

Microsoft EXCEL

"İLERİ DÜZEY"

Ders Notları

www.farukcubukcu.com

İÇİNDEKİLER

M

I. MAKROLAR4

A. Makro Nedir	4
B. Makro Türleri.....	4
II. MAKRO GELİŞTİRME.....	5
A. Makro Geliştirmede Kullanılacak Adımlar	5
B. Makro İletişim Kutusu.....	6
C. Makro Senaryoları.....	7
D. MAKROYU ÇALIŞTIRMA YÖNTEMLERİ.....	7
1. Makroyu Microsoft Excel Tool Menüünden çalıştırmak.....	7
2. Makroyu bir düğme,çizim nesnesi yada grafik yardımı ile çalıştırmak.	7
3. Bir nesneye bir makro atama adımları.....	7
4. Makroyu Klavye kısayolundan çalıştırmak	8
5. Makroyu Araç çubuğu butonundan çalıştırmak.	8
6. Makroyu bir özel olaya bağlı olarak çalıştırmak.	8
E. MAKRO DÜZENLENME.....	9
1. Create Macro yöntemi ile Makro oluşturmak.	12
III. EXCEL 97 VBA.....	14
A. Visual Basic Editörü	14
1. Özellikler Penceresi	14
2. Proje Gezgini.....	15
3. Kod Penceresi.....	15
B. Bir Procedure'ın Yapısı	16
1. Genel Yordamlar	16
2. Olay Yordamları.....	17
3. Fonksiyonlar.....	17
C. Nesneler.....	19
1. Özellikler.....	19
2. Nesnelerinin Önemli Özellikleri	20
3. Metotlar.....	20
4. Nesnelerin Önemli Metotları.....	20
D. Gösterim.....	21
1. A1 gösterimi ile hücre ve aralık.	21
2. İndeks Numarası Kullanarak Hücre Gösterimi	21
3. Satır ve Sütunların Gösterimi	22
4. Kısayol Gösterimin Kullanılması	23
5. Adlandırılmış Hücrelerin Gösterimi.....	23
6. Göreceli Adresleme.....	24
7. Range Nesnesi Kullanarak Hücre Gösterimi.....	25
8. Çalışma Tablosundaki Bütün Hücrelerin Gösterimi	25
E. Temel Giriş Çıkış Deyimleri.....	25
1. MsgBox Deyimi	Hata! Yer işareti tanımlanmamış.
2. MsgBox Fonksiyonu	Hata! Yer işareti tanımlanmamış.
3. Giriş Kutularının Kullanımı	Hata! Yer işareti tanımlanmamış.
F. PROGRAM DENETİMİ	HATA! YER İŞARETİ TANIMLANMAMIŞ.
1. Döngüler.....	Hata! Yer işareti tanımlanmamış.
2. Do...Loop Deyimi	Hata! Yer işareti tanımlanmamış.
Döngünün Başında ve Sonunda Denetim.....	Hata! Yer işareti tanımlanmamış.
3. Bir Koşul Doğru Oluncaya Kadar Döngü	Hata! Yer işareti tanımlanmamış.
4. For...Next Deyimi	Hata! Yer işareti tanımlanmamış.

5. Karar Yapıları.....	Hata! Yer işareti tanımlanmamış.
6. If...Then...Else Deyimi	Hata! Yer işareti tanımlanmamış.
7. Select Case Deyimi	Hata! Yer işareti tanımlanmamış.

I. MAKROLAR

A. Makro Nedir

Makro: Sık yapılan işleri otomatikleştirir.

Eğer Microsoft Excel’de belirli bir işi tekrarlı olarak (yinelemeli) yapıyorsanız, bu işi makro ile otomatik hale getirebilirsiniz. Örneğin her ay sonu yaptığınız işlemler. Makro, belli bir işi yapmak üzere her istediğinizde çalıştırabileceğiniz bir Visual Basic modülü içersine depolanmış komutlar ve fonksiyonlar dizisidir.

B. Makro Türleri

Makroları, oluşturma yöntemine göre iki grupta toplayabiliriz.

- Kayıt (Record) Yöntemi ile makro oluşturma.
- Visual Basic Editörü ile makro oluşturma.

Makro kaydetmek tıpkı bir kasete müzik kaydetmeye benzer. Bir kez kaydedilen müzik nasıl tekrar tekrar çalınabiliyorsa bir kez kaydedilen makro da tekrar tekrar çalıştırılabilir. Makro kaydetmeye yada yazmaya başlamadan önce makronun yapacağı işe göre kullanılacak komutlar ve adımlar planlanmalıdır. Çünkü özellikle kayıt makrosu oluşturulurken bir hata yapılırsa hatayı düzeltme adımları da kaydedilecektir. Her yeni bir makro kaydedişinizde; makro, çalışma kitabına eklenen bir modül sayfasına saklanır.

Visual Basic editörü ile de makro düzenlenebilir. Visual Basic Editörü ile makroları, bir modülden diğerine ya da çalışma kitapları arasında makro kopyalayabilir ve makro ya da modülün adını değiştirebilirsiniz .

II. MAKRO GELİŞTİRME

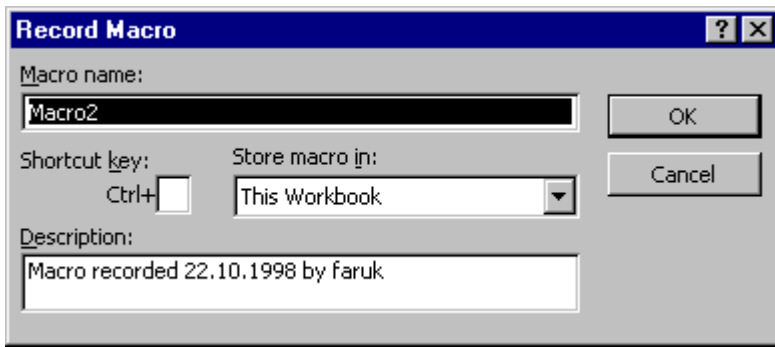
Bu bölümde bir kayıt makrosunun nasıl yaratılacağı, çalıştırılacağı ve yeniden düzenlemelerin yapılması işlemleri anlatılmaktadır. Kayıt makroları daha önce de belirtildiği gibi rutin işleri otomatik hale getiren işlem tanımlamalarıdır. Veri üzerinde yapılacak işlemler bir kez yapılırken makro olarak kayıt edilir (record). Ardından bu makro yeni veriler üzerinde kullanılır.

A. Makro Geliştirmede Kullanılacak Adımlar

Bu yöntemi kullanmadan önce yapacağımız işin adımlarını çok iyi belirlemeliyiz hatta gerekiyorsa bir kenara adım adım not alın.

Makro kaydetme:

1. **Tools** menüsünden **Macro** komutunu ve Record New Macro komutunu seçin .
2. **Macro Name** kutusuna makro için bir ad yazın . Örneğin “Kolon Topla”.



Not: Makro adının ilk karakteri harf olmalıdır. Diğer karakterler harf, rakam yada alt çizgi karakteri olabilir. Makro adında boşluğa izin verilmez ;boşluk yerine alt çizgi işareti kullanılabilir.

3. **Store macro in** kutusundan makroyu kaydetmek istediğiniz yeri seçin. Eğer makronuzun Excel’in her yerinde geçerli olmasını istiyorsanız ; makronuzu XLStart klasöründeki, **Personal Macro Workbook** a kaydedin.

