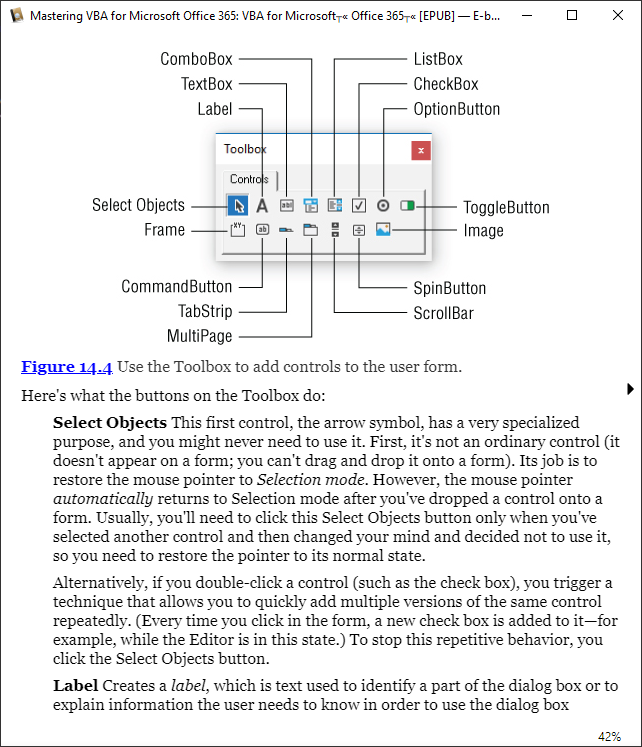
#### Crearea unei căsuţe de dialog

Orice formă utilizator este un obiect care conţine un număr de alte obiecte care pot fi manipulate individual.

Componentele unei forme se numesc controale, sunt obiecte vizuale.

Prefixuri standard:





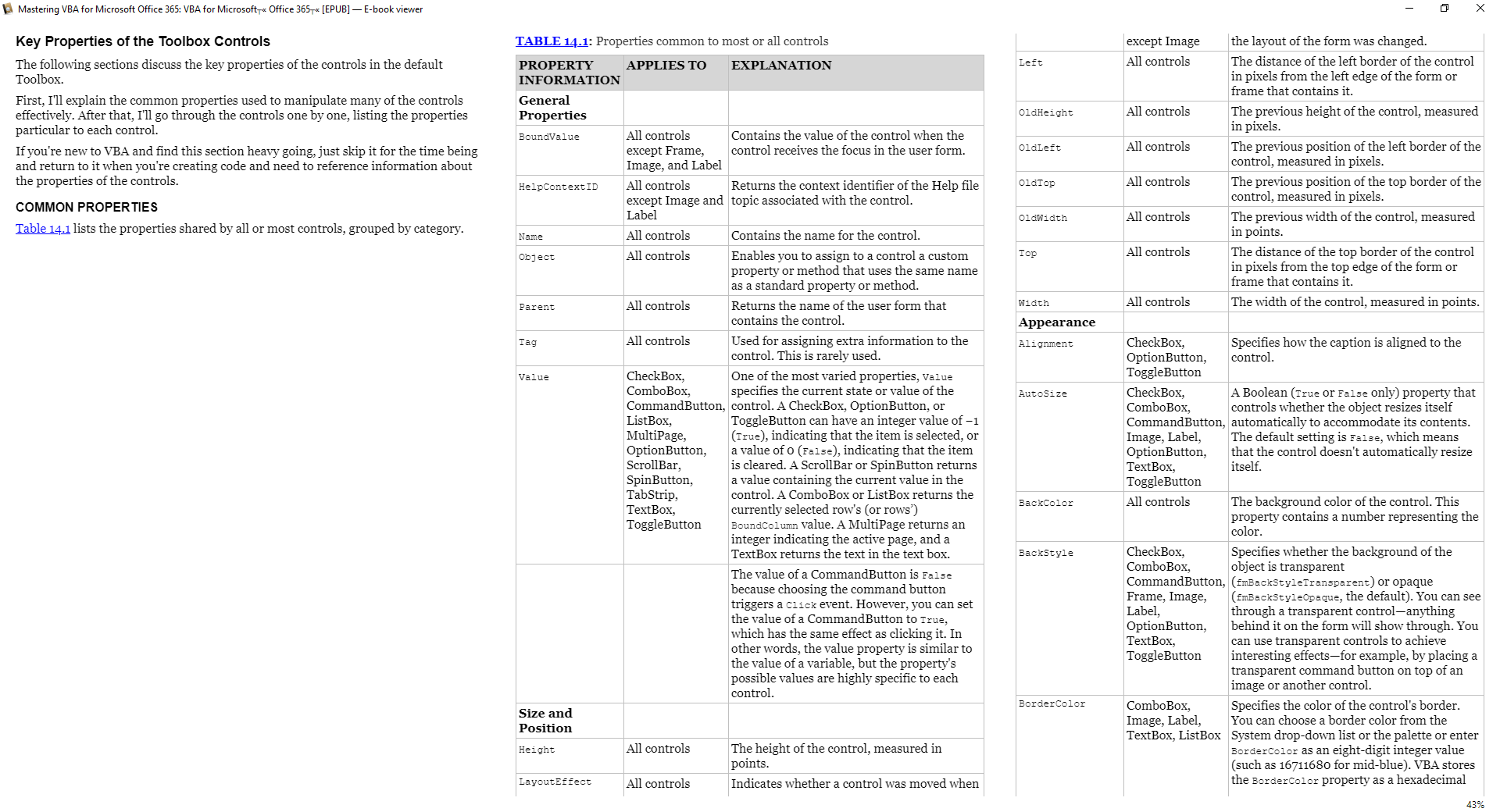
Printr-un dublu-click pe un control, se pot adăuga mai multe versiuni ale aceluiaşi control printr-un click simplu în formă. Deselectare – săgeata Select Objects.

