

## Sabitler


Sabitler içeriği sabit olan değer ve ifadelerin saklanması amacıyla kullanılırlar. C# dilinde sabitleri **const** ifadesi tanımlamak mümkündür.

Sabit değerleri tanımlamada kullanılır. Değerleri tanımlandıkları anda verilmelidir.

`const int rakam = 50;` satırında rakam isimli sabiti int türünden tanımlayarak değerini 50 olarak belirledik.

`const int rakam = 50;` const ifadesi ile tanımlanmış bir sabitin değeri  
`rakam = 10;` derleme esnasında değiştirilemez.

Değiştirilmeye çalışılırsa derleme hatası verecektir.

Code	Description
 CS0131	The left-hand side of an assignment must be a variable, property or indexer

## Değişkenler

Program içerisinde üretilen değerleri bellekte geçici olarak saklamak için kullanılırlar. Bir değişken tanımlarken değişken ismi ve değişken tipi belirtilerek tanımlanırlar.

`DeğişkenTipi DeğişkenAdı = İlkDeğer ;`

İlk değer kullanımı zorunlu değildir. İstenirse program içerisinde daha sonra değer ataması yapılabilir.

```
string adsoyad;  
adsoyad = "Özcan EKEN";
```

Aynı satırda birden fazla değişkeni tanımlamak mümkündür.

```
string ad, soyad;  
ad = "Özcan" ;  
soyad = "EKEN";  
  
int a = 5, b = 10, c;
```

Değişkenler tanımlanırken;

- Değişken ismi harf ile başlamalıdır. Diğer karakterler harf, rakam ya da özel karakterler olabilir.
- Değişken ismi en fazla 255 karakter olabilir.
- Değişken ismi tanımlanırken boşluk bırakılmamalıdır. Bunun yerine \_ (alt çizgi) gibi özel karakterler kullanılabilir.
- C#.Net içerisinde büyük küçük harf ayrımı olduğundan dolayı değişkenler hangi şekilde tanımlandıysa, program içerisinde o şekilde kullanılmalıdır.
- C#.Net içerisinde kullanılan komutlar değişken ismi olarak kullanılamaz. (AND, OR, NOT gibi)

## Değişken Tipleri

Sayısal türler:

Tür	Boyut	Kapasite	Örnek
byte	1 bayt	0, ..., 255 (tam sayı)	byte a=5;
sbyte	1 bayt	-128, ..., 127 (tam sayı)	sbyte a=5;
short	2 bayt	-32768, ..., 32767 (tam sayı)	short a=5;
ushort	2 bayt	0, ..., 65535 (tam sayı)	ushort a=5;
int	4 bayt	-2147483648, ..., 2147483647 (tam sayı)	int a=5;
uint	4 bayt	0, ..., 4294967295 (tam sayı)	uint a=5;
long	8 bayt	-9223372036854775808, ..., 9223372036854775807 (tam sayı)	long a=5;
ulong	8 bayt	0, ..., 18446744073709551615 (tam sayı)	ulong a=5;
float	4 bayt	$\pm 1.5 \cdot 10^{-45}$ , ..., $\pm 3.4 \cdot 10^{38}$ (reel sayı)	float a=5F; veya float a=5f;
double	8 bayt	$\pm 5.0 \cdot 10^{-324}$ , ..., $\pm 1.7 \cdot 10^{308}$ (reel sayı)	double a=5; veya double a=5d; veya double a=5D;
decimal	16 bayt	$\pm 1.5 \cdot 10^{-28}$ , ..., $\pm 7.9 \cdot 10^{28}$ (reel sayı)	decimal a=5M; veya decimal a=5m;

#### Metinsel Türler :

Tür	Boyut	Açıklama	Örnek
char	2 bayt	Tek bir karakteri tutar.	char a='h';
string	Sınırsız	Metin tutar.	string a="Buraya Bir Metin Gelecektir.";

#### bool

Koşullu yapılarda kullanılır. Bool türünden değerlere true, false gibi ifadeler örnek verilebilir.

#### Datetime

İçinde zaman barındıran değişken tipidir.

Tür	Boyut	Açıklama	Örnek
<code>DateTime</code>	8 byte	Tarih ve Zaman Tutar	<code>Datetime zaman = DateTime.now</code>

#### object

Bu değişken türüne her türden veri atanabilir.

## Tip Dönüşümü

C# dilinde tip dönüşümü(Type Conversion), bir değişken tipindeki verinin başka bir değişken tipine dönüştürülmesidir. Değişken tiplerini dönüştürmek amacıyla Convert metodu kullanılır. Convert metodunun bazı alt metotları şunlardır.

- `Convert.ToString()` : String veri tipine dönüştürmek için kullanılır.
- `Convert.ToInt16()` : Short (kısa tam sayı) veri tipine dönüştürmek için kullanılır.
- `Convert.ToInt32()` : Integer veri tipine dönüştürmek için kullanılır.
- `Convert.ToInt64()` : Long (uzun tam sayı) veri tipine dönüştürmek için kullanılır.
- `Convert.ToChar()` : Char veri tipine dönüştürmek için kullanılır.
- `Convert.ToBoolean()` : Boolean veri tipine dönüştürmek için kullanılır.
- `Convert.ToByte()` : Byte veri tipine dönüştürmek için kullanılır.
- `Convert.ToSingle()` : Single veri tipine dönüştürmek için kullanılır.
- `Convert.ToDouble()` : Double veri tipine dönüştürmek için kullanılır.
- `Convert.ToDecimal()` : Decimal veri tipine dönüştürmek için kullanılır.
- `Convert.ToDateTime()` : Tarih ve zaman veri tiplerine dönüştürmek için kullanılır.

## Örnek

```
static void Main(string[] args)
{
    const int urunfiyati = 110;
    const double kdvlifiyat = urunfiyati + (urunfiyati * 0.18);
    byte adet;
    Console.Write("Ürün adedini giriniz: ");
    adet = Convert.ToByte(Console.ReadLine());
    Console.WriteLine("Toplam Fiyat= " + kdvlifiyat * adet);
    Console.ReadKey();
}
```

C:\Users\Özcan\source\repos\ConsoleApp1\ConsoleApp1\bin\Debug\ConsoleApp1.exe

```
Ürün adedini giriniz: 8
Toplam Fiyat= 1038,4
```

## Örnek

```
static void Main(string[] args)
{
    const double PI = 3.1428;
    double alan, cevre, yaricap;
    Console.BackgroundColor = ConsoleColor.Yellow; //Arkaplan Rengi
    Console.ForegroundColor = ConsoleColor.Blue; //Yazı Rengi
    Console.Clear(); // Console u Temizler
    Console.Write("Yarıçapı Giriniz = ");
    yaricap = Convert.ToDouble(Console.ReadLine());
    cevre = 2 * PI * yaricap;
    alan = PI * (Math.Pow(yaricap, 2));
    Console.WriteLine("Dairenin Çevresi =" + cevre);
    Console.WriteLine("Dairenin Alanı =" + alan);
    Console.ResetColor(); // Arkaplan ve Yazı Rengini Varsayılan hale geri döndürür
    Console.ReadKey();
}
```

C:\Users\Özcan\source\repos\ConsoleApp1\ConsoleApp1\bin\Debug\ConsoleApp1.exe

```
Yarıçapı Giriniz = 10
Dairenin Çevresi =62,856
Dairenin Alanı =314,28
```