Saklama yerleri:

This Workbook: Bulunulan çalışma kitabına.

New Workbook: Yeni bir çalışma kitabına

Personal Macro Workbook: Sadece makroların bulunduğu özel bir çalışma kitabına.

Makronuzun bir açıklama içermesi için **Description** kutusuna açıklamanızı yazabilirsiniz. (Bu açıklama makronun yaptığı iş olabilir).

4. **OK** düğmesine tıklayarak makro kayır işlemine başlanır.

5. Makroya kaydetmek istediğiniz işlemleri yapın.

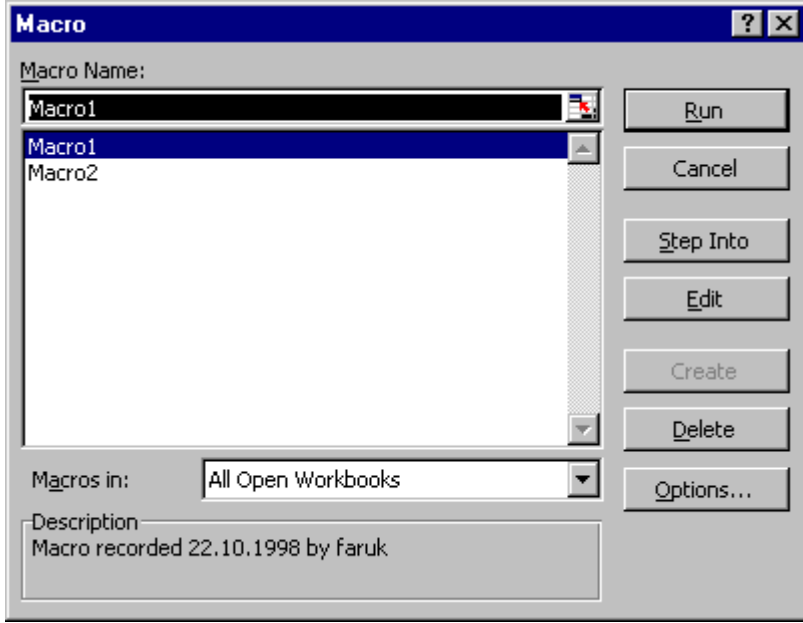
Makro çalışırken bazı hücreleri seçecek yada bu hücrelerden giriş çıkış yapacaksa bu hücrelerin neye bağlı olarak seçileceği ile ilişkili bir ön tanımlama yapmak gerekir. Yani makronun erişeceği hücreler makro çalıştırıldığı anda aktif olan hücreye bağlı olarak değişecekse (göreceli adres kullanacaksa) Stop Recording Araç çubuğundaki Relative References butonuna tıklayıp onu basılı duruma getirin, ancak bir müddet sonra hücre adresleri önceki aktif hücrelerden atkilenmeyecekse bu durumda Relative Reference butonuna bir kez daha tıklayarak onu basılmamış duruma getirin. Bir makro içersinde hem mutlak hemde göreceli adresleme tekniğini birlikte kullanabilirsiniz. Her birinin etkisi kendisinden sonra yapılan eylemleri etkiler.

6. Kaydı bitirmek için **Stop Recording** düğmesine tıklayın..

Not: Makro kayıt ederken genellikle temiz (hatasız) bir kullanım gerekir. Eğer kayıt sırasında hata yapılırsa bu hatalar makronun çalıştırılmasında da karşımıza gelir. Bunun dışında kayıt ederken durum çubuğunda “Recording” açıklaması yer alır.

B. Makro İletişim Kutusu

Kaydedilen makroları çalıştırmak, yendien düzenleme ve silmek için makro iletişim kutusu kullanılır.



İletişim kutusu seçenekleri:

Macro Name: Makronun adını içerir.

Macros In : Listelenen makroların yer aldığı projeleri gösterir.

Run: Seçilen makroyu çalıştırır.

Step Into: Makronun ilk satırını göstererek işletim için göstergesi bu satırın başına getirir.

Edit: Makro kodunu açarak görülmesini ve düzenlenmesini sağlar.

Create: Yeni bir makro yaratmak için kod penceresi açar.

Delete: Seçilen makroyu projeden çıkarır.

C. Makro Senaryoları

Makroların kayıt edilerek yaratılması ve daha sonra istenildiği kadar kullanılması; gereksinimlere göre farklı işlerde kullanılmasını sağlar.

Örneğin: Bir ay, ürün toplamının (ya da başka bir işlemin) diğer bir ürün ya da ay için de uygulanması. Bunun dışında yine her ay düzenlenen bütçe özetleme işleminin her ay otomatik olarak makro ile yapılması.

Bunun dışında biçimleme işlemleri yapan makrolar, hesaplama işlemleri yapan makrolar, grafik çizen makrolar, veriler üzerinde toplu değişiklik yapan makrolar gibi makrolar yaratılır.

Örnek: 142 numaralı ascii (Ä) karakterlerini **i** harfi ile değiştiren makronun kaydedilmesi.

1. **Tools** menüsünden **Macro** oradan da **Record New Macro**'a tıklayınız.

Record macro iletişim kutusu görüntülenecektir. Macro Name kutusuna makro adı olarak TurkceABC giriniz. Makroyu klavye kısayoluyla çalıştırmak istiyorsanız, Shortcut key kutusuna bir harf giriniz.

2. Ok düğmesine tıklayınız.

Edit menüsünden Replace ya da Klavyeden Ctrl-H kısayolunu kullanarak değiştirme komutunu çalıştırınız.

Find What kutusuna 142 (ALT+142) nolu ascii karakterini, Replace with kutusuna da **i** harfini giriniz .

Replace All düğmesine tıklayınız. Sayfadaki tüm ASCII 142 karakterlerinin yerine i harfleri yazılacaktır.

3. **Stop Recording** düğmesini tıklayarak, makro kaydını bitirin.

D. Makroyu Çalıştırma Yöntemleri

Excelde hazırladığınız bir makroyu aşağıdaki ortamlardan çalıştırabilirsiniz.

1. Microsoft Excel Tool Menüsünden
2. Düğme aracılığıyla
3. Çizilen bir nesne ya da grafik aracılığıyla
3. Klavye kısayoldan
4. Araç çubuğundan
5. Özel olaylara bağlı olarak

İpucu: Çalışmakta olan bir makroyu durdurmak için ESC ye basınız.

1. Makroyu Microsoft Excel Tool Menüsünden çalıştırmak

1. **Tools** menüsünden, Macro menüsü oradan da istenine makro seçilir.
2. Macro Name kutusundan, çalıştırmak istediğiniz makronun adını yazın.Yada altındaki listeden çalıştırmak istediğiniz makroyu seçin.
3. **Run** düğmesine tıklayın.

2. Makroyu bir düğme,çizim nesnesi yada grafik yardımı ile çalıştırmak.

Çalışma kitabınızdaki bir butona ,çizim nesnesine yada grafiğe bir makro atayabilirsiniz .Buton, çizim nesnesi yada grafiğe tıkladığınızda bunlara atanmış olan makrolar otomatik olarak çalışırlar

3. Bir nesneye bir makro atama adımları

1. Makronun atanacağı nesneye tek tıklayarak nesneyi seçin.

2. Seçili nesnenin üzerinde iken sağ tıklayıp açılan kısayol menüsünden **Assign Macro** ya tıklayın. Varolan bir makroyu atamak için **Macro Name** kutusuna nesneye atanacak makronun adını yazın ve OK düğmesine tıklayın.