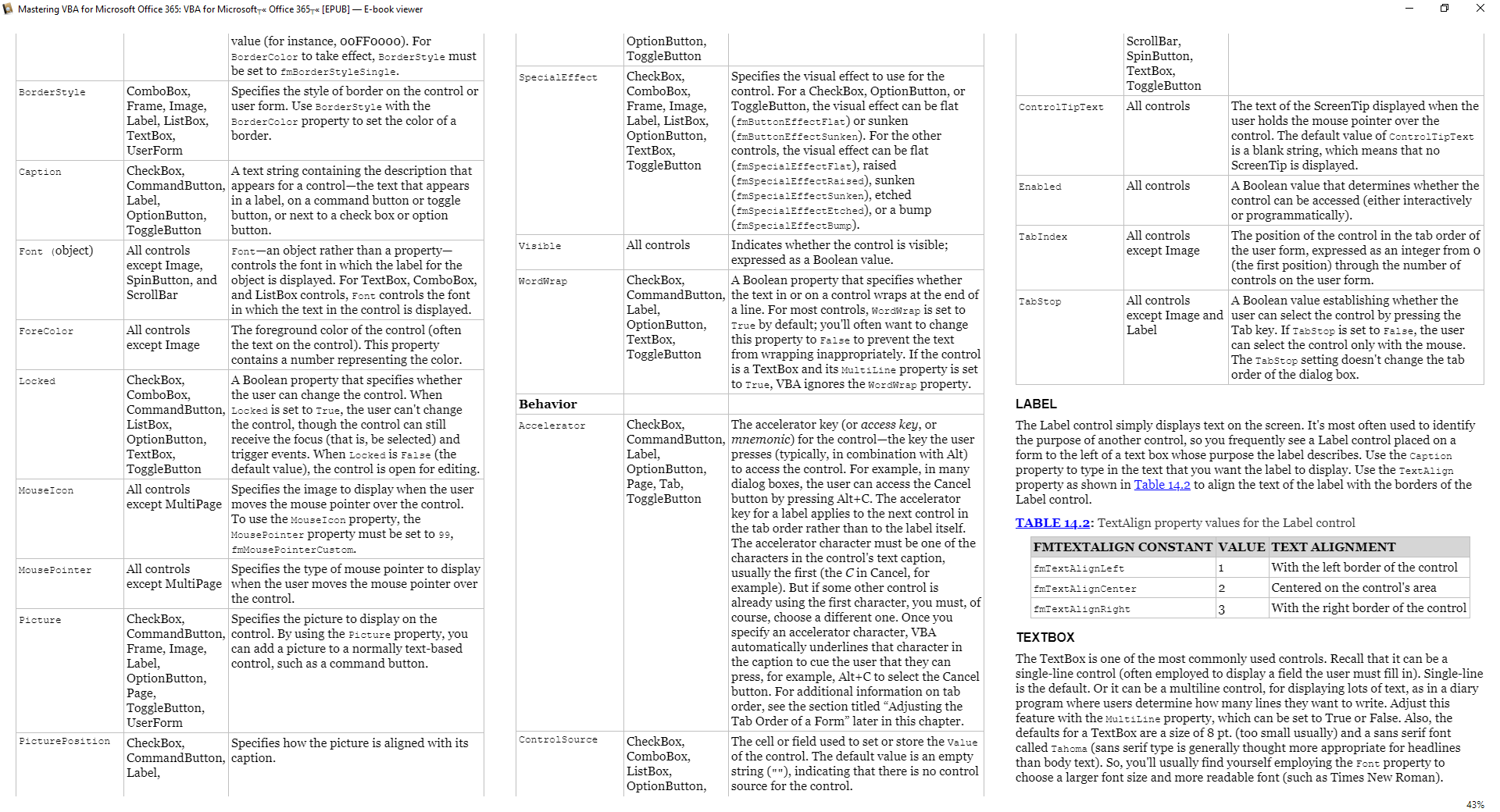
Dublu-click pe un control afişează pagina de cod pentru forma utilizator şi adaugă automat o procedură pentru evenimentul Click al controlului – vederea Code.

