

Swoje makra możesz tworzyć pisząc bezpośrednio ich kod. Dodatkowo możesz zastosować komponenty, (np. przyciski), które umożliwią uruchomienie makra w dowolnej chwili. Poniższe ćwiczenia demonstrują, jak stworzyć i używać prostych makr.

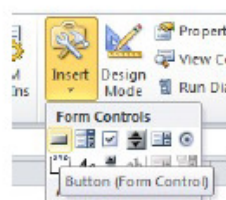
Ćwiczenie 1

W środowisku MS Excel występuje wiele komponentów służących do tworzenia interfejsu użytkownika. Jednym z najczęściej używanych jest *Przycisk*, który znajduje się w grupie *Wstaw* karty *Developer*. W celu uzyskania dostępu do tej karty konieczne jest jej uaktywnienie (o ile nie jest jeszcze wyświetlana w twojej wersji Excel-a). *Aby to zrobić:*

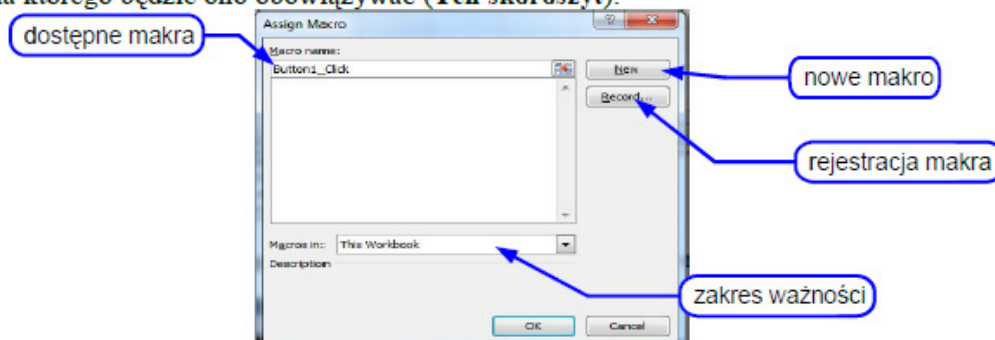
1. Na karcie *Plik* naciśnij *Opcje*
2. Naciśnij *Dostosowywanie wstążki* i aktywuj kartę *Developer* we wstążce *Karty główne*.
3. Zaakceptuj wprowadzone zmiany (OK)

Warto zauważyć, iż karta *Developer* jest dedykowana do tworzenia makr.

Umieść *Przycisk* w pustym arkuszu:

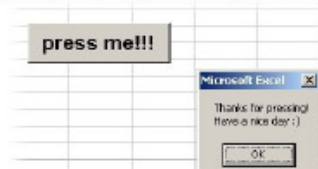


Po utworzeniu przycisku zostaje otwarte okno umożliwiające przypisanie do niego makra. Określ zakres dla którego będzie ono obowiązywać (**Ten skoroszyt**):



Zwróć uwagę, że możesz zmieniać rozmiar przycisku, jego nazwę oraz czcionkę. Następnie można przystąpić do edycji makra.

Przycisk *New* umożliwia utworzenie nowego makra. Po jego naciśnięciu otwiera się okno edytora zawierające puste makro. W celu wyświetlenia komunikatu po naciśnięciu przez użytkownika tworzonego przez nas przycisku, należy użyć funkcji *MsgBox* ("dowolny tekst").



Utwórz plik Excela (obsługujący makra) o nazwie "SomeDocument.xlsm" i wszystkie następane ćwiczenia wykonuj w tym pliku.

