

Excel Makro (VBA)

SEMİH SUNGUR

Makrolarla Çalışmak

- **Makro:** Sık yapılan işleri otomatikleştirmek için kullanılır.

- Eğer Microsoft Excel'de belirli bir işi sıklıkla tekrar ediyorsanız makroları kullanarak kaydedip, tekrar etmek gerektiğinde çalıştırarak otomatik hale getirebilirsiniz.

- **Makro Oluşturma Türleri**
  - Record Macro ile makro oluşturmak
  - Visual Basic Editörü ile makro oluşturmak
  - Module ve Form import ederek makro oluşturmak

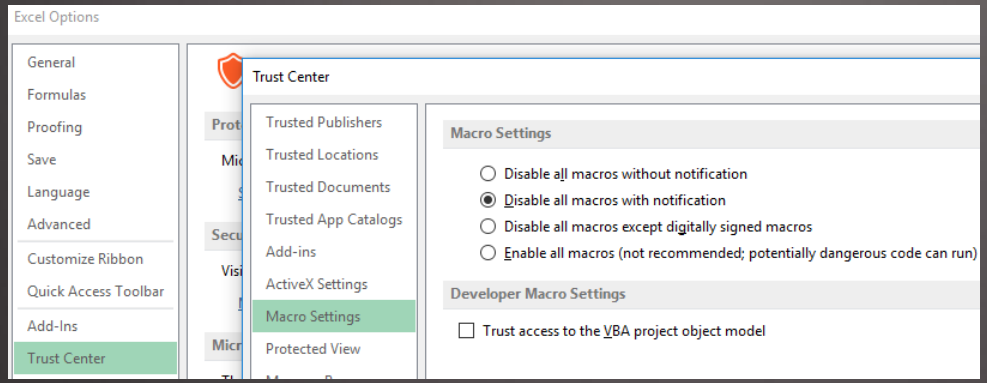
Makro Kaydetmek

1. **View sekmesindeki** Record Macro komutunu seçin.
2. **Macro Name** kutusuna makro için bir ad yazın.
3. **Store macro in** kutusundan makroyu kaydetmek istediğiniz yeri seçin.
4. **OK** düğmesine tıklayarak makro kayıt işlemine başlanır.
5. Makroya sıkça tekrar etmek durumunda olduğunuz işlemleri yapın.
6. Adımları tamamladığınızda Kaydı bitirmek için **Stop Recording** düğmesine tıklayın.

Makroları Çalıştırmak

1. View sekmesinden Macros ile
2. ALT+F8 tuş bileşimi
3. Sayfaya yerleştirilen düğme aracılığıyla
4. Bir resim veya eklenen bir nesne aracılığıyla
5. Herhangi bir sekme üzerine düğme yerleştirerek
6. VBA penceresinden run komutuna tıklayarak.
7. Özel olaylara bağlı olarak çalıştırmak

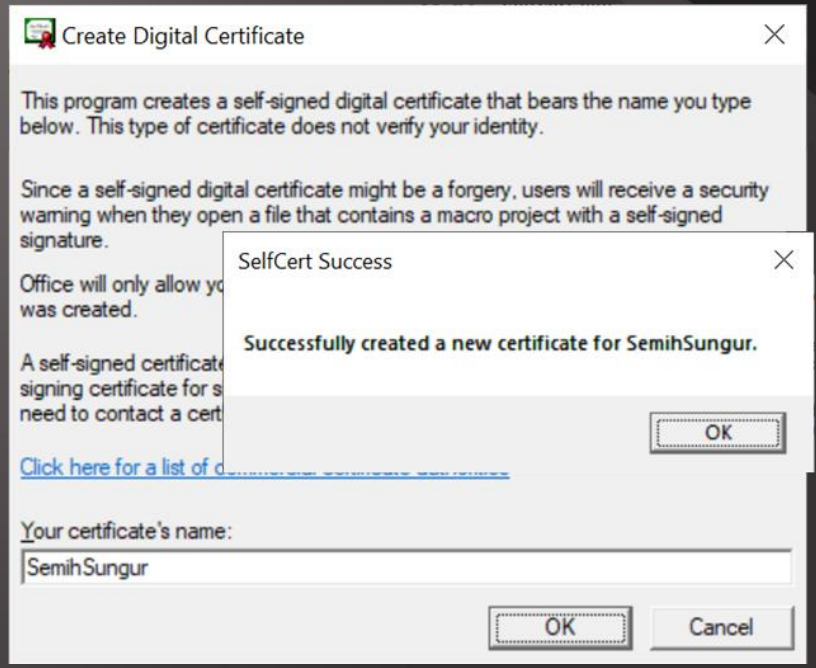
Makrolarda Güvenlik



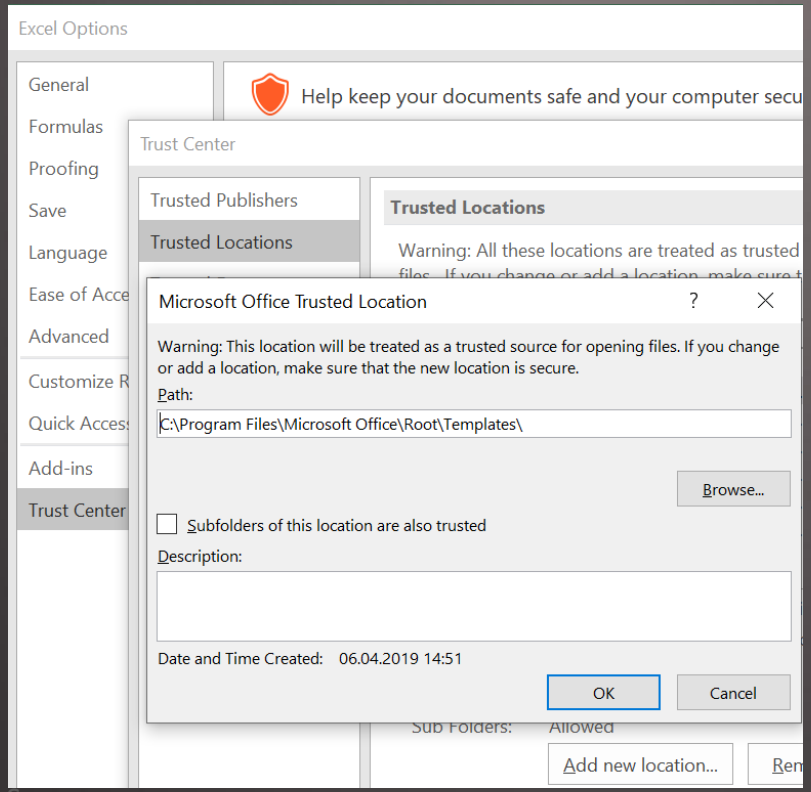
- File / Excel Options / Trust Center / Macro settings
- Developer / Macro settings
- Sayısal İmza Eklemek
- Güvenli lokasyonlar belirlemek.

Dijital Sertifika Eklemek

- Bilgisayarınızdaki SELFCERT.EXE isimli dosyayı bulup çalıştırın.
- İsim yazıp Ok seçin.
- İmza ekleyeceğiniz ofis dosyasını açın.
- Bu dosyadaki File / info / Protect workbook / Add a digital signature seçin.



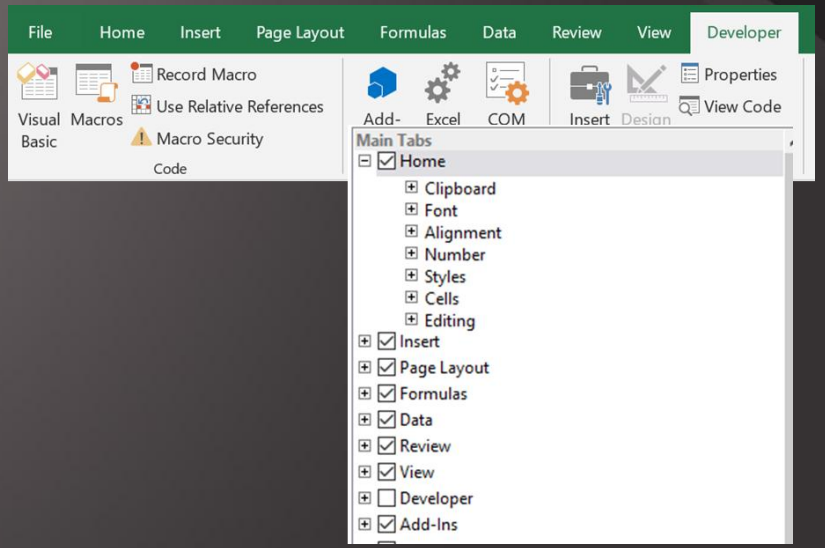
Güvenilir Lokasyonlar Belirlemek



- File / Options / Trust Center / Trust Center Settings seçin.
- Trusted Locations tıklayın.
- Add new locations seçin
- Güvenli klasörü belirleyin.

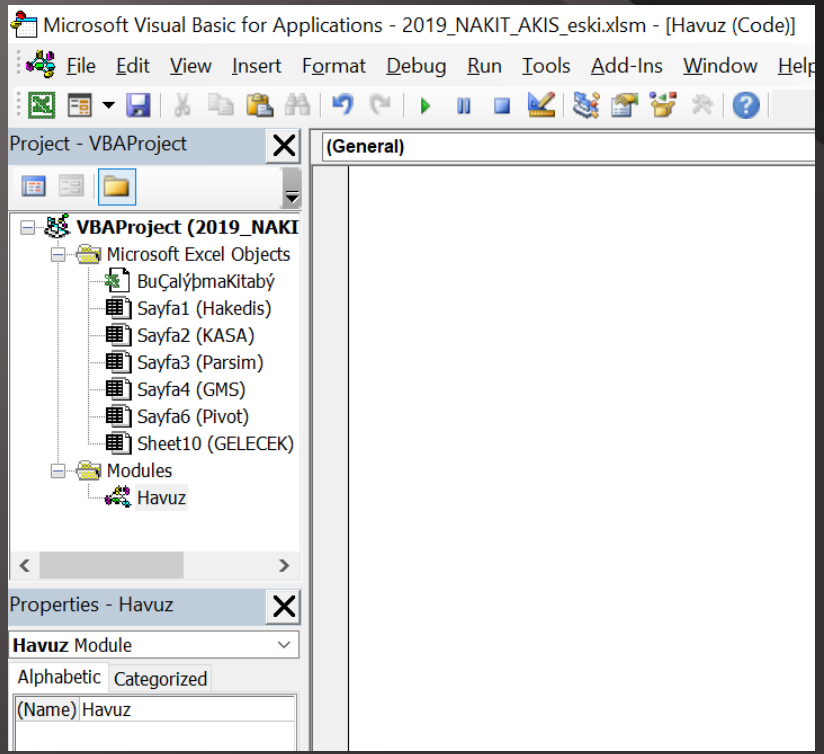
Geliştirici Sekmesi

- **File** ardından **Excel Options** seçin ve **Customize Ribbon** seçebilir veya şerit üzerinde sağ tıklayıp **Customize Ribbon** seçebilirsiniz.
- Daha sonra **Developer** seçeneği seçili hale getirin.