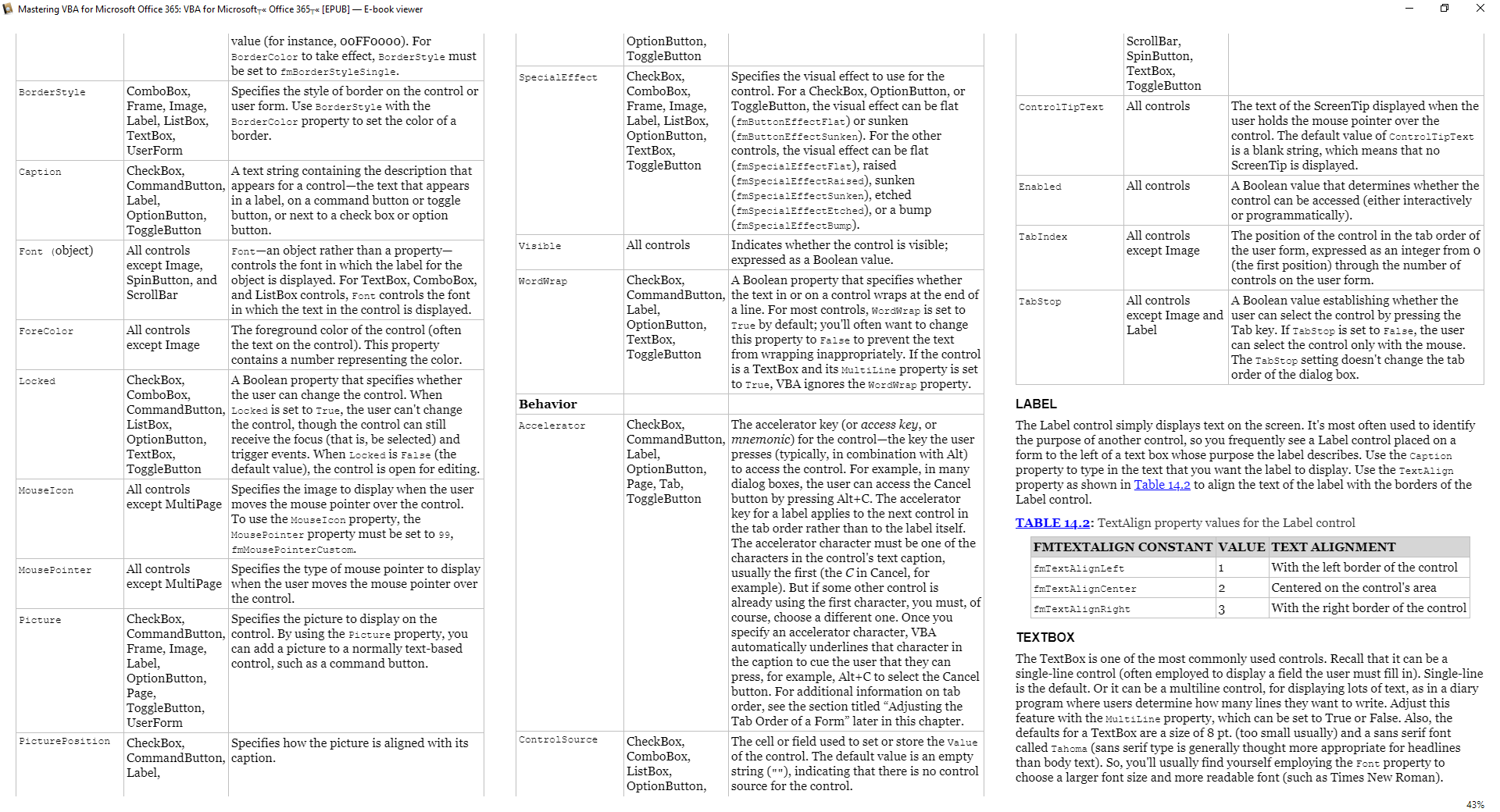
Revenirea la vederea Design, pentru a vedea forma – Shift + F7.

