

## 3.VİSUAL BASIC'DE DEĞİŞKEN TANIMLAMA

Değişken tanımlarken Visual Basic'te Dim bildiri deyimini kullanabiliriz. Değişkenin tanımlanması hafızada ayrılacak hafıza miktarının belirli olmasını sağlar. Eger değişkenlerin tipini belirtmeden bir kullanım yaparsak bu değişkenlerin Variant tipinde olduğu kabul edilir.Bu da hafızada gereksiz yer kaybına sebep olur.Eger tanımlanan bütün değişkenlerin tiplerinin belli olmasını isterseniz kod penceresinin General,Declarations kısmına **Option Explicit** yazılır.

Ayrıca değişken tanımlarken kullanılacak bir başka bildiri deyimide Def- bildiri deyimidir.Bu bildiri deyimi daha genel tanımlamalar yapmak için kullanılır..Def bildiri deyimleri aşağıda verilmiştir.

Def bildiri deyimi projenin general,declarations bölümünde tanımlanmalıdır

DefBool	:	Boolean	tipinde	değişken	tanımlamak	için	kullanırız.
DefByte	:	Byte	tipinde	değişken	tanımlamak	için	kullanırız.
DefInt	:	Integer	tipinde	değişken	tanımlamak	için	kullanırız.
DefLng	:	Long	tipinde	değişken	tanımlamak	için	kullanırız.
DefCur	:	Currency	tipinde	değişken	tanımlamak	için	kullanırız.
DefSng	:	Single	tipinde	değişken	tanımlamak	için	kullanırız.
DefDbl	:	Double	tipinde	değişken	tanımlamak	için	kullanırız.
DefDate	:	Date	tipinde	değişken	tanımlamak	için	kullanırız.
DefStr	:	String	tipinde	değişken	tanımlamak	için	kullanırız.
DefVar	:	Variant	tipinde	değişken	tanımlamak	için	kullanırız.

### Örnek :

```
Private Sub Form_Load()  
Dim Ad As String  
Dim Maas As Currency  
Dim D_tarihi As Date  
Dim Adres As String
```

```
Dim Sira As Integer
```

```
End Sub
```

Burada görmüş olduğunuz gibi 5 adet değişken tanımlanmaktadır. Ad değişkeni string tipinde bir değişkendir. Maaş değişkeni Currency tipinde bir değişkendir. D\_tarihi değişkeni Date tipinde bir değişkendir. Adres değişkeni string tipinde bir değişkendir. Sira değişkeni integer tipinde bir değişkendir.

Aşağıdaki satırları formun general, declarations bölümüne yazalım.

### Örnek :

```
DefInt A-C
```

```
DefStr S
```

```
DefVar K
```

```
DefDateD
```

```
DefSng V
```

Bu örnekte A , B, C ile başlayan bütün karakterler Integer türünde olmak zorundadır. D ile başlayan bütün karakterler Date türünde olmak zorundadır. V ile başlayan bütün karakterler Single türünde olmak zorundadır. K ile başlayan bütün karakterler Variant türünde olmak zorundadır. S ile başlayan bütün karakterler String türünde olmak zorundadır.

Def deyimi dim deyiminden farklıdır. Def ile sadece bir harf yada harf aralığı belirtilebilir. Burada belirtilen harf ile başlayan bütün değişkenler artık o bildiri deyimindeki tiptedir

**Örnek :**

```
Private Sub Form_Load()  
Dim ad As String , soyad As String  
Dim maas As Currency  
End Sub
```

Bu örnekte ad ve soyad isimli iki değişken string türünde ve maas değişkeni ise Currency tipindedir.

### 3.1 DEĞİŞKENLERE İSİM VERME KURALLARI

Visual Basic'de daha etkin bir programlama yapabilmek için kullanılan veri tiplerini ve veri yapılarını iyi bir şekilde bilmeliyiz. Burada önemli olan nerede hangi veri tipini kullanacak oluşumdur. Veri tiplerini anlatmadan önce kullanacağımız değişkenleri nasıl isimlendireceğimizi görelim.

**Değişkenleri isimlendirirken daha sonra bakdığınızda akılda kalabilecek ve anlamlı bir isim veriniz.**

Aşağıda verilen kurallar doğrultusunda değişkenler tanımlayabilmemiz mümkün olmaktadır. Tanımlanacak değişkenlerin ilk karakteri mutlaka bir harf ile başlamalıdır. Geri kalan karakterler ; harflerden , rakamlardan, alt çizgi karakterinden oluşabilir. Değişken isimlerinde noktalama işaretlerini , matematiksel ve mantıksal ve karşılaştırma operatörleri kullanamayız. Değişken isimleri 255 karaktere kadar uzunlukta olabilir. Aşağıda değişken tanımlamaları ile ilgili örnekler verilmiştir.