Visual Basic Editor

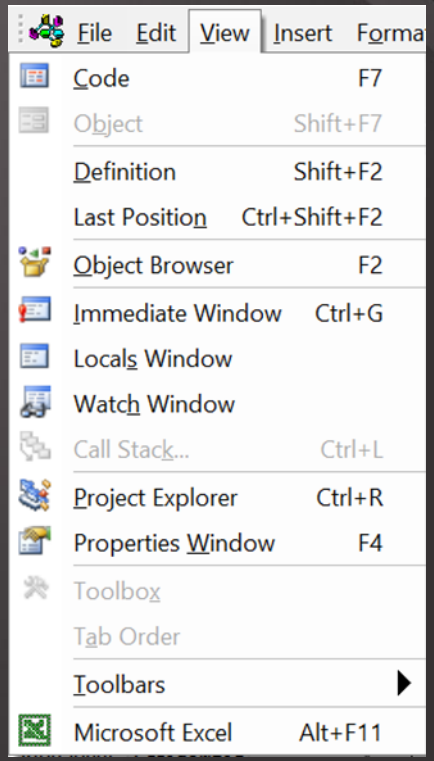
- **Project Explorer (Proje Gezgini) :** Bir çalışma kitabı ile ilişkilendirilmiş kodların hepsi çalışma kitabı içinde kaydedilen bir proje içinde saklanır. Proje gezgininde açık bulunan her çalışma kitabı, sayfalar, modül ve formlar bulunur. Bu ekrandan yönetilir.
- **Properties Window (Özellikler Penceresi):** Projedeki nesnelerin renk, başlık, boyut, yükseklik gibi özellikleri bu penceden düzenlenebilir.
- **Code Window:** Program komutlarının yazıldığı ekran



Editör Görünüm Seçenekleri

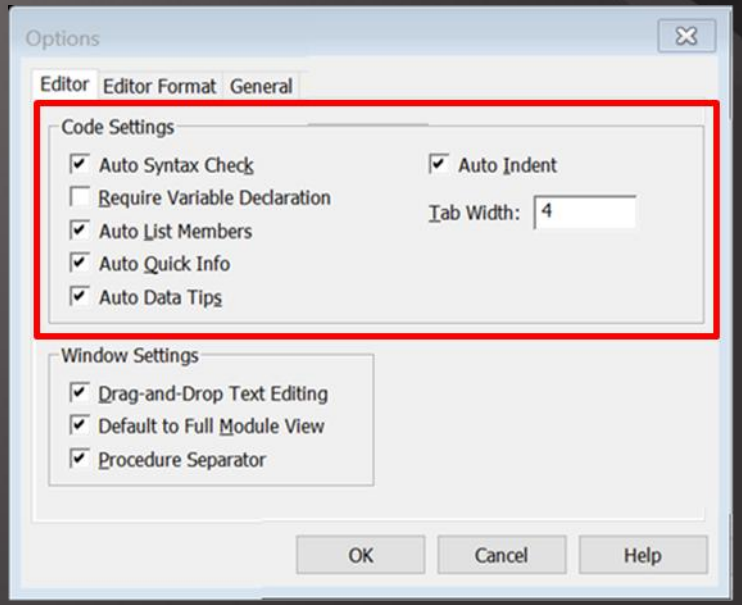
Editör görünürlüğünü View menüsündeki seçeneklerden değiştirebilirsiniz.

- **Object Browser:** Nesneleri görüntüler.
- **Immediate Window:** Debug.print ile yazdırlan değerleri görüntülemek için.
- **Locals Window:** Değişken tip ve aldığı değerleri görüntülemeyi sağlar.
- **Watch Window:** Değişkenleri bu alana sürükleyerek tip ve değerini görüntüler.



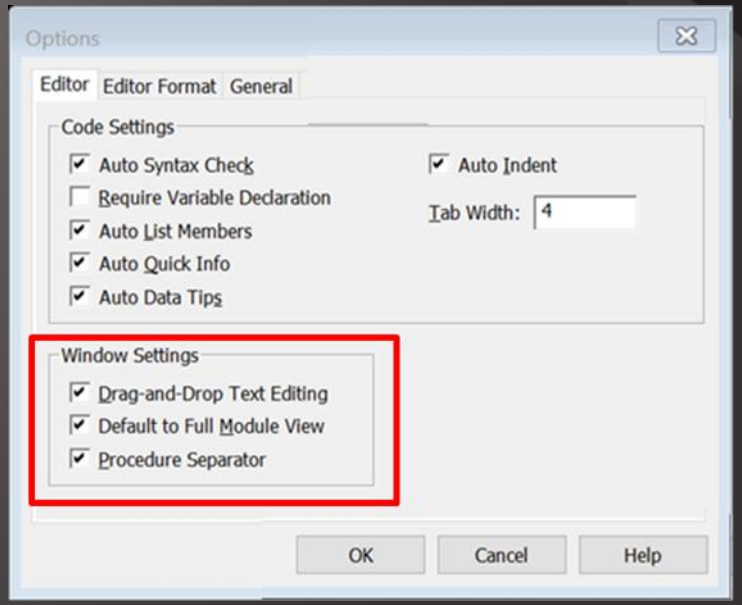
VB Editor Genel Seçenekler

- **Auto Syntax Check:** Bir kod satırı tamamlandığında söz dizilimi kontrol edilerek sorun hata mesajı ile belirtilir.
- **Require Variable Declaration:** Bütün değişkenlerin kullanılmadan önce dim deyimi ile deklare edilmeleri gerekliliğini belirtir. (Option Explicit)
- **Auto List Members:** Bir deyim yazılırken bu değimi tamamlayan öğeler listesi otomatik olarak belirir.
- **Auto Quick Info:** Bir fonksiyon adı yazıldığında fonksiyon parametreleri hakkında yardım bilgileri belirir.
- **Auto Data Tips:** İmlecin üzerinde bulunduğu değişkenin o anki değerini görüntüler.
- **Auto Indent:** Her yeni kod satırı otomatik olarak önceki satırla aynı şekilde girintili görünür.

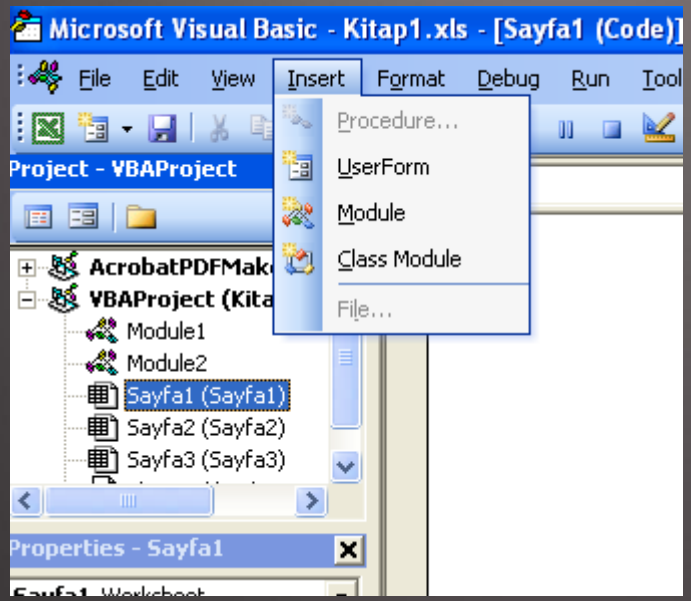


VB Editör Pencere Seçenekleri

- **Drag and Drop Text Editing:** Seçilmesi halinde, kod bölümlerinin sürekle bırak ile taşınmasına ve CTRL + Sürükle bırak ile kopyalanmasına imkan verir.
- **Default to Full Module View:** Seçimi halinde module ve form görünümlerinin ekranı kaplanmış olarak görüntülenmesini sağlar.
- **Procedure Separetar:** Seçili iken module içinde birden fazla procedure veya function olduğunda aralarına çizgi koyulmasını sağlar.



Kodlara Giriş

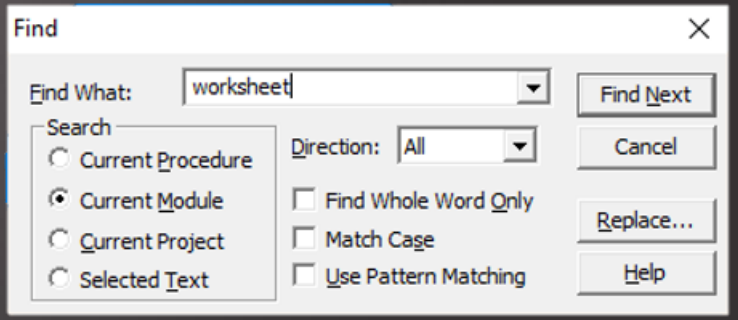


- **Kullanıcı Formu:** Görsel arabirim nesnelerine ait kodlar içerir.
- **Modül:** VBA yordam, fonksiyon ve veri tanımlamalarına ait kodlar içerir.
- **Sınıf Modülü:** Görsel arabirim sunmayan nesneler kümesine ait kodlar içerir.