W ćwiczeniach 2 i 3 używana jest przedstawiona poniżej funkcja **SheetExists** służąca do sprawdzania, czy dany arkusz istnieje. Skopiuj jej kod do edytora VB:

```
Function SheetExists(BookName As String, SheetName As String) As Boolean
    Dim Book As Workbook
    Dim Sheet As Worksheet

    'returns false if a worksheet name is an empty string
    If Len(SheetName) = 0 Then
        SheetExists = False
        Exit Function
    End If

    'checking if a worksheet exists
    Set Book = Workbooks(BookName)
    For Each Sheet In Book.Sheets
        If Sheet.Name = SheetName Then
            SheetExists = True
            Exit Function
        End If
    Next Sheet

    SheetExists = False
End Function
```

Ćwiczenie 2

Wstaw do arkusza przycisk i przypisz do niego procedurę wstawiającą pusty arkusz do skoroszytu (zapisz plik). W celu przypisania użyj wyrażenia *Call* poprzedzającego nazwę procedury (np. Call mojabprocedura). Wykorzystaj przykładowy kod, podany poniżej:

```
Sub SimpleAddSheet()
    Dim SheetFound As Boolean
    Dim BookName As String
    Dim NewSheetName As String

    BookName = "SomeDocument.xlsm"
    NewSheetName = "new worksheet"

    'check if a worksheet "new worksheet" exists in "SomeDocument.xlsm" workbook
    SheetFound = SheetExists(BookName, NewSheetName)

    'if a worksheet does not exist, create it
    If Not SheetFound Then
        Workbooks(BookName).Activate
        ActiveWorkbook.Sheets.Add
        ActiveSheet.Name = NewSheetName
    End If
End Sub
```

Ćwiczenie 3

Dodaj do kodu z poprzedniego ćwiczenia funkcjonalność pytania użytkownika o nazwę nowego arkusza. Użyj:

```
NewSheetName = InputBox("Wprowadz nazwe arkusza:", "Dodaj nowy arkusz", "cokolwiek")
```

Po uruchomieniu makra powinno się pojawić okno z pytaniem o nazwę arkusza.

Ćwiczenie 4

Utwórz makro pytające o nazwę arkusza do usunięcia, a następnie usuwające go (jeżeli arkusz istnieje). Jeżeli arkusz nie istnieje, powinien zostać wyświetlony odpowiedni komunikat. Użyj kodu:

```
If SheetFound Then
    Workbooks(BookName).Sheets(SheetNameToDelete).Activate
    ActiveWindow.SelectedSheets.Delete
Else
    MsgBox "Nie znaleziono arkusza...", vbOKOnly, "Warning"
End If
```

Ćwiczenie 5

Utwórz makro przełączające wyświetlanie siatki. Kod:

```
ActiveWindow.DisplayGridlines = Not ActiveWindow.DisplayGridlines
```

Ćwiczenie 6

Utwórz makro zamieniające w wybranym zakresie wyrównywanie wartości całkowitych w komórkach, w zależności od ich wartości. Skorzystaj z kodu:

```
Dim MyRange As Object
Dim CCell As Object
Dim Alignment As Integer

Set MyRange = ActiveWindow.Selection
For Each CCell In MyRange
Alignment = (CCell.Value + 1) Mod 3
With CCell
    Select Case Alignment
        Case 0
            .HorizontalAlignment = xlLeft
        Case 1
            .HorizontalAlignment = xlCenter
        Case 2
            .HorizontalAlignment = xlRight
    End Select
End With
Next CCell
```

Ćwiczenie 7

Napisz makro zmieniające w wybranym obszarze kolor komórek zawierających dane liczbowe. Jeżeli wartość komórki jest większa od 50, to powinno być czerwone, w przeciwnym razie powinno być niebieskie. Skorzystaj z kodu:

```
Dim Row As Integer
Dim Col As Integer

'loop over all selection rows and columns
For Row = 0 To Selection.Rows.Count - 1
    For Col = 0 To Selection.Columns.Count - 1
        'highlight the cell according to its value
        With ActiveCell.Offset(Row, Col)
            If .Value > 50 Then
                .Interior.ColorIndex = 3
            ElseIf .Value <= 50 Then
                .Interior.ColorIndex = 5
            End If
        End With
    Next Col
Next Row
```