Procedura UserForm\_Initialize rulează atunci când forma este iniţializată pentru afişare.

#### Proprietăţi comune ale controalelor din Toolbox:



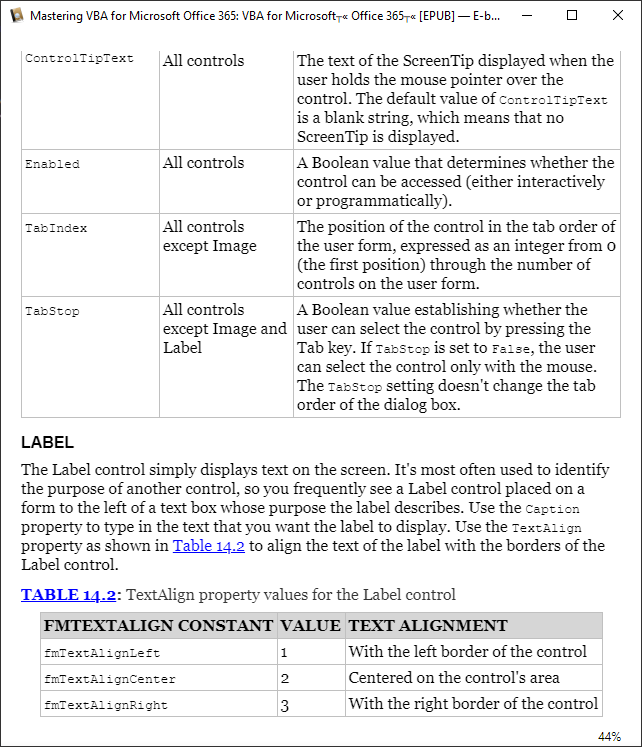




#### Label

Controlul Label afişează text pe ecran pur şi simplu.

Proprietatea Caption: textul afişat.