VB Editörde Bul ve Değiştir

- Current Procedure: Seçili Fonksiyon veya Sub içinde.
- Current Module : Seçili modulde.
- Current Project : Seçili projede.
- Selected Text : Proje içindeki seçili aralıkta.
- Direction : Yukarı, aşağı veya hepsinde
- Find Whole Word Only : kelimenin tamamına eşitse
- Match Case : Büyük küçük harf duyarlı arama
- Use Pattern Matching : Desen biçim eşleşmeye duyarlı

CTRL+F : Bul
CTRL+H : Değiştir
F3 : Sonraki kayda git,
SHIFT+F3 : Önceki kayda git



Kod Yazarken

- **Açıklama Eklemek:** VBA'da tek tırnak karakterinden sonra gelen ifadeler birer açıklama olarak kabul edilir. Satırın başında olmak kaydıyla REM sözcüğü de açıklama satırı için kullanılır.
- **Komutun alt satırdan devamını sağlamak:** VBA kaynak kodu, satır başına bir deyim olacak şekilde yazılır. Bazen tek bir kod satırı çok uzun olabilir ve editörde bununla çalışmak kolay olmayabilir. Böyle durumlarda bir den fazla satıra bir deyimi bölmek gerekir.
- **Birden fazla satırı tek satırda toplamak:** Kodlar her satırda bir deyim olacak şekilde yazılır. Ancak tek bir satırda ":" işareti ile ayrılarak birden fazla deyim yazılabilir.

```
(General)
' Buraya açıklama yazabilirsiniz
Rem REM ile de açıklama satırı eklenebilir.
Sayi = 1 ' Açıklamalar bu şekilde de eklenebilir.

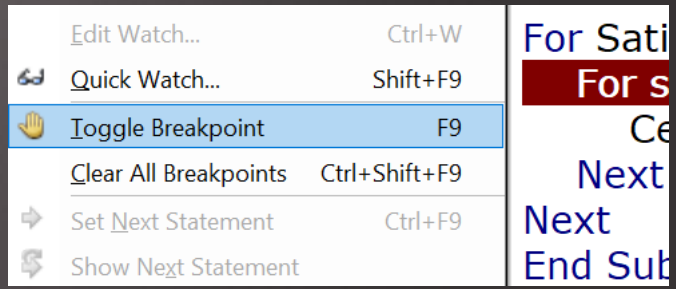
MsgBox _
"Kodun alt satırdan devamı '_' ile sağlanır."

If Sayi = 1 Then: MsgBox "BİR": End If
```

Break Point

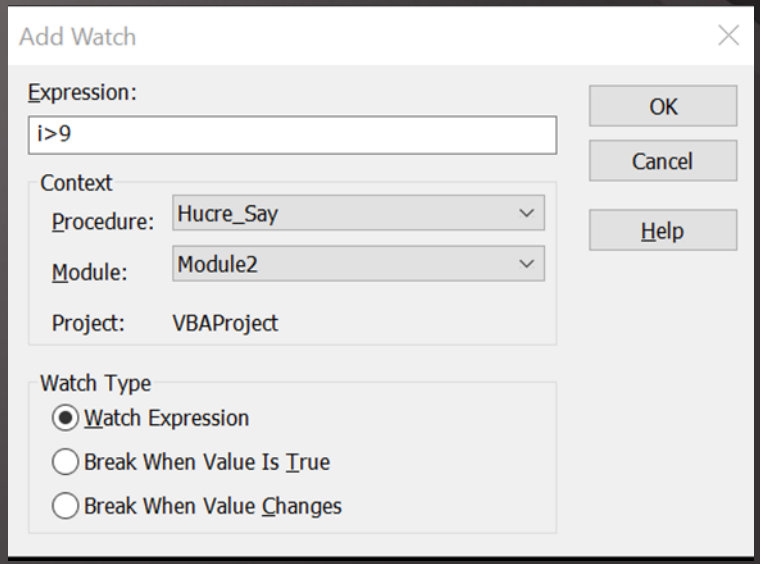
- Breakpoint, kodlarınızı kontrol etmek için belirli noktalara durak koymayı sağlayan araçtır.

- İzleme pencereleri ise, ister Break modunda olun ister olmayın kodunuzun o an tam olarak ne yaptığını gösterebileceği gibi tek satırlık bir kodun doğrudan burada çalıştırmaya imkan verir.
- Debug / Toogle Breakpoint (F9) ile ekleyebilir, Debug / Clear All Breakpoints (CTRL+Shift+F9) ile temizleyebilirsiniz.



İzleme Pencereleri

- Immediate Window:
  - Ctrl+G tuş bileşimi ile aktive edebilrsiniz. Debug.Print veya ?Degisken ile izleyebilirsiniz.
- Locals Window
  - Değişkenlerin tipleri ve aldığı değerleri gösterir.
- Watch Window
  - Değişkenlerin tipleri ve aldığı değerleri belli koşula göre gösterir.





Programlama

Klasik Programlama

Basic

Fortran

Pascal

Nesne Tabanlı Programlama

C#

VB.NET

JAVA

Nesne Kavramı



- **Properties** (Özellikler): Nesnelerin özellikleri vardır. Bunları öğrenerek veya değiştirerek programlama yapılır. Örneğin Range nesnesinin fontunun color özelliği hücredeki yazının rengini temsil eder.
- **Methods** (Metotlar): Nesne üzerinde eylem gerçekleştirir. Örneğin sheet nesnesinin clear metodu gibi.
- **Events** (Olaylar): Nesnelerin başına gelebilecek olayların programları tetiklemesi sağlanabilir.

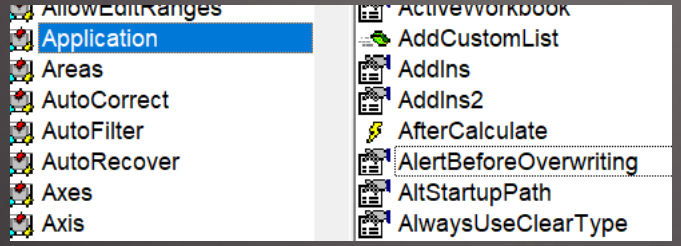
Properties

Methods

Events

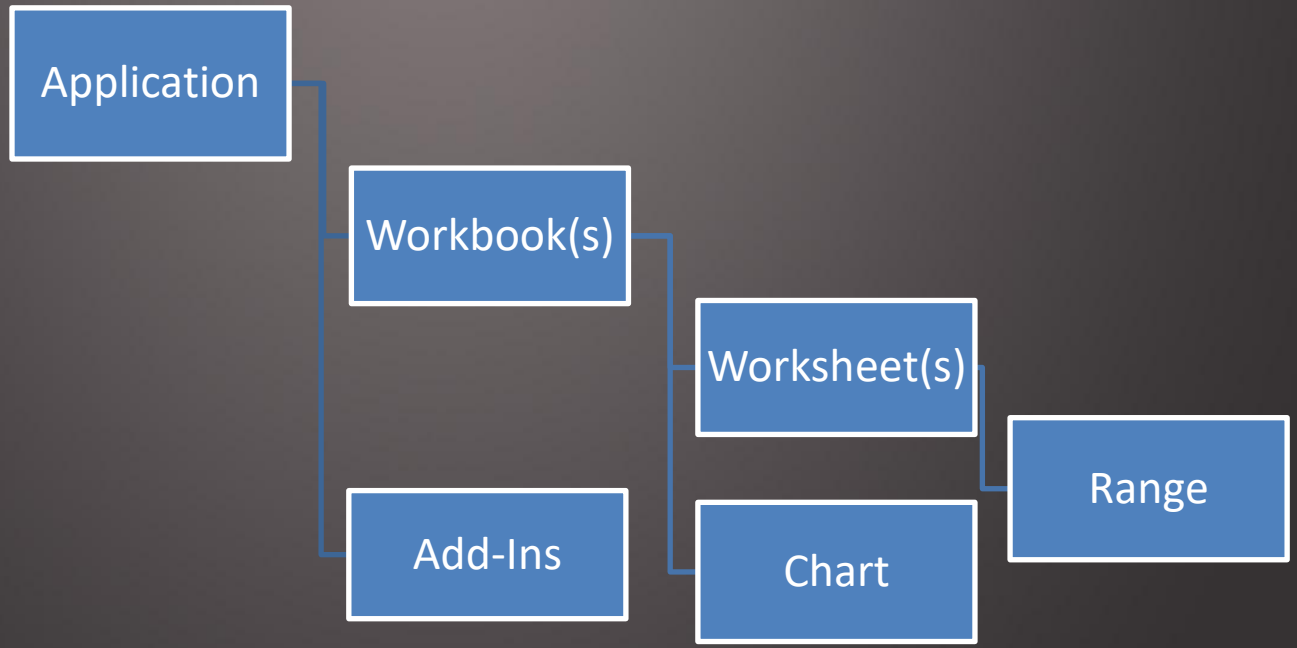
Collections (Kolesiyonlar): Bir nesnenin birden fazla alt nesnesini ifade eder. Örneğin çalışma kitabının alt nesnesi olarak birden fazla sheet bulunur. Bu nesne grubu Sheets koleksiyonu olarak adlandırılır

Excel Nesneleri



- Nesne yönelimli programlamada, her şey nesne olarak kabul edilir ve bütün işlemler nesne etrafında gerçekleşir. Excel'de bu nesneler çalışma kitaplıkları (workbook), çalışma sayfaları (worksheets), aralıklar ve hücrelerdir. Nesneler bir hiyerarşi içerisinde bulunur.