**Örnek :**

```
Adı , soyadı , yaşı , Maas98 , Dogum_Yeri , SANAT_DALI
```

Yukarıda geçerli değişken isimlerine örnek verilmiştir

**Örnek :**

```
2ADI , ad soyad , Mal+Bildirimi
```

Burada da geçersiz değişken isimlerine örnekler verilmiştir.

2ADI : Çünkü ilk karakter bir sayı ile başlamış.

Ad soyad : Değişken isminde boşluk kullanılmış

Mal+Bildirimi : Değişken isminde geçersiz bir karakter kullanılmış.

Zaten bu değişkenleri tanımlarken Visual Basic bizi hata mesajı ile uyaracaktır

## 3.2 DİZİLER

Aynı tür bilgileri bellekte tutmak için kullanabileceğimiz listelere dizi adı verilir. Dizi kullanmanın avantajı aynı tür bilgiler bir listede tutularak daha hızlı işlem yapılabilmesi sağlanmış olacaktır. Visual Basic'de dizi şu şekildedir Dizi\_adi(İndis) olacaktır.

### Örnek:

```
Private Sub Form_Load()  
    Dim ad(2) As String  
    Dim no(2) As Integer  
    ad(0) = "ali"  
    ad(1) = "ahmet"  
    ad(2) = "ebru"  
    no(0) = 133  
    no(1) = 56  
    no(2) = 67  
End Sub
```

Bu örnek'te ad ve no olmak üzere iki adet dizi tanımlanmaktadır. ad dizisi string türünde bilgileri tutacak ve no dizisi ise integer türündeki bilgileri saklayacaktır. Görmüş olgunuz gibi dizilere bilgi atama şekli dizi\_adi(indis\_sırası)=atanacak\_bilgi şeklindedir.

Eger diziyi sıfırdan değilde bir den itibaren başlatmak istersek diziyi tanımlamadan önce Option Base 1 satırını eklemeliyiz.

Burada dizi indisleri sıfırdan başlayarak tanımlama yaparken bizim belirttiğimiz değere kadardır. Yani bizim bu dizilere atayabileceğimiz veri sayısı diziyi tanımlarken belirttiğimiz indis değerinden bir fazla olacaktır.

Eğer değer atama yaparken belirttiğimiz sınırların dışına çıkarsak hata oluşur. Dizilere değer atarken dizi sınırlarını kontrol etmekle olası bir hatayı önlemiş oluruz.

### 3.2.1 STATİK DİZİLER

Bu tip dizilerde kullanılacak yer sayısı sabittir. Bu tip diziler sadece tanımlandıkları modül içerisinde kullanılabilirler. Statik dizi şu şekilde tanımlanabilir. Dim dizi\_adi(sayı) As Veri\_Tipi Tüm proje içinde kullanılacak bir dizi tanımlanmak isterse standart modülün General,Declarations bölümünde yukarıdaki şekildeki gibi tanımlanmalıdır.

#### Örnek :

```
Option Base 1

Dim a(5) As Integer

Private Sub Command1_Click()

Text1.Text = a(1) + a(2)

End Sub

Private Sub Form_Load()

a(1) = 10

a(2) = 20

End Sub
```

Bu örnekte görüldüğü gibi projenin general,declarations kısmında a() dizisi tanımlanıyor.Bu dizi tanımlanmadan önce dizi indislerinin 1 den itibaren başlayacağını belirten Option Base1 satırı koda dahil edilmiştir. Form1 yüklendiğinde bu dizinin ilk elemanına 10 sayısı ikinci elemanına 20 sayısı atanıyor. Eğer kullanıcı Command1 isimli butona tıklarsa dizinin ilk ve ikinci elemanları toplanarak Form üzerinde Text1

adlı nesnenin Text özelliğine atanıyor. Yani TextBox'ın bu sayıların toplamını göstermesi sağlanıyor.

### 3.2.2 DİNAMİK DİZİLER

Bu tip dizilerde kullanılacak yer sayısında bir sınırlama yoktur. Bu tip diziler ilk başta şğıdaki şekildeki gibi bir tanımlamaya ihtiyaç duyarlar. Dim dizi\_adı( ) As Veri\_Tipi Daha sonra bu dizi kullanılacak iken botunu belirtmek gerekir bunun içinde aşğıdaki gibi bir tanımlama yapılmalıdır. ReDim dizi\_adı(boyut ) As Veri\_Tipi Artık bu veri dizisini projemiz içinde kullanabiliriz.