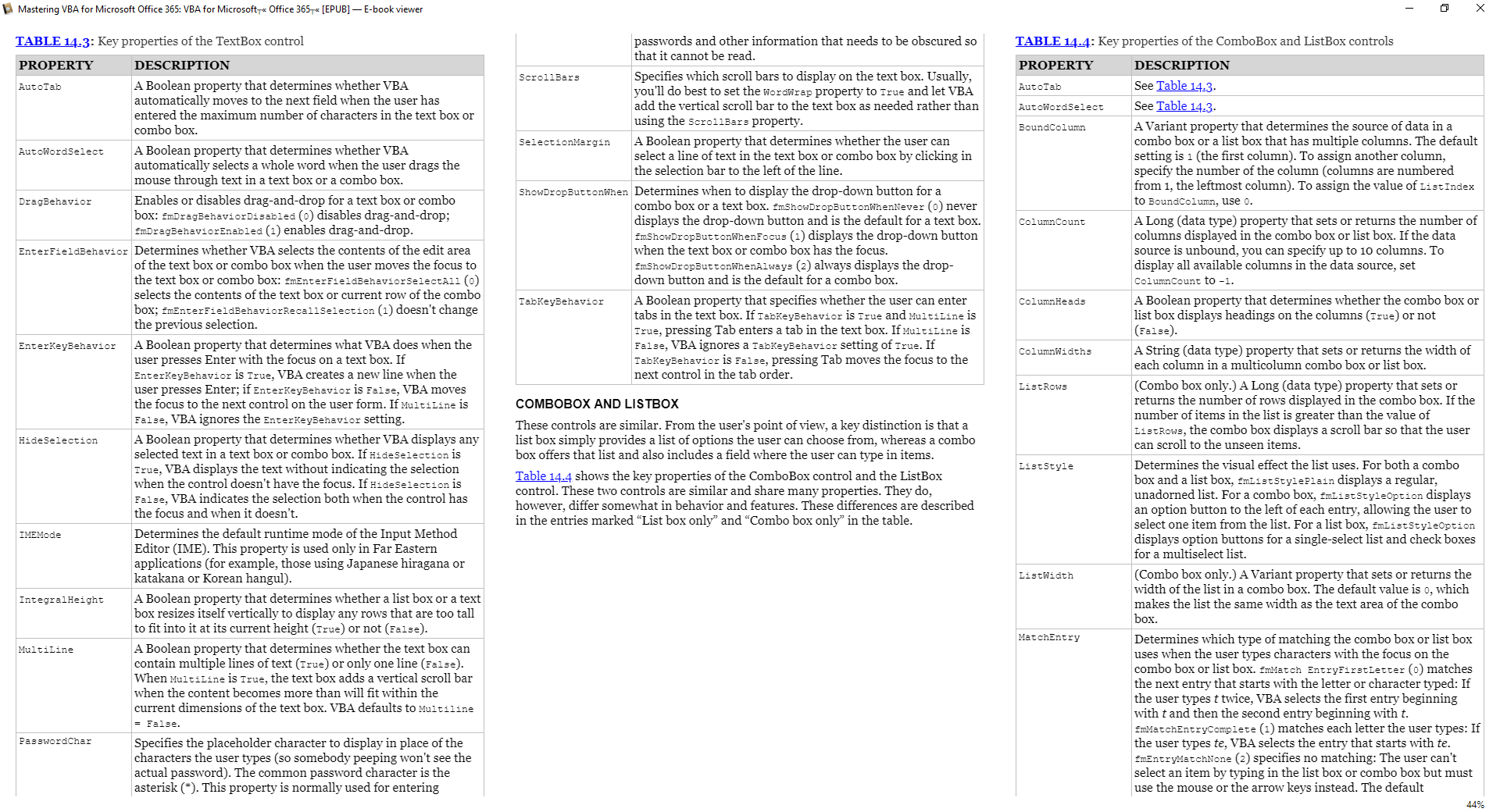
#### Textbox

Preia date introduse de utilizator.

Poate fi un control pe o singură linie (implicit) sau multilinie (proprietatea MultiLine setată la True).

Introducerea mai multor linii: Shift + Enter.

Adăugarea de bare de defilare: proprietatea ScrollBars = 1 (fmScrollBarsHorizontal), 2 (fmScrollBarsVertical) sau 3 (fmScrollBarsBoth).



#### ComboBox şi ListBox

Sunt controale similare:

- o căsuţă listă furnizează o listă din care utilizatorul poate alege;

- o căsuţă combo oferă acea listă şi, de asemenea, include un câmp unde utilizatorul poate tipări itemi;

- o listă porneşte goală şi se creează o procedură pentru a iniţializa itemii care se vor afişa în listă: Code > UserForm > Initialize => lstNumLista.AddItem “Text item”.

Ex:

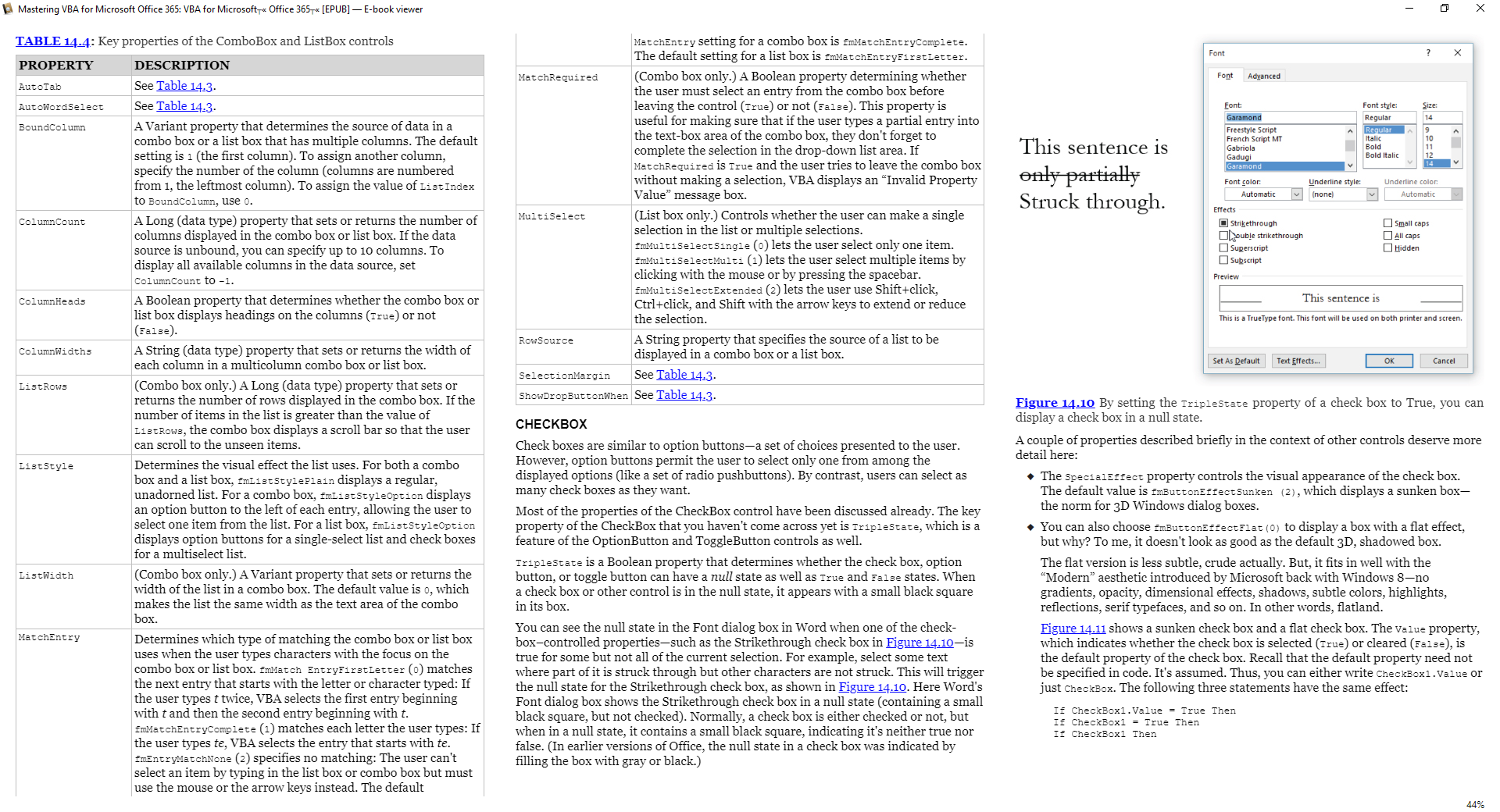
Private Sub UserForm\_Initialize()