Excel’de Nesne Hiyerarşisi



Excel Nesne Yapısı

- **Application**: En üst düzeydeki nesne Application olarak adlandırılır. Application nesnesi uygulamayı temsil eder.
- **Workbook**: Workbook nesnesi açık durumdaki bir çalışma kitabını temsil eder. Excel'i normal olarak başlattığınızda uygulama (Application) otomatik olarak yeni ve boş çalışma kitabı açar.
- **Worksheet**: Excel'deki programlama çalışmalarının büyük kısmı doğrudan Worksheet nesnesi ile temsil edilen çalışma sayfaları ile ilişkilidir.
- **Range**: Range nesnesi aralık veya tek bir hücreyi, bir çalışma sayfasındaki bütün bir satırı yada sütünü veya çalışma kitabının farklı bölgelerinden seçilmiş hücre bloklarını temsil eder.
- **Chart**: İster bir çalışma sayfasına gömülü olsun, ister bir grafik sayfasında, bir çalışma kitabındaki her grafik bir Chart nesnesiyle temsil edilir.
- **Nesnelerin Parent Özelliğini Kullanmak:** Bir üst nesneyi öğrenmek için kullanılır.
  - Activesheet.Parent.Name

Range ve Cell Nesnesi

- Özellikler
  - Resize
  - Value
  - Value2
  - Formula, FormulaR1C1
  - Count
  - HasFormula

- Metodlar
  - Select
  - Activate
  - Offset
  - Delete
    - Shift:=**xlToLeft**
    - **Shift:=xlToUp**
  - NoteText Text: ".... "
  - ClearNotes
  - Clear …



Hücreler Hakkında

- Range Kullanımı
- […] Kullanımı
- Cells
- Item Kullanımı
  - Cells.item(1,1) - Cells.item(1,"A")
- Rows ve Columns Kullanımı
- Activecell ve Selection Kullanımı
- Offset Kullanımı
- Areas Kullanımı
- End Kullanımı
- CurrentRegion ve UsedRange(worksheet özelliği)

- SpecialCells(Type,Value)
  - **Type**
    - xlCellTypeAllFormatConditions
    - xlCellTypeAllValidation.
    - xlCellTypeBlanks
    - xlCellTypeComments
    - xlCellTypeConstants
    - xlCellTypeFormulas
    - xlCellTypeLastCell
    - xlCellTypeSameFormatCondition
    - xlCellTypeSameValidation
    - xlCellTypeVisible
  - Value
    - xlErrors
    - xlLogical
    - xlNumbers
    - xlTextValues

Aralık Seçmek

- Range("A1").Select
- Columns("A:A").Select
- Rows("1:1").Select
- Cells.Select
- Activecell.Currentregion.Select
- Selection.
  - EntireColumn.Select
  - EntireRow.Select
  - Selection.Count
  - Selection.Rows.Count
  - Selection.Areas.Count

Hücre Özellikleri

- Hücre Değerini Okumak ve Değiştirmek
  - Value
  - Value2
  - Text
- Hücre Rengini Öğrenmek ve Değiştirmek
  - Interior
  - Font
- Formül Yazmak
  - Formula
  - FormulaC1R1

```
Sub Hucre_Properties()
    Debug.Print Range("A1").Value
    Debug.Print Range("A1").Value2
    Debug.Print Range("A1").Text
    Debug.Print
    Range("A1").Formula
    Debug.Print
    Range("A1").Font.Color
End Sub
```

Hücreleri Doldurmak

- Range("B1:H1").FillRight
- Range("B1:H1").FillLeft
- Range("B1:B11").FillDown
- Range("B1:B11").FillUp
- Range("B1:B2").AutoFill Destination:=Range("B1:B20")
- Range("B1:B2"). AutoFill _ Destination:=Range("B1:B10"), Type:=xlFillSeries
- Range("B1").FlashFill



Sheet / Activesheet Nesnesi

- Sayfaları seçmek ve aktif etmek
  - Sheets(1).Select veya sheets(1).activate - Sheets.item(1).Select - Sheets(Array(1, 2)).Select
- Sayfaya isim verme - Sheets.name="SAYFAM"
- Sayfa Eklemek, Taşımak ve Kopyalamak
  - Sheets.add - Sheets.Add.Move after:=Sheets(**n**) - Sheets.Add.Move before:=Sheets(**n**) - Sheets(1).Move after:=Sheets(3)
- Gizlemek / Göstermek
  - Sheets(1).visible=true/false (xlSheetVisible, xlSheetHidden, xlVerySheetHidden
- Sayfayı Korumak (Protect – UnProtect)
- Sayfalar arasında gezinmek
  - Sheets(1).Next.Select, – Activesheet.Previous.Select
- Sheets.count
- **Sheet** ve **ActiveSheet** kavramı.

Workbook Nesnesi

- Workbook Nesnesinin Özellik ve Metotları
  - Workbooks.Open FileName:=" ….. ", ReadOnly:=True, Password:="1"
  - Workbooks.Close Savechanges:=True/False
  - Workbooks(1).Activate
  - Activeworkbook.save
  - Activeworkbook.saveas filename:=""
  - Workbooks.add
  - Workboos(1).HasPassword
  - Workbooks(1).Saved
  - Workbooks(1).ReadOnly
  - Workbooks(1).PrintPreview - PrintOut
- Activeworkbook – Thisworkbook - Me
- Workbook / Activeworkbook / Thisworkbook

Window Nesnesi

- Windows(1).Caption
- Windows(1).Activate – ActivateNext – ActivatePrevious
- Windows(1).Close
- Windows(1).NewWindow
- Windows(1).Zoom=100
- Windows(1).DisplayWorkbookTabs=True / False
- Windows(1).DisplayGridlines=True / False
- Windows(1).GridlineVolor=vbred
- Windows(1).DisplayFormulaBar = True / False
- Windows(1).DisplayStatusBar = True / False
- Windows(1).DisplayHorizontalScrollBar = True / False
- Windows(1).DisplayVerticalScrollBar = True / False
- Windows(1).FreezePanes = True / False



Application Nesnesi

- Application.Username
- Application.DisplayFullScreen=True/False
- Application.DisplayFormulabar=True/False
- Application.DisplayStatusbar=True/False
- Application.DisplayScrollsbar=True/False
- Application.DisplayAlerts=True/False
- Application.Screenupdating=True/False
- Application.StatusBar
- Application.Caption
- Application.OperatingSystem
- Application.Activeprinter
- Application.Version
- Application.RecentFiles

- Application.Close
- Application.Quit
- Application.Calculate
- Application.Undo

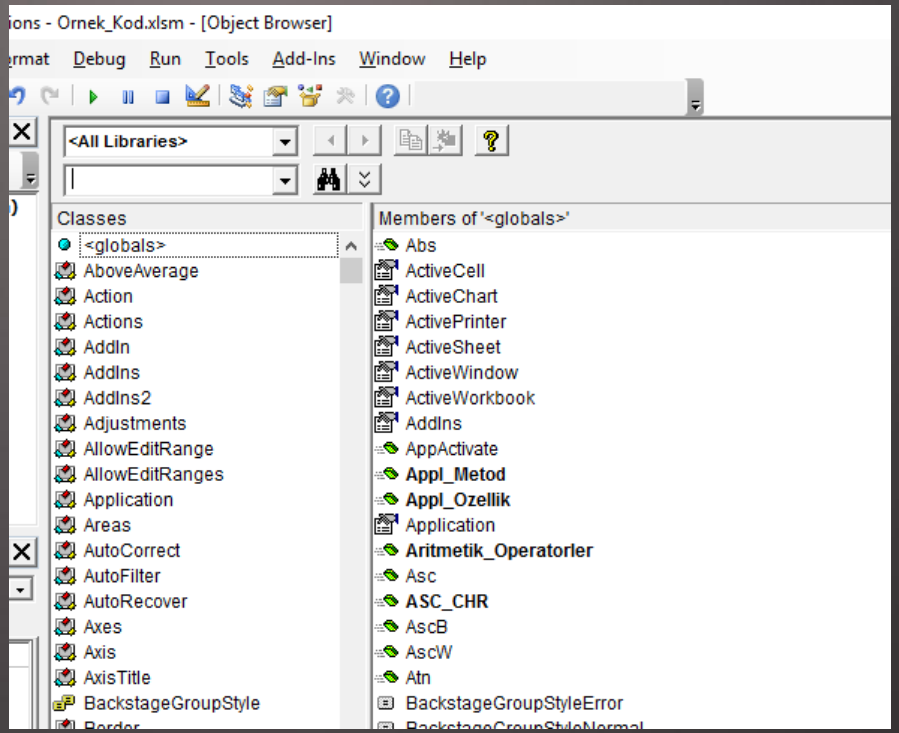
- Application.worksheetfunction.sum (range("A1:B5"))

- Application.GetOpenFileName (Title:="Dosya Seç", FileFilter:="Excel Files *.xls* (*.xls*),")

Sayfa Örnek Uygulaması

```
Sub Sayfa_Sırala()
Dim i As Integer Dim j As Integer
If Worksheets.Count = 1 Then Exit Sub
        For i = 1 To Worksheets.Count - 1
                For j = i + 1 To Worksheets.Count
                        If Worksheets(j).Name < Worksheets(i).Name Then
                        Worksheets(j).Move before:=Worksheets(i)
                        End If
                Next j
        Next i
End Sub
```

Object Browser

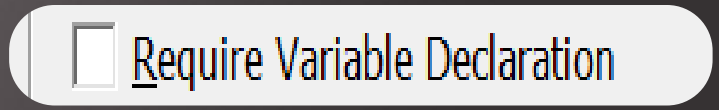


With ... End With

```
With Range("C3")
        .Value = "Semih"
        .Interior.ColorIndex = 3
        .Font.ColorIndex = 6
        .Font.Size = 20
        .Font.Bold = True
        .Font.Italic = True
End With
```

Sabit ve Değişkenler

- **Sabitler:** Const anahtar sözcüğü ile sembolik bir sabit oluşturulur.
  - Const PI_SAYISI As Double=3.14
- **Değişkenler:**
  - Programın çalıştırılması esnasında veri yada nesne referanslarının saklanabileceği adlandırılmış içeriklere değişken(variable) denir.
  - Eğer Option Explicit deyimi kullanılmışsa değişkenin kullanılmadan önce tanımlanması zorunludur. Bu durum hata yapmayı önleyecektir.



Enum Sabitleri

```
(Declarations)
Private Enum Bolgeler
akdeniz '0
Karadeniz '1
Marmara '2
Ege '3
Diğer '4
End Enum
```

- **Enum Sabitleri (Enumeration Constant)**
  - vb ile başlayanlar: VBA library'sinin sabitleri
  - xl ile başlayanlar : Excel library'sinin saibtleri
  - mso ile başlayanlar: Office library'sini sabitleri
- Ayrıca Declaration bölümünde kendi Enum sabitlerinizi tanımlayabilirsiniz.

Değişkenlere İsim Verme Kuralları

- İlk karakterlerin harf olması gerekir.
- Boşluk karakteri ve .,!,@,&,$,# karakterleri kullanılmaz.
- İsmin uzunluğu 255 karakteri geçemez.
- VBA anahtar sözcükleri veya fonksiyonları kullanılmaz. (Not, if, select, Range vs.)
- Türkçe karakterleri kullanmayınız.
- Anlaşılır ve açıklayıcı olması rahat ve kullanışlıdır.

Değişkenlerin Kapsam ve Ömrü

- **Yerel Değişkenler:** Dim sözcüğü ile yordam veya fonksiyon içinde tanımlanır ve sadece bu yordam içinde yaşar.

- **Modül Düzeyi Değişkenler**: Dim ve Private ile tanımlandığında sadece o modülde kullanılabilir. Public ile tanımlandığında ise tüm Project içinde kullanılabilir.