Eğer yeni kayıt makrosu kaydedip nesneye onu atamak istiyorsanız Record düğmesine tıklayın

4. Makroyu Klavye kısayolundan çalıştırmak

Bu işlem iki yoldan yapılır:

1. Makro (ilk defa) yaratılırken
2. Makro yaratıldıktan sonra

1. **Tools** menüsünden Macro komutunu seçin.
2. Macro Name kutusuna yeni makronun adını yazın.
3. Shortcut key kutusuna makro çalıştırmak için kullanacağınız harfi yazın.

Mevcut bir makroya bir kısayol tuşu atamak istediğiniz zaman:

1. **Tools** menüsünden Macro komutunu seçin.
2. Option düğmesine tıklayın.
3. Shortcut key kutusuna bir harf giriniz.

Kısayol olarak CTRL + harf (küçük harfler için) veya CTRL+ SHIFT + harf (büyük harf için) tuşlarına basabilirsiniz. Kısayol tuşu olarak rakam yada özel karakter kullanamazsınız. Kısayol olarak tanımladığınız anahtara eğer önceden Excel tarafınan bir makro atanmışsa önceki tanımlama geçersiz olacaktır. Tabiki sadece tanımladığınız makronun geçerli olduğu çalışma kitaplarında.

5. Makroyu Araç çubuğu butonundan çalıştırmak.

Bir makroyu var olan yada kendi oluşturacağınız bir araç çubuğundaki butondan çalıştırabilirsiniz .

Araç Çubuğu'na Makro Atama Adımları

1. View menüsünden Toolbars komutunu seçilir.
2. Customize komutunu seçin.

Makro atamak istediğiniz araç çubuğu görüntüde değilse Toolbar sekmesine tıklayıp görüntülenmesini istediğiniz araç çubuğunun başındaki onay kutusunu fare ile tıklayarak doldurun.

3. **Commands** sekmesine tıklayıp, **Categories** listesinden **Macros** bölümüne tıklayın ve görüntüye gelen **Custom Button** nesnesini bir araç çubuğunun üzerine sürükleyin.
4. Araç çubuğunun üzerine sürüklediğiniz özel düğme nesnesine sağ tıklayarak açılan menüden **Assign Macro** ya tıklayın.
5. Macro Name kutusuna butona atamak istediğiniz makronun adını yazın.

6. Makroyu bir özel olaya bağlı olarak çalıştırmak.

Excelde bir olay meydana geldiğinde bir makronun kendiliğinden devreye girmesi için makroya özel adlar verilebilir.

Otomatik çalışan makro adları ;

Auto_Open: Bu makronun etkin olduğu çalışma kitabı açılırken çalışır.

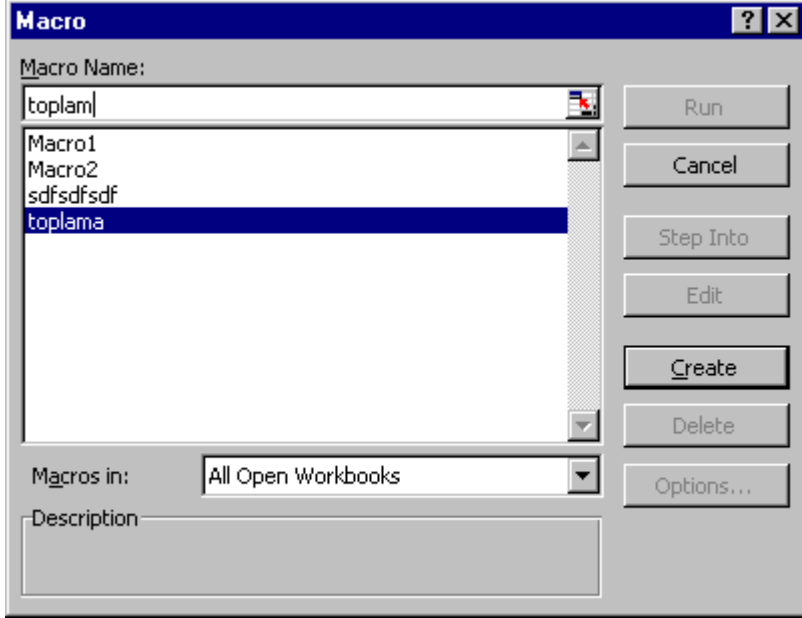
Auto_Close: Bu makronun etkin olduğu çalışma kitabı kapatılırken çalışır.

E. MAKRO DÜZENLENME

Makro düzenlemek, Record macro yöntemi ile kaydedilmiş makrolar üzerinde değişiklik yapmak ya da Create Macro yöntemi ile yeni makro geliştirmeyi kapsar

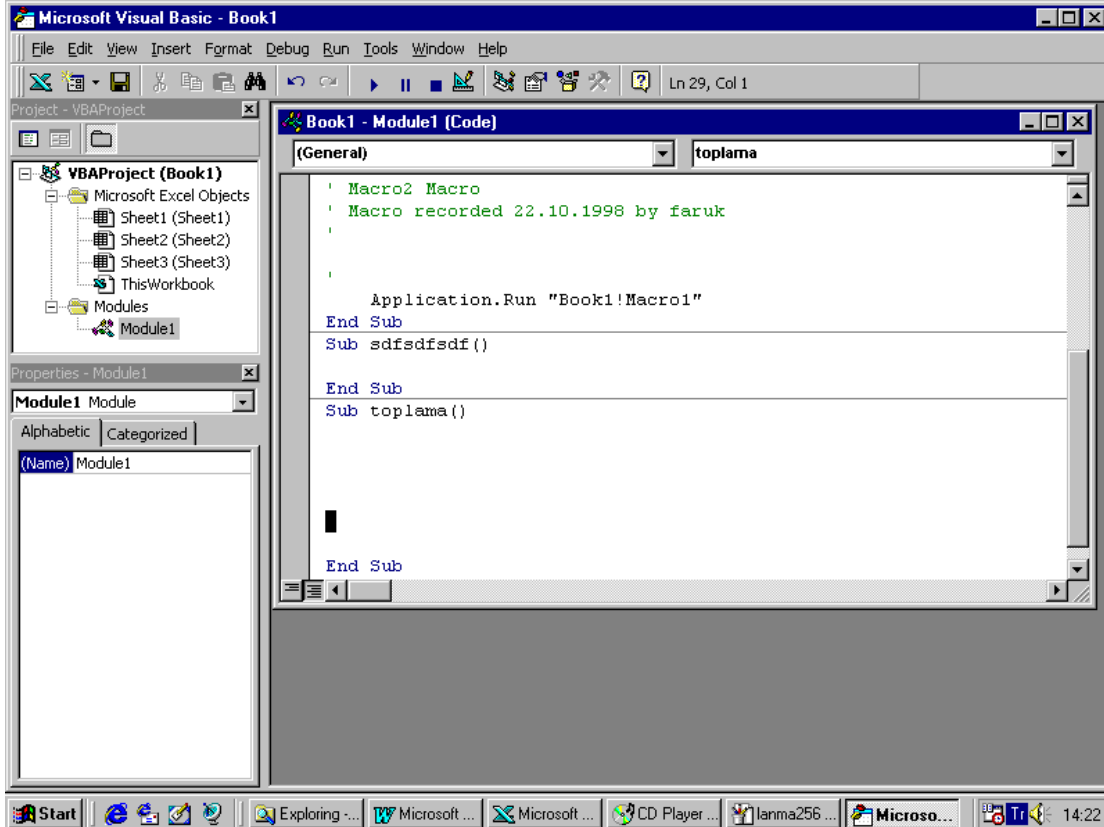
Record yöntemi ile geliştirilmiş makroları düzenlemek için:

1. **Tools** menüsünden Macro, oradan da **Macros** seçeneğine tıklayın .



