

## VISUAL BASIC'te DEĞİŞKENLER

### INTEGER ( TAM SAYI )

Integer değişkenler, tam sayı değerlerini barındıran değişkenlerdir. Program kodunun ilgili modüllerinde kısmında **DefInt** ile tanımlanabilirler veya bu tip bir değişkenin sonunda % işareti yer alır. Integer türü değişkenler **-32768** ile **+32767** aralığında değer taşıyabilirler.

Örnek :

```
DefInt X, Y
Z% = 2345
X = 18572
Y = 45678 (Bu tanımlama +32767'den büyük olduğu için hatalıdır.)
```

DefInt satırında tanımlanan integer türü değişkenlerde % işareti kullanılmayabilir. Ancak DefInt satırında tanımlanmamış bir değişken integer olarak kullanılacaksa (yukarıdaki örnekte Z% değişkeni) bu durumda değişkenin sonuna % işareti eklenir.

### LONG INTEGER ( UZUN TAM SAYI )

Long Integer türü değişkenler de benzer olarak **DefLng** sözcüğü ile tanımlanabilirler veya bir değişkenin sonuna & işareti konularak da tanımlanabilir. Bu uzun tam sayı değişkenleri **-2,147,483,648** ile **+2,147,483,647** arasında değer taşıyabilirler.

Örnek :

```
DefLng A - D
A = 98765422
C = 12345678
X& = A
```

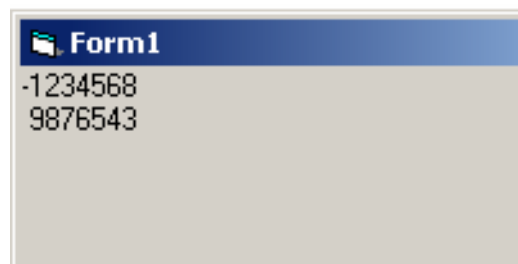
DefLng ifadesinde uzun tam sayı değişkenleri yukarıdaki örnekte tanımlandığı şekilde belirtildiğinde bu aralıktaki değişkenler (bu örnek için A, B, C ve D değişkenleri) birlikte uzun tam sayı olarak tanımlanmış olur. Değişkenleri ayrı ayrı tanımlamak ta mümkündür (DefLng A, T, V, Z şeklinde de olabilirler). Yukarıdaki ifade ile Alfabetik olarak A'dan D'ye kadar olan harfler veya bu harflerle başlayan değişken isimleri Long Integer değişken olarak tanımlanmışlardır.

### SINGLE PRECISION ( TEK DUYARLIKLI )

Bir değişkenin sonunda ! sembolü yer alıyorsa veya **DefSng** sözcüğü ile tanımlanmışsa bu değişken tek duyarlıklı özelliindedir. Taşıyabileceği değerler, **negatif** sayılar için **-3,402828E38** ile **-1,401298E-45** aralığında, **pozitif** sayılar için ise **+1,401298E-45** ile **+3,402823E38** aralığında yer alır. Tek duyarlıklı değişkenler 7. hanesine kadar hassas olup daha uzun sayılar yuvarlatılırlar. Örneğin ; X!=1234567,890123 sayısı 1234568 değerine yuvarlatılır.

Örnek :

```
Project1 - Form1 (Code)
Command1
Private Sub Command1_Click()
X! = -1234567.890123
Y! = 9876543.21
Print X!
Print Y!
End Sub
```



Form1  
-1234568  
9876543

## DOUBLE PRECISION ( ÇİFT DUYARLIKLI )

Bir değişkenin sonunda # sembolü yer alıyorsa veya **DefDbI** sözcüğü ile tanımlanmışsa bu değişkenler çift duyarlıklı değişkendir. Bu tür değişkenler, negatif sayılar için **-1,79769313486232E308** ile **-4,940656645841247E324** arasında ve pozitif sayılar için de **4,940656645841247E324** ile **1,79769313486232E308** arasında değer taşıyabilirler. Bu sayılar ilk 16 haneye kadar hassastırlar. 16 haneden uzun sayılar 16. haneye yuvarlatılır.