- **Static Değişkenler**: Static anahtar sözcüğü ile tanımlanan değişkenler tanımlandığı bu yordama ait olmakla birlikte her yordamın yeniden çalıştırılması sırasında yeniden oluşturulmaz. Bir kere oluşturulur ve bu değişken ile işleme devam edilir.

Sayısal veri tipleri

İsim	Tip	Aralık	Boyut
Byte	Tamsayı	0 - 255	1 byte
Integer	Tamsayı	-32.768 - 32.767	2 byte
Long	Tamsayı	-2.147.483.648	4 byte
Single	Kayan nokta	-3.4x1038 - 3.4x1038	4 byte
Double	Kayan nokta	-1.79x10308 - 3.4x10308	8 byte
Currency	Kayan nokta	-9.22x1011 - 9.22x1011	8 byte

Sayısal Olmayan Veri Tipleri

İsim	Boyut
String	Max 65.400
Date	8 byte
Boolen	1 bit
Object	4 byte
Variant	16 byte

- Metin veya karakter katarı (string) veri tipi değişken uzunlukta katardan oluştuğu için deklarasyon esnasında büyüklüğü tanımlamak gerekmez.
- Dim isim As String
- Sabit uzunluktaki bir karakter katarı deklare edildiğinde sabit bir büyüklüğe sahip değişken tanımlanmış olur.
- Dim adres As String*20 adres değişkeni 20 karakter uzunluğundadır.

Mantıksal Veri Tipi

- Sadece iki değer (true / false) alabilen verileri tutmak için Boolean değişken tipinden faydalanılır.

  – Dim Cinsiyet As Boolean
  – Cinsiyet=True

Tarihsel Veri Tipi

Tarih (date) terimi saat ve takvim tarihi anlamına gelir. Tarihler kayan noktalı sayılar olarak saklanır.

```
Dim Dogum_Tarihi As Date

Dogum_Tarihi=«22.01.1990»
```

Nesne Veri Tipi ve Nesne Tanımlama

- Nesne Veri Tipi: Bir nesneye ait referansları tutar. Erken bağlama veya geç bağlama olarak adlandırılan yöntemler ile iki şekilde kullanılır.
  - Dim nesne As Object

Dizilerle Çalışmak

- Diziler aynı amaç için kullanılan ve dolayısı ile da aynı türden olan çok sayıdaki değişkeni tek bir değişkenmiş grubu içinde kullanmaya imkan tanıyan yapılardır. Bu yapılar sayesinde döngü komutları ile bu değişken içindeki tüm elemanlara seri ve sistematik bir şekilde erişilebilir. Sheet üzerinden okumaya oranla 500 kata kadar daha performans artışı sağlar.
- Dim ogrenci(25) as string (Bu ifade 25 elemanlı bir string dizisi tanımlar. Yani 25 tane ad-soyad bilgisi tutabilecek bir dizi tanımlanmıştır.)
- Dim matris(4, 8) as integer (Dört satır, sekiz sütundan oluşan bir tamsayı değişkenleri tanımlamıştır.)
- Dim veriler (4,6,8) as double (Dört satır altı sütun ve bunun gibi sekiz düzlemden oluşmuş bir küp tanımlar.)
- Dim Gunler(1 to 12, 1 to 31) As String
- Gunler(5, 15)="Pazar"

Dizi Tanımlamak

- Statik Diziler: Dizinin eleman sayısı başta belli olduğu dizilerdir. Statik diziler dim ay(12) şeklinde tanımlanır.
- Dinamik Diziler: Dizinin boyutu ve kaç elemanlı olacağı belli değilse dinamik dizi tanımlanır. dim calisanlar() şeklinde tanımlanır.

- Dizi değişkenler, redim ifadesi ile genişletilir. Tanımlama sırasında redim ifadesi ile beraber preserve ifadesi de kullanılırsa dizi içerisinde daha önceden yer alan değerler korunmuş olur.
- Redim veri(50) as integer
- Redim preserve veri(100) as integer
- Erase Veri Dizi içlerini boşaltır.
- Lbound, Ubound: Ubound dizideki indisin ilk değerini, LBound ise dizideki indisin son değerini bulur.

```
Dim Aylar(2) AS String
Aylar(0)="Ocak"
Aylar(1)="Şubat"
Aylar(2)="Mart"

Redim Preserve Aylar(3)
Aylar(3)="Nisan"

For i=Lbound(Aylar) TO
Ubound(Aylar)
        Debug.Print Aylar(i)
Next
```

Variant Veri Tipi

- VBA'nın en esnek değişken tipidir. Bir değişkeni bir tip belirtmeden deklare ederek variant değişken oluşturulabildiği gibi As Variant olarak da oluşturulur.
  - Dim xxx

    veya
  - Dim xxx As Variant
- Bu veri tipinin esnekliği her tipte veriyi tutabilmesinden kaynaklanır. Sayıları, metinleri, tarihleri, dizileri tutabilir. Ancak esnekliği nedeniyle hafızada daha fazla yer tutar.

Değişkenlerin Türünü Öğrenmek

- Vartype
  - vbInteger 2
  - vbLong 3
  - vbSingle 4
  - vbDouble 5
  - vbCurrency 6
  - vbDate 7
  - vbString 8
- Isnumeric
- IsDate
- IsArray

```
Sub dene()
Dim Tarih As Date

Tarih = "15.05.2019"
MsgBox IsDate(Tarih)

Tipi = VarType(Tarih)
End Sub
```

Kısaltmalarla Değişken Tanımlama

- ABBREVIATIONS (kısaltmalar):
  - Değişken adının sonuna özel bir karakter ekleyerek bir değişken türü belirlenebilir.

- Dim sayi% 'Integer
- Dim uzunsayi& 'Long
- Dim toplam! 'Single
- Dim alttoplam# 'Double
- Dim odeme@ 'Currency
- Dim adi$ 'String
- Dim enuzunsayi^ ' 64 bit'te LongLong

DEF ile Değişken Tanımlama

- Declaration alanında DEF ile baş harflerine belli bir tür tanımlaması yapılabilir.

- DefInt i
- DefStr s
- DefDate d
- DefDbl b
  - iSayi=0
  - sSehir=""
  - dDTar=#01/01/2021#
  - bFiyat=10,5

Değişken Tür Dönüştürme

- cDec
- cDate
- cLng
- cDbl
- CBool
- CByte
- CCur
- CInt
- CSng
- CStr
- CVar



Operatörler

Diğer programlama dillerinde olduğu gibi, VBA dilinde de, gerek aritmetik işlemleri yapabilmek için gerekse de mantıksal karşılaştırmaları veya mantıksal işlemleri yapabilmek için operatörler bulunur. Bu operatörlerin çoğunluğu bir çok programlama dilinde aynı olmakla birlikte ufak tefek farklılıklar bulunabilir.

- Aritmetik Operatörler
- Karşılaştırma Operatörleri
- Mantıksal Operatörler
- Birleştirme Operatörü (&)



Aritmetik Operatörler

- **^** Üs (exponentiation)
- ***** Çarpma (multiplication)
- **/** Bölme (division)
- **** Tamsayıya bölme(integer division)
- **Mod** Bölümden kalan (modulus arithmetic)
- **+** Toplama (addition)
- **-** Çıkarma (subtraction)
- **&** birleştirme(string concatenation)
- **( )** Parantez

Karşılaştırma Operatörleri

- = Eşit (Equality)
- <> Eşit değil (inequality)
- < Küçük (less then)
- > Büyük (greather then)
- <= Küçük eşit (less then or equal to)
- >= Büyük eşit (greather then or equal to)
- **Like** Benzer (?, *, #, [], [!])
- **Is** Var

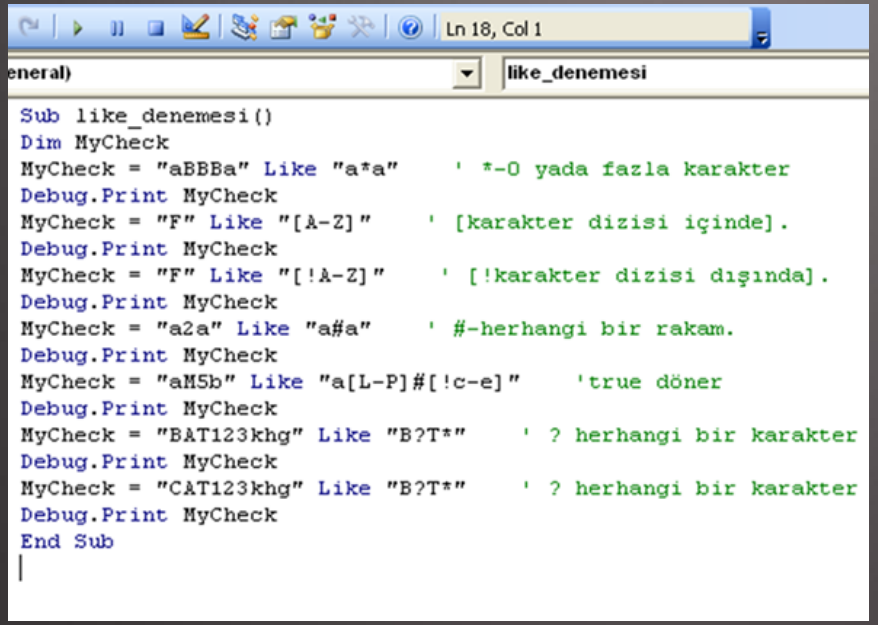
Mantıksal Operatörler

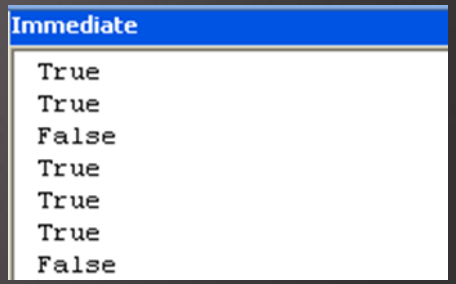
- **Not**
- **And**
- **Or**
- **Eqv**: mantıksal iki ifadenin karşılaştırması için kullanılır. Sonuçlar aynıysa true sonucu verir.
- **Xor**: or işlecinin tersi.
- **Imp**: Koşul1 True (1), Koşul2 False(0) ise False değilse **True** sonucunu verir.