#### Örnek :

```
Option Base 1

Private Sub Form_Load()

Dim s() As String

End Sub

Private Sub Command1_Click()

ReDim s(10) As String

s(1) = "Selam "

s(2) = "Ayşe"

Text1.Text = s(1) + s(2)

End Sub
```

Bu dizi tanımlanmadan önce dizi indislerinin 1 den itibaren başlayacağını belirten Option Base1 satırı koda dahil edilmiştir. Form1 yüklendiğinde s adlı bir dinamik dizi tanımlanmaktadır. Kullanıcı Command1 isimli butuna tıkladığında s dizisinin boyutu belirtilerek yeniden tanımlanmıştır. Bu dizinin ilk elemanına "Selam " degeri ikinci elemanına "Ayşe" degeri atanıyor. Dizinin ilk ve ikinci elemanları toplanarak Form üzerinde Text1 adlı nesnenin Text özelliğine atanıyor. Yani TextBox'ın bu degerlerin toplamını göstermesi sağlanıyor.

**Örnek :**

```
Option Base 1

Private Sub Form_Load()

Dim s() As String

End Sub

Private Sub Command1_Click()

ReDim s(10) As String

s(1) = "selam"

s(2) = "fatih"

Text1.Text = s(1) + s(2)

End Sub

Private Sub Command2_Click()

ReDim s(5) As String

s(2) = "fatih"

Text1.Text = s(1) + s(2)

End Sub
```

Bu örnekte yukarıdaki örnekten tek farklı yan olarak bir Command butonun arkasına yazılmış kod bulunuyor. Eger kullanıcı Command1 adlı butondan sonra bu butona tıklarsa ne olacak ona bakalım. Command2 butonuna tıklanıldığında s() dizisi tekrardan boyutu 5 olarak tanımlanıyor. s() dizisine daha önce atanmış bütün degerler siliniyor. Yani s() dizisi bir nevi boşaltılıyor ve yeniden boyutlandırılıyor. Eger s() dizisine daha önce atanmış degerler korunmak istenirse aşağıdaki şekildeki gibi bir tanımlama yapılmak zorundadır. ReDim Preserve s(boyut) As Veri\_tipi

### 3.3 TİP DÖŞÜMLERİ

Visual Basic'de kullandığımız sayısal değişkenleri diğer veri tiplerine dönüştürebiliriz. Bu işlemi yapan fonksiyonlara tip dönüşüm fonksiyonları adı verilir. Tip dönüşüm fonksiyonları aşağıda verilmiştir.

Fonksiyon	Geri	Dönen	Deger	Yaptığı	İşlem
CBool(ifade)	Boolean	Matemetiksel	ifadeyi	Boolean türüne	dönüştürür.
CByte(ifade)	Byte	Matemetiksel	ifadeyi	Byte türüne	dönüştürür.
CCur(ifade)	Currency	Matemetiksel	ifade	Currency türüne	dönüştürür.
CDate(ifade)	Date	Matemetiksel	ifade	Date türüne	dönüştürür.
CDbl(ifade)	Double	Matemetiksel	ifade	Double türüne	dönüştürür.
CDec(ifade)	Decimal	Matemetiksel	ifade	Decimal sayıya	dönüştürür.
CInt(ifade)	Integer	Matemetiksel	ifade	tam sayıya	dönüştürür.
CLng(ifade)	Long	Matemetiksel	ifade	Long türüne	dönüştürür.
CSng(ifade)	Single	Matemetiksel	ifade	Single türüne	dönüştürür. CStr(ifade) String
Matemetiksel	ifade	String	türüne	dönüştürür.	
Aşağıda	çeşitli örneklerle	tip dönüşümleri	açıklanmaya	çalışılmıştır.	

