

## 1. DİZİLER

Program yazarken birbiriyle ilgili ve aynı türde olan verileri saklayabilmek için dizi şeklindeki veri yapılarına ihtiyaç duyulur.

Örneğin, bir sınıftaki öğrencilerin not ortalamalarını saklamak için dizi kullanılabilir.

Sınıf mevcudu 100 olan bir sınıftaki bütün öğrenciler için ayrı ayrı değişken tanımlamak zorunda kalmaktansa, 100 elemanlı tek bir dizi tanımlamak çok daha pratik olacaktır.

### 1.1. C# dilinde dizi tanımlama

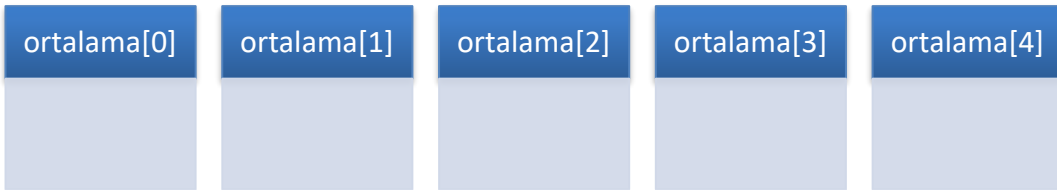
C# dilinde diziler aşağıdaki gibi tanımlanır.

```
var dizi-adi = new degişken-tipi[eleman-sayısı];
```

Örnek olarak, aşağıda 5 elemanlı bir tamsayı dizisinin nasıl tanımlandığı gösterilmiştir.

```
var ortalama = new int[5];
```

Yukarıdaki kod işletildiğinde hafızada 5 ayrı tamsayının saklanabileceği bir dizi için yer ayrılmış olur. Ancak dizinin elemanlarına hiçbir değer atanmadığı için dizi henüz boştur.



Yukarıdaki tek satırlık tanımlama ile sanki adları ortalama[0], ortalama [1], ortalama[2], ortalama[3] ve ortalama[4] olan 5 ayrı değişken oluşturulmuş gibi olur.

Fark edileceği üzere, dizinin ilk elemanının sıra numarası (indeksi) 0 iken, son elemanın sıra numarası 4'tür. Yani eleman sayısının bir eksiğidir.

**Örneğin 100 elemanlı notlar isimli dizinin ilk elemanı notlar[0] iken, son elemanın notlar[99] olacaktır.**

## 1.2. Dizi elemanlarına başlangıç değeri atama

Diziler tanımlanırken dizilerin başlangıç değeri de aşağıdaki şekilde verilebilir. Bu durumda dizinin eleman sayısı ayrıca belirtilmez, otomatik olarak hesaplanır.

### Örnek

```
var ortalama = {3, 5, 2, 8, 9, 6};
```

Yukarıdaki örnekte 6 elemanlı bir tamsayı dizisi tanımlanmıştır ve her bir elemanın başlangıç değeri atanmıştır. Bu tanımlamadan sonra elemanların değerleri şöyle olacaktır.

ortalama[0] = 3

ortalama[1] = 5

ortalama[2] = 2

ortalama[3] = 8

ortalama[4] = 9

ortalama[5] = 6

### Örnek

Aşağıdaki örnekte, başlangıç değeri tanımlanırken verilen 10 elemanlı bir dizinin elemanları sırayla ekrana yazdırılmaktadır.

```
var sayilar = {36, 58, 72, 48, 29, 61, 83, 41, 19, 27};  
for (var i = 0; i < 10; i++) {  
    Console.WriteLine(sayilar[i]);  
}
```

Dizinin elemanlarının sıra numaraları (indeksleri) 0 – 9 aralığındadır. Bu yüzden bütün elemanları ekrana yazdırabilmek için for döngüsünün döngü değişkeni 0-9 arasında değerler alacak şekilde ayarlanmıştır. Döngünün içerisinde dizinin kaçınıcı elemanının ekrana yazdırılacağı döngü değişkeninden alınmaktadır.

Ekran çıktısı aşağıdaki gibi olacaktır.

```
36
58
72
48
29
61
83
41
19
27
```

### Örnek

Aşağıdaki örnekte, başlangıç değerleri tanımlanırken verilen 10 elemanlı bir dizinin elemanları ters sırayla (sondan başa doğru) ekrana yazdırılmaktadır.

```
var sayilar = {36, 58, 72, 48, 29};
for (var i = 4; i >= 0; i--) {
    Console.WriteLine(sayilar[i]);
}
```

5 elemanlı bu dizinin elemanlarını sondan başa doğru ekrana yazdırmak için for döngüsünün döngü değişkeni 4'ten 0'a doğru birer birer azalan değerler alacak şekilde ayarlanmıştır.

Ekran çıktısı aşağıdaki gibi olacaktır.

```
29
48
72
58
36
```

### Örnek

Aşağıdaki örnekte, başlangıç değerleri tanımlanırken verilen 10 elemanlı bir dizinin en büyük elemanı bulunup ekrana yazılmaktadır.

```
var sayilar = {36, 58, 72, 48, 29, 61, 83, 41, 19, 27};  
var enBuyuk = sayilar[0];  
for (var i = 1; i < 10; i++) {  
    if (enBuyuk < sayilar[i]) enBuyuk = sayilar[i];  
}  
Console.WriteLine("En büyük sayı: " + enBuyuk);
```

Ekran çıktısı aşağıdaki gibi olacaktır.

```
En büyük sayı: 83
```