Karşılaştırma Operatörü LIKE Kullanımı

- **?** : Herhangi bir karakter
- **#** : Herhangi bir rakam
- ***** : Bir veya daha fazla karakter
- *[aralık] :* Verilen aralıkta bir karakter
- *[!aralık]:* Verilen aralık dışında bir karakter

Like Kullanımına Örnekler



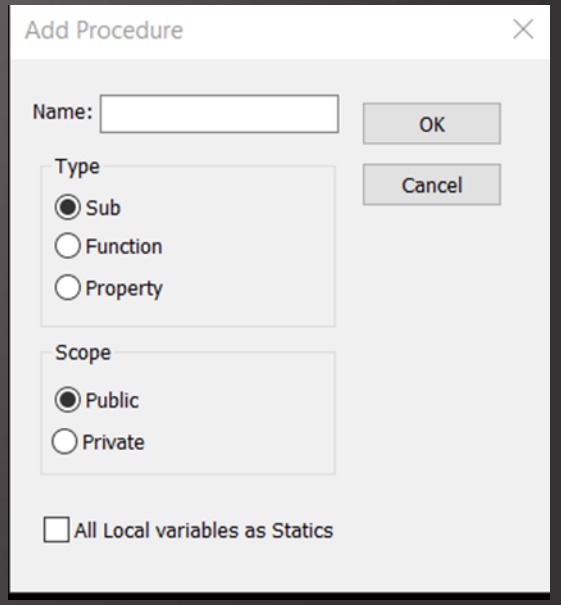


Kod Nereye (Tekrar)

- **Workbook Modülleri**: Çalışma kitabını ilgilendire veya Çalışma kitabı olaylarına bağlı olarak çalışmasını istediğiniz kodlar bu bölüme yazılır. Çalışma kitabı ile ilgili bir takım olaylarda çalışması gereken kodlar var ise bu bölümdeki ilgili EVENT bölümlerine bu kodlar yazılır.
- **Worksheet Modülleri**: Çalışma kitabına eklenen sayfa sayısı kadardır. Sayfayı ilgilendiren kodlar ilgili modüle yazılırlar.
- Form Modülleri: Eklenen form nesnesi içine yazılır.
- **Serbest Modüller**: Tüm çalışma ortamını ilgilendiren kodlar serbest modüllere yazılır.

Procedure Hakkında

- Procedure Çeşitleri
  - Sub
  - Function
  - Property
- Kapsamı
  - Public
  - Private



Procedure & Fonctions

```
Call Altyordam(Arg1, Arg2)
Sonuc = FonkAdi(Arg1, Arg2)
```

```
Sub AltyordamAdi(argümanlar)
    ..... .....
    ..... .....
    Kodlar
    ..... .....
End Sub
```

```
Function Fonk_Adi(argümanlar)
.....
Kodlar
.....
    FonksiyonAdi = dönen_değer
    .....
End Function
```

- Alt yordamlar diğer yordamlardan ismi ile veya CALL deyimi ile çağırılır.
- Fonksiyonlar alt yordamlardan farklı olarak geriye bir değer döndürür.

Alt Procedure ve Argüman Kullanımı



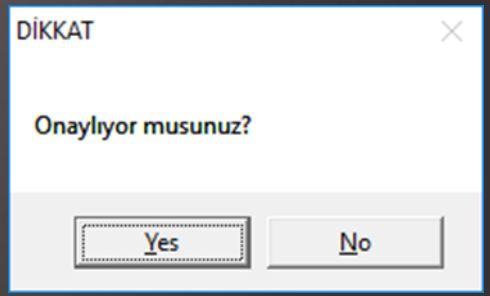

```
Sub yordamOrnek(isim as string, Optional sicil As integer=0)

....

End Sub
```

MSGBOX

- Kullanıcıya mesaj vermek için kullanılır. Kullanıcının seçimi dönen değeri öğrenmek mümkün olur.
- Sonuc=MsgBox(prompt[, buttons] [, title] [, helpfile, context])
  - prompt: İletişim kutusunda görüntülen metindir.
  - Buttons: iletişim kutusunda olmasını istediğiniz seçenekleri belirler.
  - title: İletişim kutusunun başlık çubuğunda gösterilmesi istenen metin
  - helpfile, context: İletişim kutusuna bağlı yardım dosyası ve içeriğin help dosyasındaki yerini gösterir.



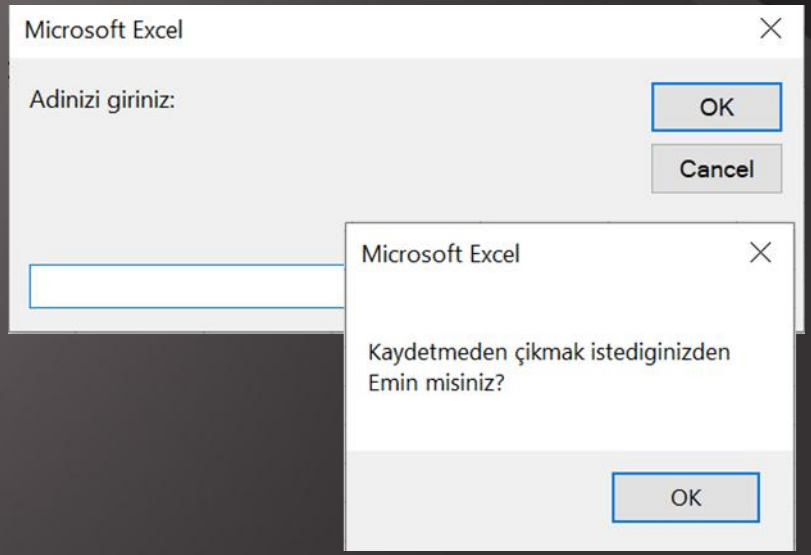


INPUTBOX

- InputBox(prompt, title, default, xpos, ypos, helpfile, context)
  - prompt: İletişim kutusunda görüntülen metindir.
  - title: İletişim kutusunun başlık çubuğunda gösterilmesi istenen metin
  - default: varsayılan yanıt metni
  - xpos, ypos: İletişim kutusunun ekranın sol üst köşesinden itibaren twip cinsinden konumu (1.44 twip 1 inch)
  - helpfile, context: İletişim kutusuna bağlı yardım dosyası ve içeriğin help dosyasındaki yerini gösterir.
- Sonuc = InputBox(Message, Title, , , , "DEMO.HLP", 11)
- Application.Inputbox şeklinde kullanım farkı.
  - Type belirlenebilir. --Application.Inputbox(prompt:="...",type:=1)
  - Giriş hücreden seçilebilir.
  - Formulalocal kullanılabilir. (Activecell.FormulaLocal=inputboxdan(prompt:="…",type:=0)

İletişim Kutularına Satırbaşı Yapmak

- **vbCrLf, vbCr, vbLf, vbNewLine veya Chr(13):** MSGBOX veya INPUTBOX içindeki mesajı alt satıra taşır.



IF Blok Kontrol Deyimi

```
If şart Then Blok1 Else Blok2
```

```
If şart Then
    Blok
End If
```

```
If şart Then
    Blok1
ElseIf Şart2 Then
    Blok2
ElseIf Şart3 Then
    Blok3
Else
    Blok4
End If
```

```
Sonuc = IIf(Şart, Doğruysa, Yanlışsa)
```

```
If condition Then
    [ statements ]
[ ElseIf condition-n Then
    [ elseifstatements ]]
[ Else
    [ elsestatements ]]
End If
```

Select Case Blok Kontrol Deyimi

VB

```
Dim Number
Number = 8     ' Initialize variable.
Select Case Number     ' Evaluate Number.
Case 1 To 5     ' Number between 1 and 5, inclusive.
    Debug.Print "Between 1 and 5"
' The following is the only Case clause that evaluates to True.
Case 6, 7, 8     ' Number between 6 and 8.
    Debug.Print "Between 6 and 8"
Case 9 To 10     ' Number is 9 or 10.
Debug.Print "Greater than 8"
Case Else     ' Other values.
    Debug.Print "Not between 1 and 10"
End Select
```

Goto Blok Kontrol Deyimi

- Kod içinde başka bir kod bloğuna gitmek için kullanılır.
  - Tanımlama şekli:

    GOTO GidilecekYer
  - Gidilecek kod bloğu

    GidilecekYer:

Hata Denetimleri

- Yazım hataları (Syntax Error)
- Derleme hataları(Compile error):
  - Next yazılmamış bir For döngüsü
  - Olmayan bir prosedürü çağırmak
- ÇalışmaZamanı hataları(Runtime error):
  - 0'a bölme,
  - Olmayan bir sayfaya/dosyaya erişmek

- Compile etmek
- Hata yakalamak
  - On Error Resume Next
  - On Error GoTo etiket
  - On Error GoTo 0
  - On Error GoTo -1

Kontrol Fonksiyonları

- iif([sınama], [doğruysa değer], [yanlışsa değer])
- Choose([Değer], "BİR", "İKİ", "ÜÇ,…)
- Switch([sınama1], [sonuç1], ([sınama2], [sonuç2]…)

For … Next Döngüsü

```
'FOR ... Next Döngüsü Kullanımı
For SayacSayi = ilkDeger To BitisSayi Step ArtisSayi
    '''''''''Kodlar buraya
    '''''''''
     Exit For '    İhtiyaç halinde
Next SayacSayi   ' SayacSayi kullanımı isteğe bağlı
```

```
'For Each ... Next Döngüsü Kullanımı
For Each SayacSayi In Koleksiyon
    '''''''''Kodlar buraya
    '''''''''
     Exit For '    İhtiyaç halinde
Next SayacSayi   ' SayacSayi kullanımı isteğe bağlı
```

DO – WHILE Döngü Deyimi

```
Do While Condition
        .....
        [Exit Do]
Loop
```

```
Do Until Condition
        .....
        [Exit Do]
Loop
```

```
While Condition
        .....
Wend
```

```
Do
        .....
        [Exit Do]
Loop While Condition
```

```
Do
        .....
        [Exit Do]
Loop Until Condition
```

Do - Loop While Döngüsü

```
Do While Kosul
    ' Kodlar buraya
    [Exit Do]
Loop


Do
    ' Kodlar buraya
    [Exit Do]
Loop While Kosul
```

- **Do While:** Koşul gerçekleştiği sürece çalışır. Koşul gerçekleşmezse döngüye hiç girmez.
- **Loop While:** Koşul gerçekleştiği sürece çalışır ama koşul gerçekleşmese bile bir kez mutlaka döngü içindeki komutlar çalıştırılır.