2. İstedığınız makroyu seçin ve **Edit** düğmesine basın.

Bu gelen ekran VBA dilinde kod yazmak yada önceden yazılmış makroları tekrar düzenlemek amacıyla kullanılan bir kod geliştirme ekranıdır.



Makro kaydederken yapmış olduğunuz işlemlerin VBA dilindeki karşılıkları **Sub** Macro adı ve **End Sub** deyimleri arasında göreceksiniz.

Bu bölümde şimdilik yapabileceğimiz en basit işlem önceden kaydettiğimiz. bir makronun içeriğini değiştirmek ve düzenlemektir.

Not :Excel ve Makro modul sayfası arasında geçiş için Taskbar yada Klavyeden Alt-Tab kısayolunu kullanabilirsiniz .

SheetEkle adıyla kaydettiğimiz makronun kodları aşağıdaki şekilde görünecektir.

```
Sub SheetEkle()  
,  
' SheetEkle Macro  
' Macro recorded 2/5/98 by sau9  
,  
' Keyboard Shortcut: Ctrl+q  
,  
    Sheets.Add  
End Sub
```

Başlangıç olarak bu kodları satır satır açıklayıp ne işe yaradıklarını anlamaya çalışalım.

Sub SheetEkle()

Bu satır makromuzun adının SheetEkle olduğunu gösteriyor. Sub deyimini bunun bir altprogram olduğunu gösteriyor.

Tek tırnak (') işareti ile başlayan satırlar programın çalışmasını etkilemez ancak program içersine yazılacak açıklama satırları tırnak işareti başlaması gerekir.

Sheet.Add

Programın tek komut satırı olan bu satır Sheet tipinde bir nesne ekleneceğini gösteriyor.

End sub

ve programın bittiği satır.

Sheet.Add satırını Sheet.Delete olarak değiştirip makromuzu bir kez de bu şekliye çalıştıralım . Tabi makroyu çalıştırmadan önce Excel penceresine geçmek gerekir.

Diğer bir örnek; Ascii karakterlerini Türkçe harfle değiştiren makronun Edit Macro ortamında düzenlenmesi.

1. Tools menüsünden Macro oradan da macros a tıklayınız
2. Macro name kutusundan TurkceAbc adlı makroyu seçip **Edit** düğmesine tıklayınız.
3. TurkceABC makrosu aşağıdaki şekilde görüntülenecektir.

```
Sub TURKCEABC()  
,  
' TURKCEABC Macro  
' Macro recorded 2/5/98 by sau9  
,  
' Keyboard Shortcut: Ctrl+Shift+Q  
,
```

```
Cells.Replace What:="İ", Replacement:="i", LookAt:=xlPart, SearchOrder _  
:=xlByRows, MatchCase:=False
```

End Sub

Yukarıdaki örnek sadece 142 nolu ascii karakterini değiştirecek şekilde. Ancak çok sayıysa farklı karakterin değiştirilmesi isteniyorsa her biri için teker teker kayıt makroları oluşturmak zaman alır bunun için elimizdeki bu kayıt makrosu üzerinde bazı değişiklikler yaparak bu sorunu çözelim.

What ="İ" satırı aranan karakteri, Replacement="i" satırı da yerine konacak karakteri gösterir.

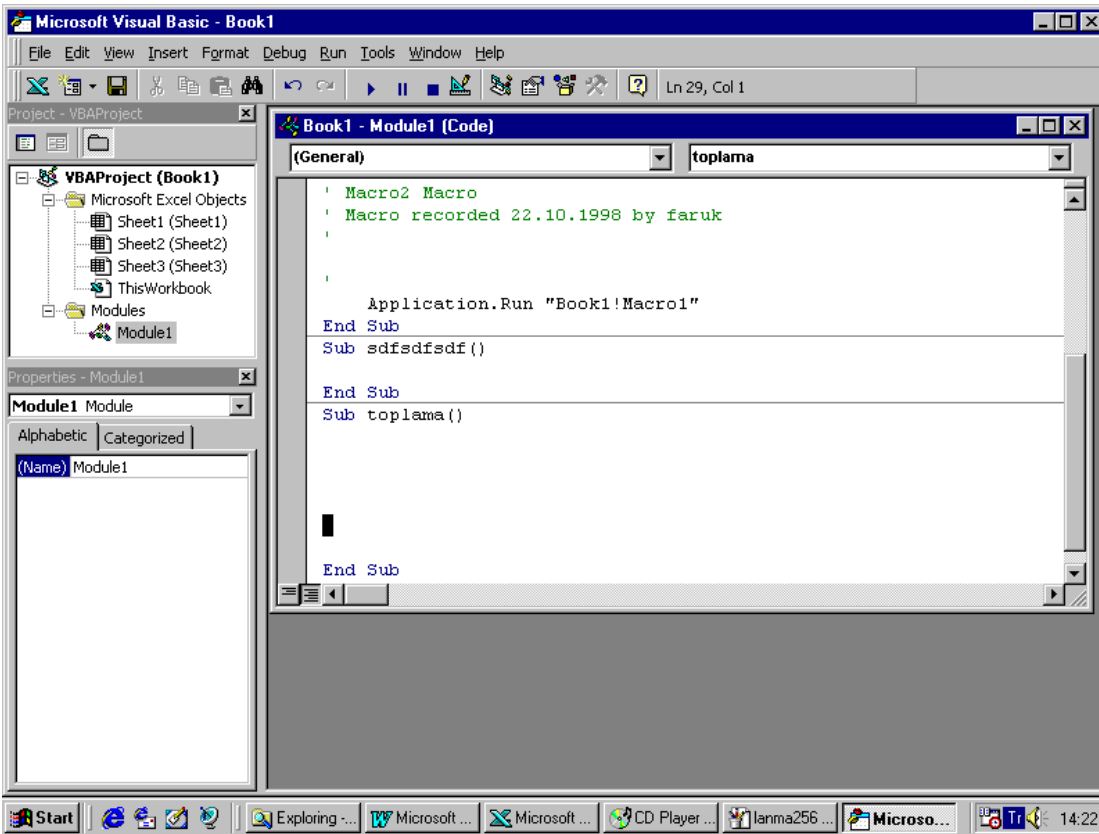
Makromuzu aşağıdaki şekle getirelim.

```
Sub TURKCEABC()  
,  
' TURKCEABC Macro  
' Macro recorded 2/5/98 by sau9  
,  
' Keyboard Shortcut: Ctrl+Shift+Q  
,  
  
Cells.Replace What:="y", Replacement:="i", LookAt:=xlPart, SearchOrder _  
:=xlByRows, MatchCase:=False  
Cells.Replace What:="f", Replacement:="u", LookAt:=xlPart, SearchOrder _  
:=xlByRows, MatchCase:=False  
  
End Sub
```

Excel penceresine geçip makroyu çalıştırın.

1. Create Macro yöntemi ile Makro oluşturmak.

Tool menüsünden Macro ya oradan Macros'a tıklayın Macro Name kutusuna makroya vermek istediğiniz (yeni) adı yazıp **Create** butonuna tıklayın.