## STRING ( KARAKTER ZİNCİRİ )

Bir değişkenin sonunda \$ sembolü yer alıyorsa veya **DefStr** sözcüğü ile tanımlanmışsa bu değişkenler string değişkenlerdir. İçeriğinde metin türü (cümle, kelime v.b.) ifadeler içerirler. Bu nedenle bu tür değişkenler matematiksel ifadelerde kullanılmamalıdır. String değişkenler sabit uzunluklu veya değişken uzunluklu olabilirler. Sabit uzunluklu string değişkenlerde bünyesinde barındırdığı ifadenin uzunluğu **00** ile **65400** karakter, değişken uzunluklu string ifadelerde ise **0** ile **2000000000** arasında olabilir.

## CURRENCY ( PARASAL FORMAT )

Değişken ismi sonunda @ sembolü veya **DefCur** ifadesi ile tanımlanırlar. Ondalık noktadan önce 15 ve ondalık noktadan sonra 4 hane olarak formatlanırlar. Ondalık nokta öncesinde her 3 hanede bir ayraç gösterilir. Bu tür değişkenler - **922,337,203,685,477.5808** ile **922,337,203,685,477.5807** arasında değer alır ve bu formatta gösterilirler.

## DATE ( TARİH )

Tarih ve zaman türü bilgiler için kullanılırlar. Değişken ismi **DefDate** ifadesi ile tanımlanırlar. Değişkene aktarılacak değer, sayının başında ve sonunda # olacak şekilde yazılarak aktarılır. Zaman olarak **1.Ocak.100** tarihi ile **31.Aralık.9999** tarihleri arasında değerler kullanılabilir. Örnek :

```
DefDate T
T = #December 15,2002#
```

## BYTE

Byte değişken türü, **0** ile **255** arasında değerler alabilir ve **DefByte** tanımlama ifadesi ile belirlenirler.

## BOOLEAN ( MANTIKSAL DEĞİŞKENLER )

Bu değişkenler **DefBool** sözcüğü ile tanımlanırlar ve yalnızca **True** (doğru) ve **False** (yanlış) değerlerini alabilirler.

## VARIANT

Variante değişkenler **DefVar** ifadesi ile tanımlanırlar. Eğer bir değişken herhangi bir değişken tipi olarak tanımlanmamış ise kullanıldığı ilk yerde otomatik olarak variante değişken şeklinde alınır. Bu tip değişkenler her türlü değer (tarih, nümerik, string v.b.) aktarılabilir.

## KULLANICI TARAFINDAN BELİRLENEN VERİ TİPİ ( USER DEFINED TYPES )

Çeşitli tipte verileri bir paket halinde toplayarak tek bir isim altında bir veri grubu şeklinde kullanmaya yarar. Bu tür veriler program kod sayfasında **Declarations** kısmında tanımlanırlar ve **Type** ve **End Type** sözcükleri arasında yer alırlar. Örnek:

```
Type Telefon
  Ad as string * 10
  Soyad as sting * 10
  Num as Double
  Kayit as integer
End Type
```

Bu şekilde tanımlanan bir veri paketi daha sonra kullanılacağı bir modülde **Private ... as ....** ifadesi ile veya **Public .... As ....** ifadesi ile o modüle aktarılır. Örnek:

```
Private Tel as Telefon
```

Veya

```
Public Tel as Telefon
```

Daha sonra bu aktarma tanımlamasının yapıldığı modülde bu değişkenlere değer verilmesi grup adı kullanılarak yapılır. Örnek:

```
Tel.Ad = "Orhan"
Tel.Soyad = "Cerit"
Tel.Num = "5555555"
Tel.Kayit = 1
```

## DEĞİŞKEN TİPLERİNİN BELİRTİLMESİNDE DIM DEYİMİ KULLANIMI

Eğer kullanılan değişkenler yukarıda anlatıldığı şekilde **Def...** ifadeleri kullanılarak tanımlanmamışsa, **DIM** ifadesi ile de tanımlanabilirler: Örnek:

```
Dim A as Integer
Dim B as Currency
Dim C as String
```