Do - Loop Until Döngüsü

```
Do Until Kosul
    ' Kodlar buraya
    [Exit Do]
Loop

Do
    '  Kodlar buraya
    [Exit Do]
Loop Until şart
```

- **Do Until:** Koşul gerçekleşinceye kadar çalışır, Koşulun doğru olması halinde döngüden çıkar. Koşul gerçekleşmezse döngüye hiç girmez.
- **Loop Until:** Koşul gerçekleşinceye kadar çalışır, Koşulun doğru olması halinde döngüden çıkar. ama koşul gerçekleşmese bile bir kez mutlaka döngü içindeki komutlar çalıştırılır.

While - Wend Döngüsü

- Do döngüleri daha kullanışlı ve esnek bir yapı sunar. Bu nedenle While – Wend genellikle tercih edilmez. Sadece koşula bağlı olarak çalışır, For ve Do döngülerinde olduğu kullanılan Exit deyimi ile döngüden çıkış burada çalışmaz.

```
While Kosul
    ' Kodlar buraya
Wend
```

Döngü Örnek Uygulama

```
'Doğrusu Girilene Kadar Şifre soran örnek
Sub SifreSor()
Do
    Sifre = InputBox("parola")
    If Sifre = "123" Then
            Exit Sub
        Else
            MsgBox "hatalı şifre"
    End If
Loop
End Sub
```

Metin Fonksiyonları

- LCASE
- UCASE
- LEFT
- RIGHT
- MID
- LEN
- LTRIM
- RTRIM
- TRIM
- RND, RANDOMIZE
- STR
- VAL
- SPACE
- STRING
- STRCOMP
- ASC
- CHR
- INSTR
- INSTRREV
- STRREVERSE

Sayısal Fonksiyonlar

- INT
- FIX : (int ‘den farkı negatif sayılarda önceki sayıya)
- ROUND

Tarih Fonksiyonları

- Date
- Now
- Time
- Day
- Month
- year
- weekday

Kontrol Fonksiyonları

- isnumeric
- isdate
- islogical
- isempty
- isnull
- iserror
- isobject

Sort Metodu

- Range.Sort
  - Key1:=Range("A1"),
  - Order1:=xlAscending veya xlDescending
  - Header:=xlYes veya xlNo
  - Orientation:=xlSortColumns veya xlSortRows

Find, FindNext, FindPrevious Metotları

- Range.Find
  (
    - What:=Aranacak değer
    - After:=Aramaya başlanacak yer
    - Lookin:= (xlFormulas, xlValues, xlComments)
    - Lookat:= (xlWhole, xlPart)
    - Serchorder:=(xlByRows, xlByColumns)
    - SearchDirection:=(xlNext, xlPrevious)
    - MatchCase:= (True, False)

)

AdvancedFilter Metodu

- Range.AdvancedFilter
  (
    - Action:= (xlFilterCopy, xlFilterinplace)
    - CriteriaRange:=(Range)
    - CopyToRange:=(Range)
    - Unique:=(True/False)
)

Dosya ve Klasörler



Dosya Erişimi

- Verileri Okumak
- Dosyaya Yazmak
- Dosyaya Veri Eklemek

Dosya Yönetimi

- Dosya Oluşturmak
- Dosya Silmek
- Dosya Taşımak

- CURDIR
- RMDIR
- MKDIR
- CHDIR
- CHDRIVE
- FILECOPY
- NAME
- KILL
- DIR
- FILELEN
- FILEDATETIME
- GETATTR

FilesytemObject ile Dosya ve Klasör

- Referanslara Ekleme
  - Microsoft Scripting Runtime
- Nesne Tanımlama
  - Dim Fso As Scripting.FilesystemObject
  - SET Fso = New Scripting.FilesystemObject

Klasör İşlemleri

- Klasör İşlemleri
  - Fso.FolderExists("…")
  - Fso.CreateFolder("…")
  - Fso.DeleteFolder("…")
- Klasör tanımlama (klasördeki dosyalar)
  - Dim **Klasor** As Scripting.Folder
  - Dim **Dosya** As Scripting.File

  - SET **Klasor**=Fso.GetFolder("…")
    - // FOR EACH **Dosya** IN **Klasor**.Files

Dosya İşlemleri

- Dim Dosya As Scripting.File

- Dosya işlemleri
  - Fso.FileExists("…")
  - Fso.FileCopy Source:="…", Destination:="…", OverwriteFiles:=True/False
  - Fso.DeleteFile("…")

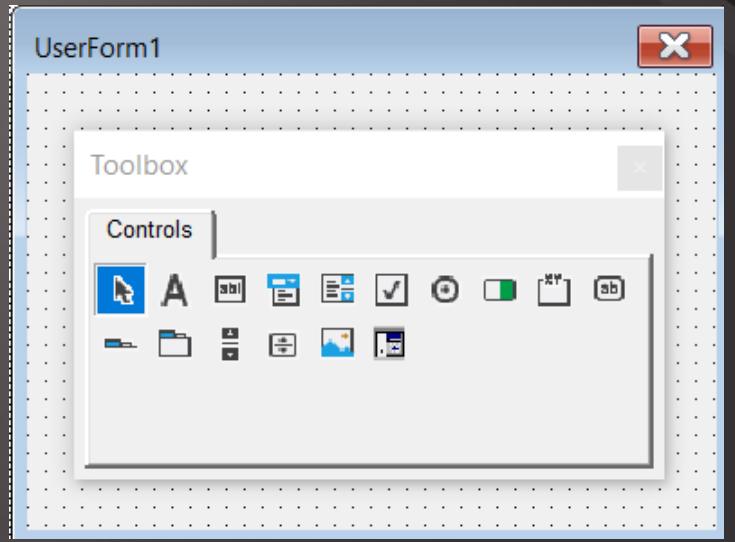
  - SET **Dosya**=fso.GetFile("…")
  - Debug.Print **Dosya**.Name, **Dosya**.Path, **Dosya**.DateCreated, **Dosya**.DateModified, **Dosya**.Type, **Dosya**.Size

Sıralı Erişimli Dosya İşlemleri

- Open “Dosya Adı” For AçmaModu [Access ErişimModu] [Lock] As [#]DosyaNo[Len=KayıtUzunluğu]
  - **Dosya Adı**: Açılacak veya oluşturulacak dosyanın adı.
  - **Açma Modu:** Dosyanın amaca uygun olarak Append, Output, Input, Binary veya Random modlarından birinde açılabilir.
  - **Erişim Modu:** Açma moduna uygun olmak kaydıyla dosyanın Read, Write, or Read Write amaçlı kullanılmasıdır. Belirtme olmadığında Read Write modu geçerlidir.
  - **Lock**: Açma ve Erişim moduna uygun olacak şekilde dosyanın Shared, Lock Read, Lock Write, ve Lock Read Write paylaşım ve kilit özelliğidir. Aksi belirtilmediğinde Shared modu geçerlidir.
  - **DosyaNo**; Birden fazla ile aynı anda çalışılacaksa (en fazla 511 adet) her biri bir numara ile çağrılır.
  - **KayıtUzunluğu**; Açılan dosya dan okunacak ve dosyaya yazılacak kayıt uzunluğu belirlenir. (en fazla 32767 bytes)

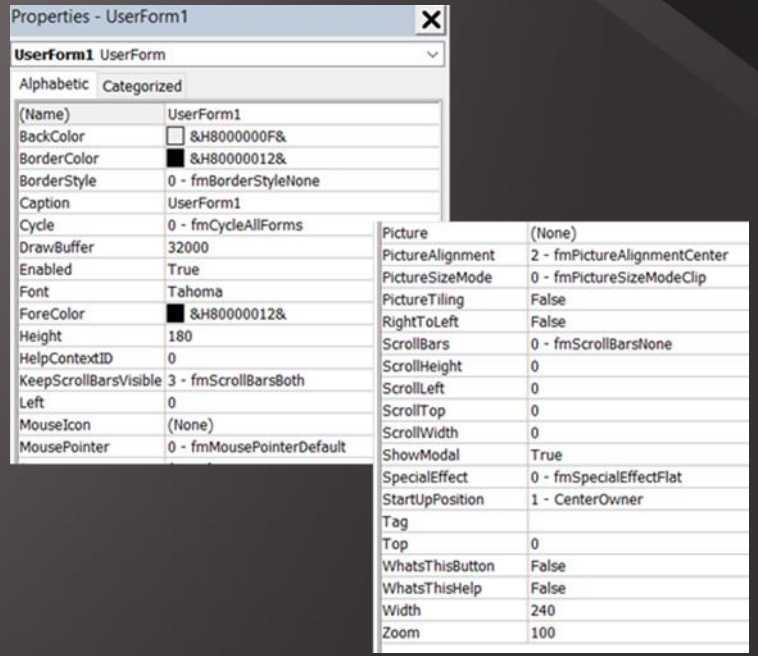
Form Kontrol Elemanları

- Label : Etiket
- TextBox: Metin kutusu
- ListBox: Liste kutusu
- ComboBox: Açılır liste kutusu
- CheckBox: Onay düğmesi
- OptionButton: Seçenek düğmesi.
- ToggleButton: Açık / Kapalı
- Frame: Kontrolleri gruplamak için kullanılır.
- CommandButton: Komut Düğmesi
- Image: Resim kutusu



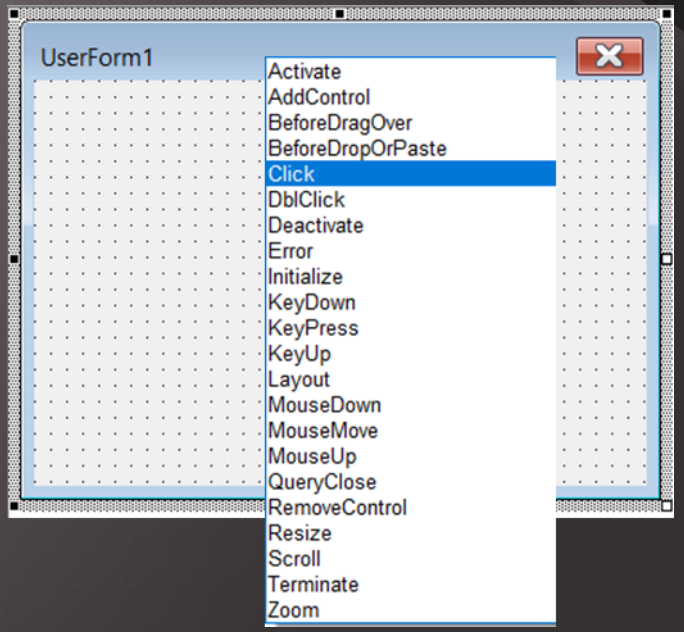
Kullanıcı Formu Özellikleri

- Caption: Formun başlık kısmında görüntülen metin.
- Height: Punto cinsinden formun yüksekliği (1 punto = 1/72 inç)
- MousePointer: Formun üzerine geldiğinde fare imlecinin görüntüsü
- Name: Koleksiyona başvurmak için kullanılan isim
- Picture: Arka plan resmi olarak kullanılacak resim dosyası
- PictureAlignment : Bir resmin formdaki yerini belirler.
- PictureSizeMode: Resim büyüklüğünün nasıl değiştirileceğini belirtir.
- ShowModal: True ise uygulamanın diğer bölümleri çalıştırılmadan önce form kapatılmalıdır. False ise uygulamanın diğer kısımları çalışırken form ekranda kalabilir.