#### Örnek1:

```
A=10 , B=5 , C=10 , D=0  
check = CBool(A < B) 'check=False  
check = CBool(A > B) 'check=True  
check = CBool(A = C) 'check=True  
check = CBool(D) 'check=False  
check = CBool(B) 'check=True
```

#### Örnek 2 :

```
A=10 , B=5 , C=0  
check = CByte(A < B) 'check=0  
check = CByte(A > B) 'check=1  
check = CByte(A = C) 'check=0
```

**Örnek 3 :**

```
A=1 , B=2 , C=36000 , D=36001  
check = CDate(A) 'check=12/31/1899  
check = CDate(B) 'check=1/1/1900  
check = CDate(C) 'check=7/24/98  
check = CDate(D) 'check=7/26/98
```

**Örnek 4 :**

```
A=2.4 , B=2.5 , C=2.6 , D=3.5  
check = CInt(AB) 'check=2  
check = CInt(B) 'check=2  
check = CInt(C) 'check=3  
check = CInt(D) 'check=4
```

### 3.4 VERİ TIPLERİ 1

**Currency :**

Sayısal tipdeki veriler için tanımlanmış özel bir veri tipidir. Hafızada 8 byte yer kaplarlar. 4 hane ondalık kısmı olmak üzere toplam 19 haneden oluşur. (nokta hariç) Alabileceği maximum değerler -922,337,203,685,477.5808 ile 922,337,203,685,477.5807 arasındadır. @ sembolü veya DefCur bildirimini ile Currency tipinde değişken'ler tanımlanabilirler.

**Örnek :**

```
Private Sub Form_Load()  
Dim Maas As Currency , Borc As Currency
```

```
Maas = 500000000
```

```
Borc = 68000000
```

```
Zayi@ = 340000.56
```

```
End Sub
```

**Date :**

Tarih türündeki bilgileri kullanmak için oluşturulmuş bir veri tipidir. Hafızada 8 byte yer kaplarlar. 1 Ocak ile 31 Aralık 9999 arasındaki tarihleri kullanabilirsiniz. DefDate bildirisi ile tarih türünde değişkenler tanımlayabiliriz.

**Örnek :**

```
Private Sub Form_Load()
```

```
Dim Dogum_tarihi As Date , Evlenme_Tarihi As Date
```

```
Dogum_tarihi =#Dec,9,1977#
```

```
Evlenme_Tarihi=#May,15,1998#
```

```
End Sub
```

**Boolean :**

Mantıksal veri tipleri için kullanılır. İki seçenektan birisini alabilir. Bunlar True veya False değerleridir. Bellekte 2 byte yer işgal ederler. Boolean tipindeki bir değişkeni tanımlamak için DefBool sözcüğü kullanılır.

**Örnek :**

```
Private Sub Form_Load()
```

```
Dim Durum As Boolean , Duyum As Boolean
```

```
Durum = True  
Duyum = False  
End Sub
```

**Byte :**

0 ile 255 arasındaki tamsayıları ifade etmek için kullanılacak veri tipidir. DefByte deyimi ile byte tipinde değişkenler tanımlanabilir.

**Örnek :**

```
Private Sub Form_Load()  
Dim Durum As Byte , Duyum As Byte  
Durum =23  
  
Duyum = 143  
  
End Sub
```

**String :**

Metin türü bilgileri saklamak için kullanılacak veri türüdür. 16 bitlik versiyonda 0 ile 65538 arasında , 32 bitlik versiyonda ise 0 ile 2,000,000,000 arasında karakter alabilir. String türünde değişken tanımlamak için Defstr veya \$ sembolü kullanılır.

**Örnek :**

```
Private Sub Form_Load()  
Dim Adi As String , Soyadi As String  
  
Adı ="Kemal"
```

```
Soyadı = "Tas"  
No$="1323970067"  
End Sub
```

**Variant :**

Bu tipte tanımlanmış bir değişkene herhangi bir tip'te veri yüklenebilir. Yani ne tür bir veri gireceğimizi bilmediğimiz değişkenleri Variant tipinde tanımlamalıyız. Bu tür değişkenler hafızada 16 byte tan fazla yer kaplarlar. DefVar bildiri sözcüğü ile Variant türünde değişkenler tanımlıyabiliriz.

Variant türünde değişkenler tanımlamak fazla kullanışlı değildir. Çünkü hafızada fazla yer kaplarlar.

**Örnek :**

```
Private Sub Form_Load()  
Dim Ad As Variant , Maas As Variant , Tel As Variant  
Dim Dogum_tarihi As Variant  
Ad="Aydın Kale"  
Maas=200000  
Tel="500-45-00"  
Dogum_tarihi=#Apr,3,1970#  
End Sub
```

**3.5 VERİ TIPLERİ 2****Integer :**

Visual Basic'te tam sayı değişkenleri tanımlamak için kullanılır. Hafızada 2 byte yer

kaplarlar. Alabileceği değer aralığı  $-32768$  ile  $+32767$  arasındadır. DefInt bildiri deyimi ile tanımlanabilirler. Ayrıca bir değişkenin sonunda % karakteri bulunuyorsa bu değişken integer tipindedir.