lstBat.AddItem "Bat 1"

lstBat.AddItem "Bat 2"

lstBat.AddItem "Bat 3"

End Sub



#### CheckBox

Are în plus proprietatea TripleState, o proprietate booleană care determină dacă acel control poate să aibă o stare null, pe lângă stările True şi False. Caracteristica este comună şi controalelor OptionButton şi ToggleButton.

Când controlul este în starea null, el apare cu un pătrat mic negru în căsuţa lui.

Prin setarea proprietăţii TripleState la True, controlul este afişat în starea null.

Starea null înseamnă nici complet adevărat, nici complet fals – ca într-un paragraf în care numai câteva cuvinte sunt subliniate, paragraful este parţial subliniat.

Proprietatea SpecialEffect controlează aparenţa vizuală a obiectului, fmButtonEffectSunken (2) sau fmButtonEffectFlat (0).

#### OptionButton

Este util pentru alegerea unui item dintr-un grup de butoane de opţiuni reciproc exclusive.

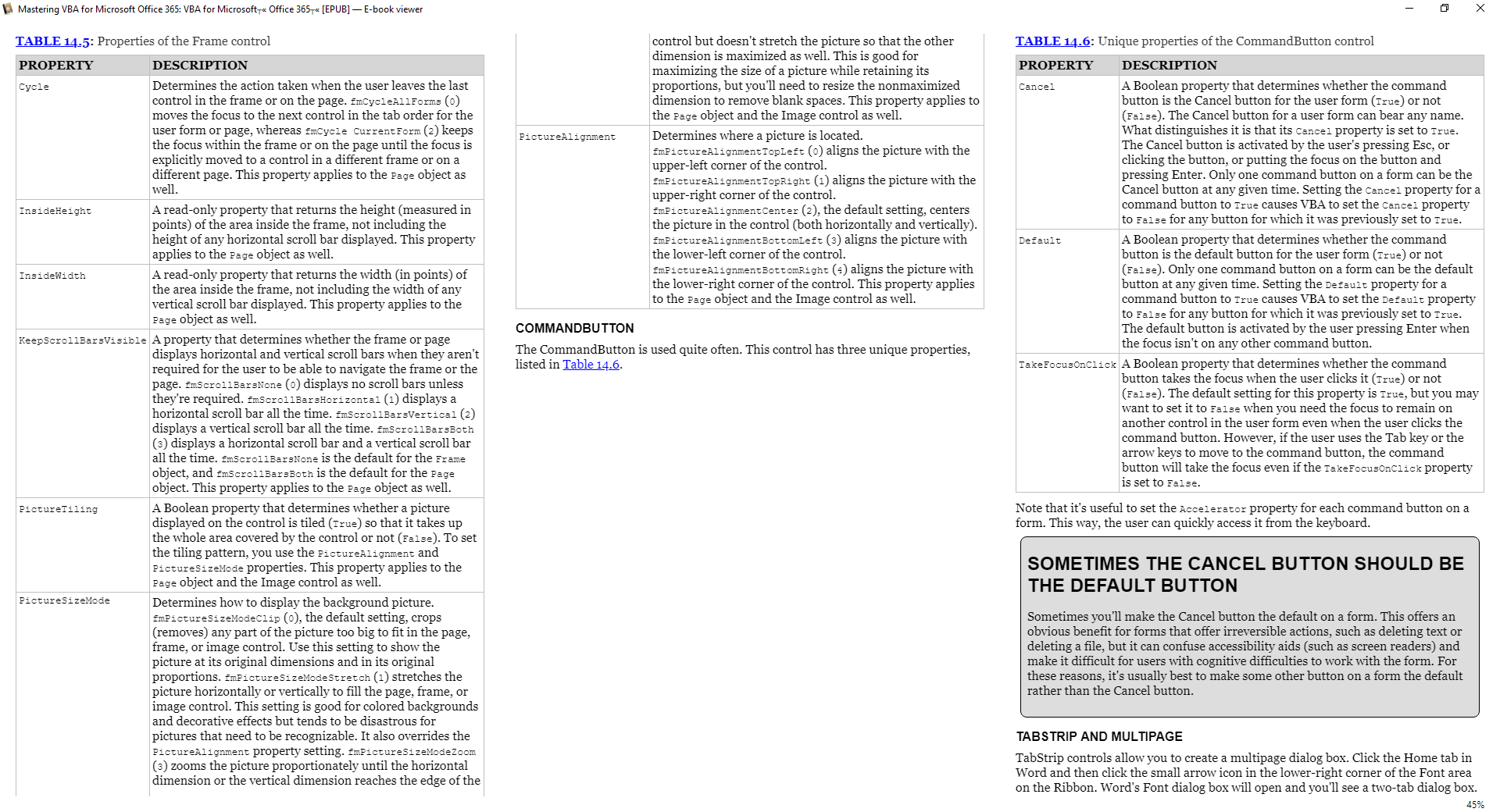
Proprietatea GroupName este unică pentru OptionButton şi ataşează butonul de opţiuni unui grup de opţiuni.