Bazı Form Olayları

- Click: Kullanıcının ilgili nesneye mouse'nin sol tuşuna veya enter tuşuna basılması ile oluşur. CheckBox kontrolünün değeri boşluk çubuğuna basıldığında değişir. Kullanıcı bir nesneye tıkladığında önce MouseDown ve MouseUp olayları gerçekleşir.
- DblClick: Bir nesne çift tıklandığında gerçekleşir. DblClick olayını tekrar kullanmak için Cancel değeri yordam içinde True olarak ayarlanır.
- KeyDown, KeyUp, Keypress : Herhangi bir tuşa basıldığında veya bırakılması olayıdır. Keycode basılan tuşa ait tuş kodunu veren bir tamsayıdır. Keypress: Yazdırılabilir bir tuşa basılması olayı.
- MouseDown, MouseUp, MouseMove: MouseDown, mouseun tuşu basılınca MouseUp, mouseun tuşu bırakıldığında MouseMove ise formun üzerinde mouse gezerken meydana gelir.

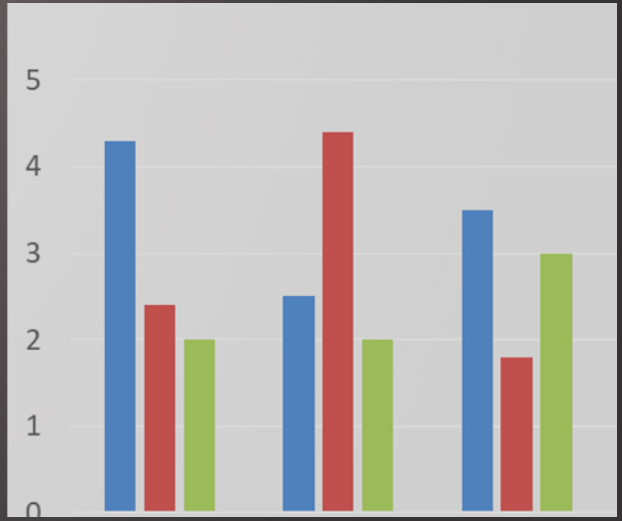


Bazı Form Olayları

- Activate - Deactivate: Formun aktif ve deaktif edildiğinde meydana gelir.
- AddControl – RemoveControl : Forma kodlar aracılığı ile, form elemanı eklenmesi halinde meydana gelir.
- Error : Form üzerinde herhangi bir hata oluşması halinde error olayı meydana gelir..
- Exit : Form elemanlarından çıkıldığında meydana gelir.
- Initalize : Formun açılışı esnasında bu olay meydana gelir.
- Change: Bir nesnenin value özelliği değiştiğinde oluşur.
- Terminate: Form kapanırken meydana gelir.
- QueryClose: Terminate olayından önce gerçekleşir.
- Resize: Formun boyutunun değiştiğinde meydana gelir.

Grafiklerle Çalışmak

```
Sub grafik()
    Charts.Add
    ActiveChart.ChartType =
    xlColumnClustered
    ActiveChart.SetSourceData
    Source:=Sheets("Sheet1").Range(
    "A1:A7")
    ActiveChart.Location
    Where:=xlLocationAsObject,
    Name:="Sheet1"
End Sub
```



ADODB ve Dış Verilerle Çalışmak

- SQL

  ConStr = "Provider=SQLOLEDB;Server= SSUNGUR\SQLEXPRESS;Database=SEMIH;User Id=xx;Password=***"

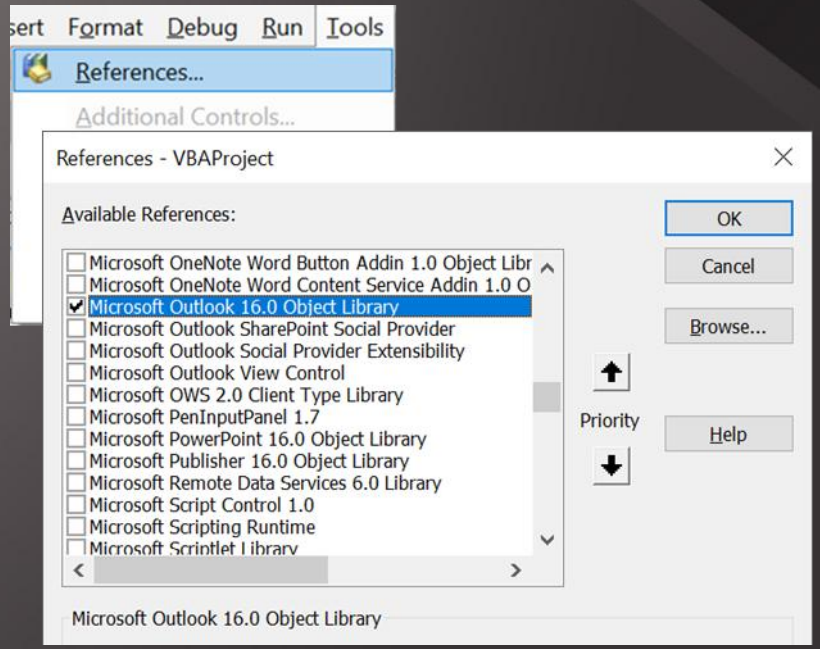
- Access
  - ConStr = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source =[DosyaAdi];User Id=xx;Password=*** "
  - ConStr = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source =[DosyaAdi];User Id=xx;Password=*** "

- Excel

  ConStr = "Provider =Microsoft.ACE.OLEDB.12.0;Data Source=[DosyaAdi];Extended Properties=Excel 12.0 Xml;HDR=YES"

Mail Gönderme Uygulaması

```
Sub Send_Mail_With_Outlook()
    Dim oApp As Object
    Dim oMail As Object
    Set oApp = CreateObject("Outlook.Application")
    Set oMail = oApp.CreateItem(0)
    With oMail
        .To = "info@semihsungur.com"
        .CC = ""
        .Subject = "Excel VBA Eğitimi"
        .Body = "Bu mail eğitim uygulaması ile gönderilmiştir."
        '.Attachments.Add ActiveWorkbook.FullName
        '.Attachments.Add ("C:\Windows\temp\test.xlsx")
        .send ‘or .Display
    End With
   Set oMail = Nothing
    Set oApp = Nothing
End Sub
```



Mail Gönderme Uygulaması-2

```
Sub Send_Mail_With_Outlook()
        Dim oApp As Outlook.Application
        Dim oMail As Outlook.MailItem
        Dim oIns As Outlook.Inspector
        Dim wDoc As Word.Document

        Set oApp = New Outlook.Application
        Set oMail = oApp.CreateItem(olMailItem)

        oMail.To = "info@semihsungur.com"
        oMail.CC = ""
        oMail.Subject = "Excel VBA Eğitimi"
        oMail.oMailbodyformat=olFormatHTML
        xBody= "Bu mail eğitim uygulaması ile gönderilmiştir."
        oMail.Attachments.Add ActiveWorkbook.FullName  veya    '.Attachments.Add ("C:\Windows\temp\test.xlsx")

        Set oIns = oMail.GetInspector
        Set wDoc = oIns.WordEditor
        wDoc.Range.InsertBefore xBody
        oMail.send ‘or .Display

        Set oMail = Nothing : Set oApp = Nothing : Set oIns = Nothing  :    Set wDoc = Nothing
End Sub
```

Kelime Sayma Örneği

```
Sub KelimeSay()
Dim k As Range, i As Integer
Kelime = InputBox("kelimeyi yazın.", "kelime gir")
i = 0
For Each Hucre In Range("A1:H10")
    If Kelime = Hucre Then Sayi = Sayi + 1
Next Hucre
MsgBox "aranan kelime sayısı " & Sayi & " adettir."
End Sub
```

Notlar için Değer Yazma Örneği

```
Function NotGetir(Puan As Integer)
Select Case Puan
    Case 0 To 40
        NotGetir = "FF"
    Case 41 To 50
        NotGetir = "FD"
    Case 51 To 55
        NotGetir = "DD"
    Case 56 To 60
        NotGetir = "DC"
    Case 61 To 70
        NotGetir = "CC"
    Case 71 To 75
        NotGetir = "CB"
    Case 81 To 85
        NotGetir = "BB"
    Case 86 To 90
        NotGetir = "BA"
    Case 91 To 100
        NotGetir = "AA"
   Case Else
        NotGetir = ""
End Select
End Function
```

90–100 : "AA"

85 – 90 : "BA"

75 – 85 : "BB"

70 – 75 : "CB"

60 – 70 : "CC"

55 – 60 : "DC"

50 – 55 : "DD"

40 – 50 : "FD"

0 – 40 : "FF"

Boş Dolu Hücreleri Sayma Örneği

```
Sub Hucre_Say()
For i = 1 To 10
    If Cells(i, 1) <> "" Then 'Eğer hücreler dolu ise
        dolu = dolu + 1 'Dolu değişkeni bir arttırılıyor
    End If 'if kapatılıyor
    If Cells(i, 1) = "" Then 'Eğer hücreler boş ise
        bos = bos + 1 ' Boş değişkenin bir arttırılıyor
    End If 'if kapatılıyor
Next i
Cells(1, 5) = dolu ' Dolu değeri E1 hücresine
Cells(2, 5) = bos ' Boş değeri E2 hücresine yazdırılıyor.
End Sub
```

TEŞEKKÜRLER



semih.sungur@sakademi.com