**Örnek :**

```
Private Sub Form_Load()  
Dim Maas As Integer  
DefInt A-C  
Oran% = 100  
A_sayı = 100  
B_sayı = Oran*A_sayı  
C_sayı = B_sayı + A_sayı - 1000  
Maas= 32767  
End Sub
```

Eğer burada tanımlanan değişkenlere daha büyük sayılar atanırsa overflow oluşur.

**Long :**

Daha büyük bir aralıkta integer yani tamsayı tanımlamak için kullanabileceğimiz bir veri tipidir. Hafızada 4 byte yer kaplar. Kullanılabilecek uç değerler  $+2,147,483,647$  ile  $-2,147,483,648$  dir. Long tipinde bir değişken tanımlamak için DefLng bildirimini veya değişken sonunda & karakterini kullanabiliriz.

**Örnek :**

```
Private Sub Form_Load()  
DefLng A-B  
Bölüm=50000
```

```
cıkan&=600000  
Kalan& = (cıkan / Bölüm ) * 10000  
End Sub
```

### Single :

Tam sayı olmayan küsüratlı sayılar için kullanabileceğimiz bir veri tipidir. Kayan-noktalı sayı olarak isimlendirilir. Single tipindeki veriler bellekte 4 byte yer kaplarlar. Negatif sayılar için alabileceği aralık -3.402823E38 ile -1.401298E-45 ,pozitif sayılar için alabileceği aralık 1.401298E-45 ile 3.402823E38 arasındadır.Single tipinde veri tanımlamak için DefSgn bildiri mi veya değişken sonuna ! karakteri konur. 7 haneye kadar hassastır.Daha sonrası yuvarlatılır.

### Örnek :

```
Private Sub Form_Load()  
DefSgn A-B  
Bölüm=50000  
Cıkan!=600000  
End Sub
```

### Double :

Visual Basic'te kullanılabilecek en büyük sayısal değerlerin veri tipidir. Hafızada 8 byte yer kaplarlar. 16 haneye kadar hassastırlar. Maximum alabileceği değerler pozitif sayılar için 4.94065645841247E-324 ile 1.797693134862232E308 , negatif sayılar için de -1.797693134862232E308 ile -4.94065645841247E-324 arasındadır. DefDbl bildiri mi veya # sembolü ile double tipinde değişkenler tanımlanabilir

**Örnek :**

```
Private Sub Form_Load()  
DefDb1 A-K  
Darı=50000  
Bugday=600000  
Arpa=340.56  
End Sub
```

**3.6 VERİ YAPILARI****Const :**

Bunlar program içinde degeri deęiřtirilemeyen sabitlerdir. Public ve Private tipinde sabitler tanımlanabiliir. Public sabitlere tüm modüller içinden ulaşılabilir.Private türündeki sabitler ise sadece tanımlandıkları modül içersinde geçerlidirler.Ayrıca Visual Basic içinde tanımlanmış çok sayıda sabit vardır. Bunlara CONSTANT.TXT dosyasını açarak inceleyebiliriz.

**Örnek :**

```
Const şehir="İstanbul"  
Const Ülke="Turkey"  
Const posta_kodu=34650  
Const tek_kod=212
```

**Type - End Type Yapısı :**

Type yapısını kullanarak programıcı farklı veri tiplerini kullanarak kendi veri yapısını oluşturabilir. Bu C deki Struct yapısına benzetilebilir. Bu yeni veri tipine record adı verilir. Herhangi bir modülün Declarations kısmında aşağıdaki gibi bir tanımlama yapabiliriz.

**Örnek :**

```
Type Ogrenci
Ad As String *10
Soyad As String *12
Not As Byte
Kredi As Integer
End Type
```

'Ogrenci veri tipi toplam hafızada 25 byte yer kaplamaktadır. Bu veri tipini kullanmak için 'Ogrenci tipinde değişkenler tanımlamak gerekmektedir.

```
Private Sub Form_Load()
Dim A As Ogrenci
Dim B As Ogrenci
```

'Bu değişkenlere bilgi atamak aşağıdaki şekilde gibidir.

```
A.Ad="Ali"
A.Soyad ="Armer"
End Sub
```

String türü değişkenlere sabit bir uzunlukta yer ayırmak istersek aşağıdaki şekilde bir tanımlama yapmalıyız.

```
Dim Name As String *12
Dim Address As String *50
```