Ardından Visual Basic editörü içinde:

Sub makroadı()

End Sub

Alanları görünecektir.

Sub makroadı()

-bu kısımda kodlarınızı yazın-

End Sub

Not : Create yöntemi ile oluşturulan makrolar da aynen Record yöntemi ile kaydedilen makrolar gibi çalıştırılırlar.

III. EXCEL VBA

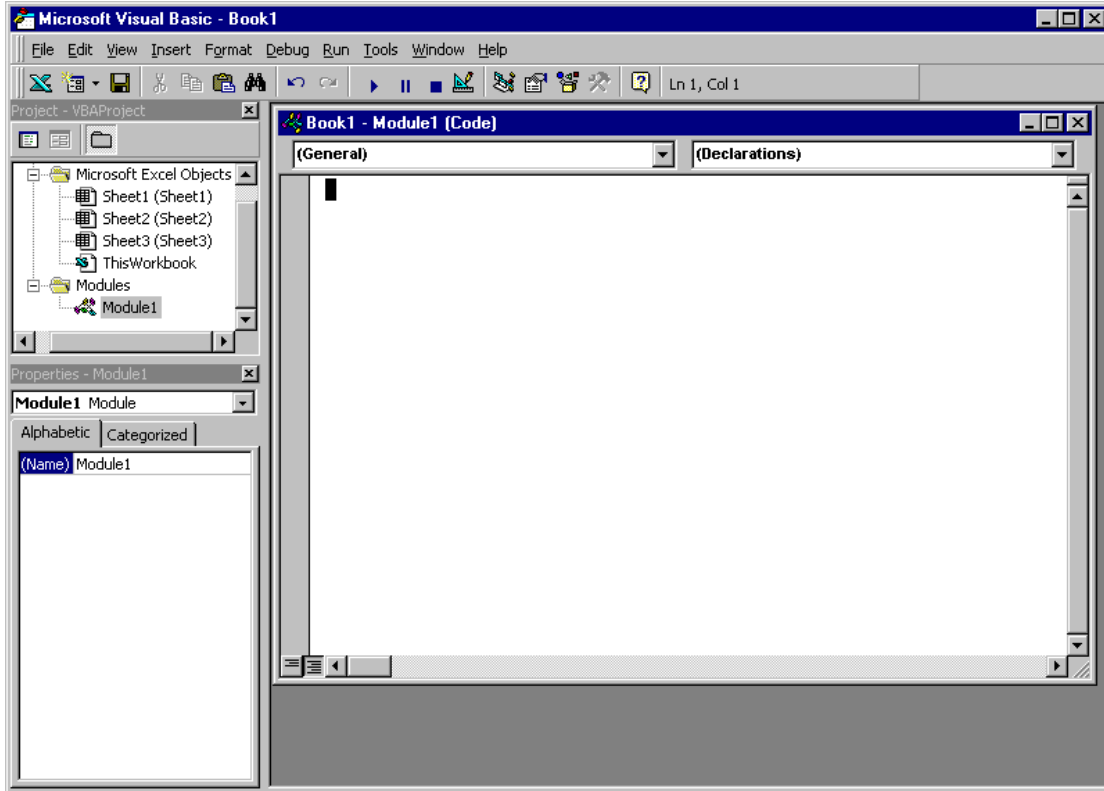
Visual Basic for Application (VBA), Microsoft Visual Basic programlama dilinin nesne temelli bir uyarlamasıdır. Nesneler Microsoft uygulamaları tarafından sağlanır.

Visual Basic for Application, Excel içinde yer alır. Excel gibi diğer uygulamalar da VBA'ı destekler. Visual Basic bilen kullanıcılar VBA'ye alışmaları çok kolay olacaktır. Excel bilen kullanıcılar ise hem hem VBA'ı öğrenecek hem de Visual Basic'i öğrenecektir.

VBA, güçlü ve kolay kullanılan bir makro dili olarak geliştirilmiştir. Uzman ve yeni başlayan kullanıcılar (kendi işleri ile ilgili) gereksinimlerini VBA ile kolayca çözebilecektir. VBA sayesinde diğer uygulamalarla da iletişim sağlanır. Birbirine bağlanabilen bu uygulamalar; Microsoft Excel, Microsoft Power Point, Microsoft Project ve Visual Basic ile yazılmış uygulamalar.

A. Visual Basic Editörü

Excel içinde bir makro ya da VBA kodu yazmak için Visual Basic araç çubuğu üzerinde yer alan “Visual Basic Editor” kullanılır.



Eğer daha önce Excel ya da Word ortamında makro yazdıysanız, Visual Basic Düzenleyicisi ilk başta daha önce görmediğiniz pencereler ve düğmelerle karışık gelecektir.

Not: Visual Basic Düzenleyici içindeki özel bir pencere hakkında bilgi almak için, o pencereyi açtıktan sonra F1 tuşuna basın.

1. Özellikler Penceresi

Bir özellik, bir nesnenin rengi ya da başlığı gibi bir karakteristiğidir. Bir nesnenin boyu, rengi, yazı tipi bilgileri bu pencere aracılığıyla düzenlenir.

2. Proje Gezgini

Bir çalışma kitabı, belge, şablon ya da sunumla ilişkilendirilmiş kodların hepsi çalışma kitabı, belge, şablon ya da sunumla beraber otomatik olarak kaydedilen bir proje içinde saklanır. Visual basic düzenleyicinin içindeki, proje gezgini ile açık bulunan her çalışma kitabı, belge, şablon ya da sunumun içindeki projeleri görebilir, değiştirebilir ve aralarında geçiş yapabilirsiniz.

Proje gezgini penceresini isterseniz değişik yerlere yerleştirebilir ya da boyutlarını değiştirebilirsiniz.

Proje gezgininde açık bulunan her çalışma kitabı, belge, şablon ya da sunum için bir proje vardır. Her proje içinde ise, olayları temsil eden nesneleri bulacaksınız. Bu nesneler; Belgeler, Çalışma kitapları ve Çalışma sayfalarıdır.

Formlar: (Kullanıcı formları olarak da adlandırılır) Kullanıcıların veri girişi ya da bilgi alış verişi için kullanılan iletişim kutularıdır.

Standart Modüller: Özel bir nesne ya da bir formla ilişkisi olmayan kodları taşırlar.

Sınıf Modüller: Özel bir nesne tipinin kodunu taşırlar.

Referanslar: Diğer projelere olan referansları taşırlar.

Bir modül ya da form içindeki kodu görebilmek için, nesneye tıkladıktan sonra, Proje gezgini penceresinin üstündeki, Kod Görüntüle düğmesine basın.



Kod Görüntüle düğmesi

Bir nesne ya da formun kendisini görebilmek için ise, nesneye tıkladıktan sonra, Proje gezgini penceresinin üstündeki, Nesne Görüntüle düğmesine basın.



Nesne Görüntüle düğmesi

3. Kod Penceresi

Bir proje içindeki kodları görebilmek için, kodu taşıyan elemana tıkladıktan sonra, Proje Gezgini penceresinin üst tarafında bulunan Kodu Görüntüle düğmesine basmanız gerekir.

İpucu: Kod penceresinde aynı anda birden fazla yordam görebilmek için, Araçlar, Seçenekler menüsündeki, Düzenleyici sekmesinden, Tam Modül Görünümünü Varsay kontrol kutusunu seçin. Her seferinde bir yordam görebilmek için ise bu kontrol kutusunu temizleyin.