Alternativ, se poate crea un grup prin plasarea unui set de butoane de opţiuni într-un control Frame.

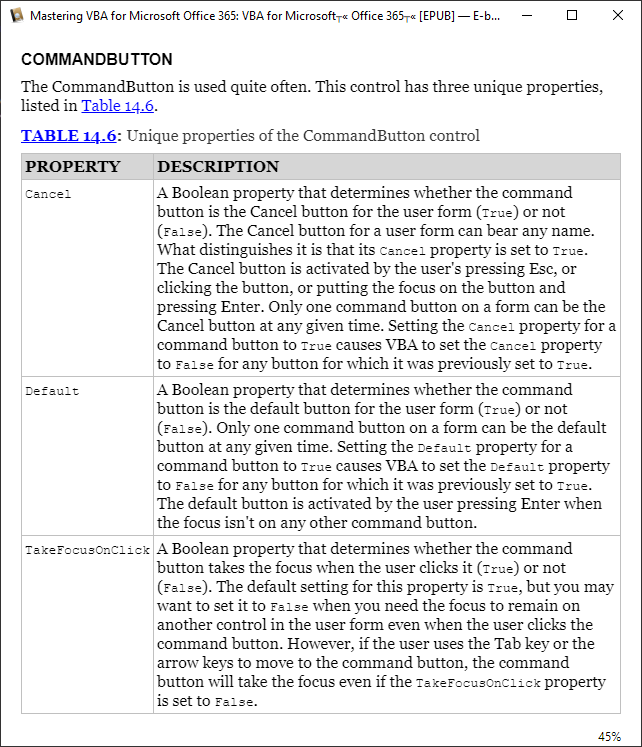
Odată grupate, butoanele devin reciproc exclusive.