Kod pencerenin üst tarafında bulunan Nesne (Object) ve Yordam (Procedure) listelerini kullanarak kodlar arasında geçiş yapabilirsiniz. Nesne kutusundan, (Genel) üzerine gelin ve Yordam kutusundan, özel bir olaya atanmamış bir Yordam adına tıklayın. Nesne kutusunda, bir nesneye tıklayın, ardından Yordam kutusundan, herhangi bir olay üzerine basarak, o olaya atanmış bulunan kodları görün.

B. Bir Procedure'ın Yapısı

Visual Basic düzenleyici ortamına yaratılacak olan program kokdu makro ya da procedure (yordam) olarak adlandırılır.

Visual Basic'te değişik türde yordamlar (procedure) kullanılır:

- Sub
- Function
- Property

Bir Sub yordamı herhangi bir değer döndürmez. Çağrılarak ya da bir olaya tepki olarak çalışır. Function yordamlar ise bir değer döndürürler. Örneğin bir faiz hesabının sonucunu döndürürler. Property yordamlar ise bir değer döndürürler, atama yaparlar ve nesnelerin referanslarını düzenlerler.

Bir Sub yordamının yapısı:

[Private|Public][Static] Sub yordam adı (argümanlar)

Deyimler

End Sub

Yordam çalıştırıldığında Sub ve End deyimleri arasında kalan satırlar işletilir. Sub yordamlar standart modüller, class modüller ve form modülleri içinde yer alırlar. Sub yordamlar varsayım olarak bütün modüllerde Public tanımlanır. Bunun anlamı bu yordamların uygulamanın her yerinden (diğer modüllerden) çağırılabilmesidir.

Argümanlar ise tanımlanan bir değişken gibidir. Çağırılan yordamdan geçen değerler olarak tanımlanır.

Visual Basic'te yordamlar (procedure) ikiye ayrılırlar:

- Genel yordamlar
- Olay yordamları

1. Genel Yordamlar

Bir genel procedure belli bir işlemi üstlenen program parçasıdır. Örneğin bir dosyanın açılması ya da raporun bastırılması için kullanılabilir. Genel yordamlar diğer yordamlardan çağırılırlar. Böylece belli bir işlevi olan bu birim programın her yerinden istenildiği kadar çağrılarak kullanılır.

Buna karşın olay yordamları ise kullanıcının yarattığı bir olay nedeniyle çalışır. Olay yordamları kullanıcıların hareketlerine tepki olarak çağırılırlar. Örneğin bir düğmeye tıklayarak bir işlemin başlatılması.

İyi bir programlama stratejisi içinde bir takım ortak görevler genel yordamlar olarak planlanmalıdır. Böylece kod tekrarına gerek kalmaz.

Yapısı:

Sub yordam adı(arguments)

-deyimler-

End Sub

2. Olay Yordamları

Olay yordamları bir olayın (event) oluşmasına tepki olarak çalışırlar. Visual Basic nesneleri (formlar, metin kutuları, düğmeler, vb) belli olayları tanırlar. Böylece olay oluştuğunda; o olay için hazırlanmış olay yordamı çalışır.

Bir komut düğmesinin olayları:

Click: Tıklama

GotFocus: Odaklanma, üzerine gelme.

KeyDown: Bir tuşa basmak.

MouseDown: Farenin bir tuşuna basmak.

Olay yordamları, bir nesnenin alt tire ile olaylara bağlı olarak geliştirilir. Bu adlar Visual Basic kod editörü tarafından otomatik olarak ya da manuel olarak oluşturulur.

Yapısı:

Nesne_Olay

Örnek:

Command1_Click

Olay yordamları Visual Basic nesnelerinin olaylarına bağlı olarak geliştirilirler: Örneğin **Form1_load** olay yordamı bir formun açılması sırasında çalışacak bir yordamı, **Command1_Click** olay yordamı da bir komut düğmesine tıklanıldığında çalışacak bir kod birimini gösterir.

Yapısı:

Sub *kontrol adı _olay adı(argümanlar)*

-deyimler-

End Sub

Olay yordamlarının adını kod editörü içinde doğrudan yazabileceğiniz gibi Visual Basic tarafından otomatik olarak koda eklenen yordam ve olay adlarını da kolayca kullanabilirsiniz. Bu işlem için kod editörü penceresinde nesne ve olay seçilir.

1. Kod editörü penceresinde Object kutusundan istediğiniz nesneyi seçin.
2. Procedure kutusundan istediğiniz procedure'ı seçin.
3. Sub procedure'ınız hazır.

Not: Burada procedure'ların adlarıyla ilgili olarak önemli bir konuyu açıklamak istiyorum. Bir komut düğmesine bir olay yordamı ekledikten sonra; örneğin Command1_Click, özellikler tablosundan Name özelliği ile kontrolün adı değiştirilirse Visual Basic bunu takip edemez ve eski procedure'ı genel bir procedure olarak modül genel düzeyine atar. Bu durumda procedure yazılmadan önce adlandırma işleminin yapılması gerekir.

3. Fonksiyonlar

Visual Basic procedure'ları sadece Sub olarak düzenlenmez. Bir diğer procedure yapısı da Function olarak bilinen fonksiyonlardır.

Bir procedure olarak düzenlenen fonksiyonlar belli bir işlemi kod olarak içeren ve bir değeri geri döndüren program parçalarıdır. Bir yordam olarak yaratılan fonksiyonların Sqr, Val gibi Visual Basic'te hazır olarak

bulunan fonksiyonlarla ilişkisi yoktur. Fonksiyonlar kullanım bakımından bir işlemi yapan ve istenildiği yerde kullanılabilen bir kod birimidir. Örneğin bir faiz fonksiyonu istenildiği program içinde istenilen yerde çağırılarak kullanılır.

Bir fonksiyon yordamının yapısı:

[Private|Public][Static] Function *yordam adı (argümanlar)* [As *tip*]

-deyimler-

End Function

Fonksiyonlar ayrı bir yordam olarak düzenlenirler. Fonksiyonların kullanımında argümanlar önemlidir. Fonksiyonlar argümanları alırlar, bir dizi işlem yaparlar ve sonucu döndürürler.

Sub yordamlar ile Function yordamlar arasında farklar şunlardır:

- Bir fonksiyon değer döndürür ve genellikle bir eşitliğin sağında kullanılır.
- Bir sub ise belli bir işi yapar. İşlemi tamamlar bir değer ya da parametre ile kullanım zorunluluğu yoktur.

Function UcretHes(deger1, deger2, deger3)

-deyimler-

End Function

Yapısı:

Ucret = UcretHes(gun, baz, katsayı)

Fonksiyon yordamlar bir değeri elde etmemizi sağlarlar. Fonksiyonlar değişken gibi veri tipine sahiptirler. Bu veri tipi dönecek verinin tipini belirler. Fonksiyonlar bir hesaplamanın parçası da olabilirler.

Toplam = Ikramiye + **UcretHes**(gun, baz, katsayı) / 360

Örneğin çapı verilen bir dairenin alanını hesaplayan bir fonksiyon ya da verilen dereceyi fahrenheit'e çeviren bir fonksiyon gibi.