#### ToggleButton

#### Frame

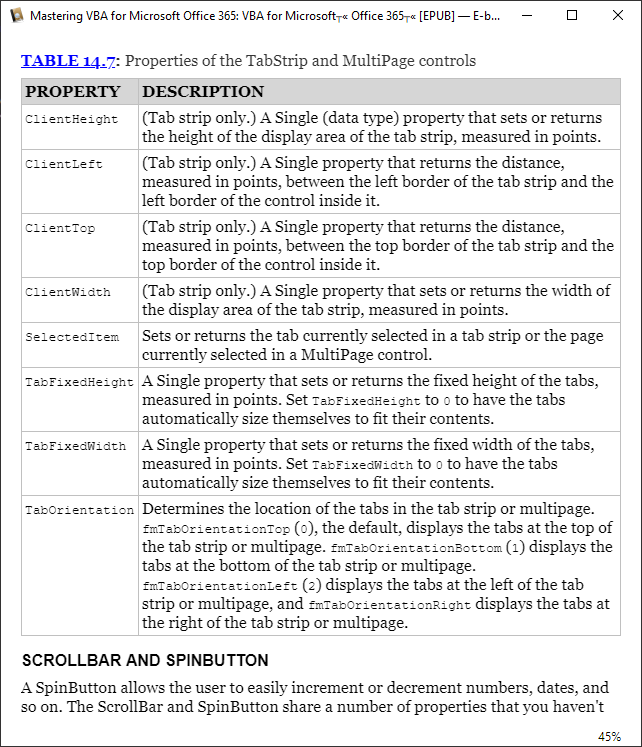


#### CommandButton

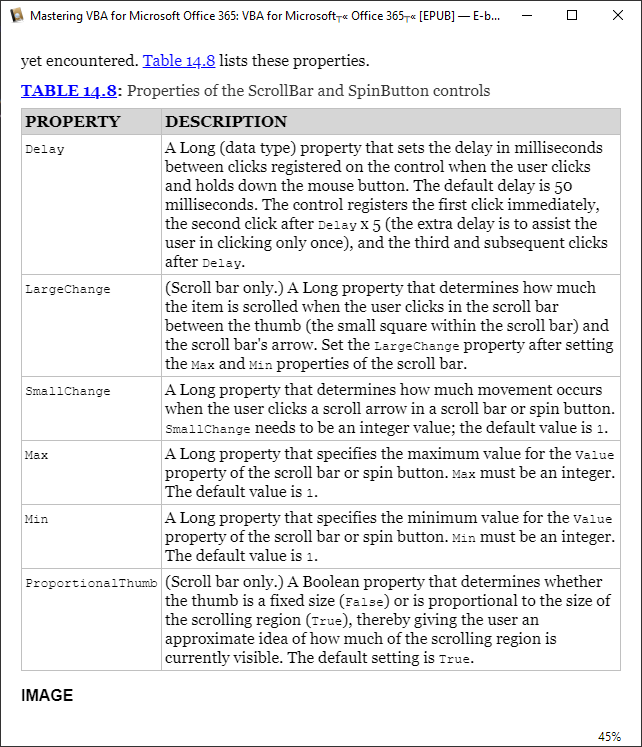


#### TabStrip şi MultiPage

Controalele TabStrip permit crearea unei căsuţe de dialog multipagină.



#### ScrollBar şi SpinButton



#### Image

Proprietăţi:

- Picture: selectarea fişierului imaginii - .bmp, .gif, .ico, .jpg, .wmf

- PictureAlignment

- PictureSizeMode

- PictureTiling

#### Page

Obiectul Page este una din paginile conţinute în obiectul MultiPage.

Proprietatea Index porneşte de la 0.

#### Tab

Obiectul Tab este unul din tab-utile conţinute în obiectul TabStrip.

Proprietatea Index porneşte de la 0.

Gruparea controalelor selectate: meniul Format > Group sau click-dreapta pe selecţie şi Group.

Alinierea controalelor: Lefts, Centers, Rights, Tops, Middles, Bottoms, To Grid.

Alinierea se face referitor la controlul dominant – cel cu pătrăţele albe de redimensionare.

Plasarea controalelor:

- meniul Format > Make Same Size > Width, Height sau Both.

- meniul Format > Size to Fit

- meniul Format > Size to Grid

- meniul Format >Horizontal Spacing

- meniul Format > Vertical Spacing

- meniul Format > Center in Form

- meniul Format > Arrange Buttons

Ajustarea ordinii tab-urilor într-o formă: click-dreapta într-un spaţiu liber al formei sau al cadrului şi Tab Order.

Legarea unei forme la o procedură:

- încărcarea/descărcarea unei forme: Load/Unload UserForm1;

- afişarea/ascunderea unei forme: UserForm1.Show/Hide;

- se poate afişa o formă direct cu Show, fără Load, care se presupune;

- setarea unui buton de comandă implicit: proprietatea Default setată la True.

Returnarea alegerilor utiliuzatorului dintr-o căsuţă de dialog:

- TextBox: proprietatea Value şi Text returnează aceeaşi valoare, la alte obiecte sunt diferite;

- OptionButton: proprietatea implicită este Value, poate fi omisă specificarea ei; poate fi verificat direct numai True sau False, Null nu poate fi verificat direct, se obţine eroare – se verifică dacă este True şi apoi dacă este False, dacă nici una, este Null;

- CheckBox: proprietatea Value poate fi True sau False;

- ListBox: proprietatea Value este un text, pentru o singură selecţie, şi Selected este un tablou cu intrările selectate dintr-o multilinie, cu proprietatea List(index).

Ex:

MsgBox lstBat.Value

MsgBox lstBat.List(i) => itemul i+1 din rândurile selectate

- ComboBox: proprietatea Value