Örnek: Dereceyi Fahrenheit'e çeviren fonksiyon

Fonksiyonun tanımı:

```
Function Fahrenheit(x)
Fahrenheit = x * 9 / 5 + 32
End Function
```

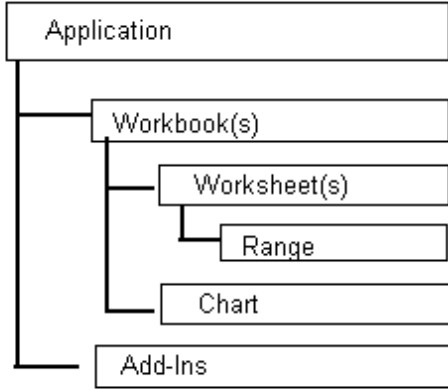
Fonksiyonun kullanımı:

```
Private Sub Command1_Click()
Dim Değer As Integer
Değer = InputBox(" ")
MsgBox Fahrenheit(Değer)
End Sub
```

C. Nesneler

Nesne (object) çalışma tablosu (worksheet), sayfa (sheet), sayfa üzerinde bulunan bir aralık, çizim, grafik, bir düğme nesne olarak adlandırılır.

Şekil: Microsoft Excel Nesneleri



Excel VBA'de işlevsel olan makrolar nesneler sayesinde yazılır. Nesneler (objects), Excel'de kontrol edilecek şeyleri temsil eder. Bunlar Workbook (Çalışma Tablosu), Rows (Satırlar), Cells (Hücreler), Charts (Grafik) vb bir şey olabilir.

Nesnelerle işlem yapmak için iki önemli aracın bilinmesi gerekir. Bunlar: Methods (Metotlar) ve Properties (Özellikler) dir.

Metotlar nesnelerle yapılacak işlemleri tanımlar. Örneğin bir alanın silinmesi uygun nesne (alan) ve uygun metot (silme işlemi) ile yapılır.

Özellikler ise nesnenin özelliklerini gösterir. Örneğin hücrenin renkli ya da italik olarak yazılması gibi.

Nesne	Açıklama
Application	Excel uygulamasının geneli
Workbook	Excel Çalışma Kitabının tamamı
Worksheet	Bir Çalışma Kitabındaki bir Çalışma Tablosu
Range	Bir ya da daha çok hücre
Chart	Bir Çalışma Tablosundaki bir Grafik

Nesne tanımlanmasında nesnenin kendini kapsayan ve kendi içindeki nesne ile birlikte yapılmasından tam nesne tanımlaması ortaya çıkar:

```
Application.Workbooks("butce").Worksheet("ocak").Range("a1")
```

1. Özellikler

Bir nesnenin özelliği onun rengi, uzunluğu gibi niteliklerini düzenlemek için kullanılır:

```
Range("A1").Font.Bold = True
Cells(5,1).Font.Size = 34
```

2. Nesnelerinin Önemli Özellikleri

Her nesnenin çok sayıda özelliği vardır. Aşağıdaki tabloda yaygın kullanılan özellikler yer almaktadır:

Özellik	Açıklama
ActiveCell	Aktif pencerenin aktif hücresi
ActiveSheet	Aktif Çalışma Kitabının aktif Çalışma Tablosu
ActiveWorkbook	Excel'deki aktif Çalışma Tablosu
Bold	Metnin harflerinin kalın görünmesi
Italic	Metnin harflerinin yatkı görünmesi
Column	Bir alandaki ilk kolonu tanımlayan sayı
Row	Bir alandaki ilk satırı tanımlayan sayı
ColumnWidth	Belirtilen alanda kolonların genişliği
Hight	Nesnenin yüksekliği
Width	Nesnenin genişliği
RowHeight	Belirtilen alandaki satırların yüksekliği
Selection	Seçilen alan ya da nesne
Value	Bir hücrenin içindeki değer

3. Metotlar

Metotlar ise nesne-temelli programlamanın işlemlerini (komutlarını) oluştururlar. Her nesne olası işlemlerini metotlarıyla tanımlar.

Metotların kullanımında nesne, ardından nokta ile metot kullanılır.

Nesne.Metot

ActiveCell.Clear

ActiveWorkbook.Close

4. Nesnelerin Önemli Metotları

Her nesnenin çok sayıda metodu vardır. Aşağıdaki tabloda yaygın kullanılan metotlar yer almaktadır:

Metot	Açıklama
Add	Aynı tipte yeni bir nesne yaratır
Clear	Belirtilen alanın silinmesini sağlar
Copy	Alanın panoya (clipboard) kopyalanmasını sağlar.
Delete	Belirtilen nesneyi siler
Open	Çalışma Kitabını açar

Close	Belirtilen nesneyi kapatır
Activate	Belirtilen nesneyi aktifleştirir

Örnek:

```
Range("A1:B5").Select
Range("A1").Activate
Range("A1:B5").Delete
Range("B5").Value = "Ne haber?"
```

D. Gösterim

Excel üzerinde işlem yaparken genellikle hücrelerdeki veriler kullanılır. Hücrelerle (cells) işlem yapmak için hücreler içindeki değerleri göstermek gerekir. Bu işlem için Range ve Cells metotları kullanılır.

```
Range ("A1")
Cells(1,1)
```

Değerleri:

```
Range ("A1").Value
Cells(1,1).Value
```

1. A1 gösterimi ile hücre ve aralık.

Range metodunu kullanarak A1 gösterimi ile hücre y ada aralık seçebilirsiniz. Aşağıdaki altprogram A1:D5 aralığındaki hücrelerin görünümünü koyu (bold) olarak değiştirir.

```
Sub AralıkFormatla()
    Workbooks("Book1").Sheets("Sheet1").Range("A1:D5").Font.Bold = True
End Sub
```

Aşağıdaki tablo Range metodu kullanarak Bazı A1 gösterimi adreslemeler yapılmıştır:

Gösterim	Anlamı
Range("A1")	A1 hücresi
Range("A1:B5")	A1 den B5 e kadar
Range("C5:D9,G9:H16")	Çoklu aralık seçimi
Range("A:A")	A kolonu
Range("1:1")	Birinci satır
Range("A:C")	A dan C kolonuna kadar
Range("1:5")	Birinci satırdan beşinci satıra kadar
Range("1:1,3:3,8:8")	Birinci, üçüncü, sekizinci satır
Range("A:A,C:C,F:F")	A, C, ve F kolonları

2. İndeks Numarası Kullanarak Hücre Gösterimi

Satır ve sütun index numaralarını kullanarak tek bir hücre GÖSTERİMİ için **Cell** özelliğini kullanabilirsiniz. Bu özellik içeriği bir tek hücre olan bir aralık değerini döndürür.

Aşağıdaki örnekte Cells(6,1) , sheet1 üzerindeki A6 hücresinin değerini Value özelliği kullanarak 10 yapmaktadır.

```
Sub DegerGir()
```

Worksheets("Sheet1").Cells(6, 1).Value = 10

End Sub

Cell özelliği tanımlı bir aralık içerisinde döngü kurulacağı zaman oldukça kullanışlı olmaktadır. Çünkü indeks numarası olarak bir değişken kullanılmasına izin vermektedir.

Aşağıdaki örnekte sheet1 in A1 den A20 ye kadar olan hücrelerine 1 den 20 ye kadar sayılar yazdırılmaktadır.

Sub DegerGir()

Dim counter As Integer

For counter = 1 To 20

Worksheets("Sheet1").Cells(counter, 3).Value = counter

Next counter

End Sub

Not: Eğer bir aralık içerisindeki hücreler üzerinde işlem yapacaksanız bu durumdan Range özelliğini kullanınız.

Örnek: Aşağıdaki örnekte değerleri artıran bir makro yazarak sayfa üzerinde düğmeler aracılığıyla çalıştırın

	A	B	C	D	E	F
1						
2	GELİR	TUTAR				
3	satışlar	34				
4	hizmet	345				
5	toplam	379				
6						
7						
8						

3. Satır ve Sütunların Gösterimi

Tüm satır yada tüm kolonla çalışmak için **Rows** ya da **Columns** özelliklerini kullanmak gerekir. Bu özellikler içeriği hücre aralığı olan bir aralık nesnesi dödürürler.

Aşağıdaki örnekte Rows(1) , birinci satırı döndürür. ColorIndex özelliğinin değeri 46 yapıp hücredeki metnin rengi kırmızı yapılmaktadır.

Sub KirmiziYap()

Worksheets("Sheet1").Rows(1).Font.ColorIndex = 46

End Sub

Not: Yukarıdaki örnekte gördüğümüz gibi altprogram adlarında Türkçe harf kullanmıyoruz.

Aşağıdaki tablo Satır ve Sütun özelliklerini kullanarak, satır ve sütun işaret etmeyi göstermektedir.

Seçim	Anlamı
Rows(1)	Birinci satır
Rows	Çalışma kitabındaki bütün satırlar
Columns(1)	Birinci kolon
Columns("A")	birinci kolon
Columns	Çalışma kitabındaki tüm kolonlar

Aynı anda birden fazla kolon ya da satırla çalışıyorsanız, daha kolay erişim için, Union metodunu kullanarak bir nesne değişkeni tanımlayınız. Bu kolon ve satırların özelliklerini değiştirmek istediğinizde de nesne değişkenini kullanınız.

Aşağıdaki örnek çalışma kitabının sheet1 birindeki birinci, üçüncü, beşinci satırların biçimini Bold olarak değiştirmektedir.

```
Sub BirKacSatır()  
    Worksheets("Sheet1").Activate  
    Dim myUnion As Range  
    Set myUnion = Union(Rows(1), Rows(3), Rows(5))  
    myUnion.Font.Bold = True  
End Sub
```

4. Kısayol Gösterimin Kullanılması

Range özelliğinin yanı sıra kısayol gösteriminde köşeli parantez içinde A1 gösterimini ya da aralık adını kullanılabilir. Bu gösterimde Range ifadesini ya da tırnak işaretlerini kullanmak zorunda değilsiniz.

Aşağıdaki örnek A1:B5 aralığındaki hücre içeriklerini silmektedir.

```
Sub ClearRange()  
    Worksheets("Sheet1").[A1:B5].ClearContents  
End Sub
```

Aşağıdaki örnek MyRange olarak adlandırılmış hücelere 30 değerini atanaktadır.

```
Sub SetValue()  
    [MyRange].Value = 30  
End Sub
```

5. Adlandırılmış Hücrelerin Gösterimi

Bu yöntemde bir adres gösterimi Range özelliğinin bir aralık adı ile birlikte kullanımından meydana gelmektedir.

Aşağıdaki örnek MayBook.xls dosyasında MyRange olarak gösterilen aralığı italik yapmaktadır.

```
Sub AralıkBicimle()  
    Range("MyBook.xls!MyRange").Font.Italic = True  
End Sub
```

Aşağıdaki örnek Report.xls çalışma kitanında bulunan Sheet1 deki Sales aralığının etrafına çerçeve çizmektedir.

```
Sub FormatSales()  
    Range("[Report.xls]Sheet1!Sales").BorderAround weight:=xlthin  
End Sub
```

Aralık seçmek için goto metodunu kullanabilirsiniz.

Goto metodu hem çalışma kitabını ve çalışma tablosunu aktif eder ardından da gösterilen aralığı seçer.

Aşağıdaki örnek MyBook.xls çalışma kitabında bulunan MyRange aralığının içeriğini silmektedir.

```

Sub ClearRange()
    Application.Goto Reference:="MyBook.xls!MyRange"
    Selection.ClearContents
End Sub

```

Aşağıdaki örnek aktif çalışma kitabı için aynı altprogramın nasıl yazılabileceğini göstermektedir.

```

Sub ClearRange()
    Application.Goto Reference:="MyRange"
    Selection.ClearContents
End Sub

```

Adlandırılmış aralıkta bulunan hücreler boyunca döngü örneği.

Aşağıdaki örnek adlandırılmış aralıktaki her bir hücre için döner ve değeri 50 den küçük olan bir hücreye rastlarsa bu hücrenin yazı rengini sarı yapar.

```

Sub SariyaBoya()
    Const limit As Integer = 50
    For Each c In Range("MyRange")
        If c.Value < limit Then
            c.Interior.ColorIndex = 27
        End If
    Next c
End Sub

```

6. Göreceli Adresleme

Başka hücre adreslerini referans kabul eden hücrelerle çalışmanın en yaygın yolu Offset özelliğini kullanmaktır.

Aşağıdaki örnekte aktif çalışma tablosundaki aktif hücrenin bir satır aşağı üç satır sağındaki hücre çift alt çizgi ile biçimlendirilmektedir.

```

Sub Underline()
    ActiveCell.Offset(1, 3).Font.Underline = xlDouble
End Sub

```

Not: Makro kaydederken göreceli adresleme yapmak için makro kaydı sırasında Stop macro butonunun yanındaki Relative References butonuna tıklayabilirsiniz.

Hücre aralığı boyunca dönmek için döngü içerisinde Cell özelliği ile birlikte değiken kullanabilirsiniz.

Aşağıdaki örnekte 3. sütunun 1 den 20. satırına kadar 5 ten 100 e kadar 5 er 5 er artan sayılar yazdırılmaktadır.

Counter değişkeni burada hem satır index numarası hemde hücreye atanacak değerin hesaplanmasında kullanılmıştır.

```

Sub Dongu()
    Dim counter As Integer
    For counter = 1 To 20
        Worksheets("Sheet1").Cells(counter, 3).Value = counter * 5
    Next counter
End Sub

```


7. Range Nesnesi Kullanarak Hücre Gösterimi

Range nesnesi için bir nesne değişkeni tanımlarsanız , değişken adını kullanarak aralık üzerinde kolayca işlem yapabilirsiniz.

Aşağıdaki örnekte MyRange adlı bir nesne değişkeni tanımlayıp değişkene aktif çalışma kitabının Sheet1'i üzerindenki A1:D5 aralığını atanarak,tanımlanan aralık Rand() fonksiyonu kullanılarak rasgele sayılarla doldurulmaktadır.

Sub Random()

Dim myRange As Range

Set myRange = Worksheets("Sheet1").Range("A1:D5")

myRange.Formula = "=RAND()"

myRange.Font.Bold = True

End Sub

8. Çalışma Tablosundaki Bütün Hücrelerin Gösterimi

Eğer Cell özelliğini index numarası kullanmadan bir çalışma tablosu üzerinden uygularsanız , bu özellik çalışma tablosundaki bütün hücrelerin seçili olduğu bir nesne değeri döndürür.

Aşağıdaki örnek aktif çalışma kitabına bağlı Sheet1 içindeki tüm hücrelerin içeriğini siler.

Sub ClearSheet()

Worksheets("Sheet1").Cells.ClearContents

End Sub

NOT: Bu doküman Faruk Çubukçu tarafından yazılmıştır. Ticari amaçlı kullanılamaz.