

Konsol Ekranı

C# içerisinde görsel ortamda Windows Form uygulamaları vasıtası ile kod yazabileceğimiz gibi görsel özellikleri olmayan uygulamaları da konsol ekranında yazabiliriz.

Konsol ekranında kullanılan fonksiyonlar **System.Console** sınıfına ait olan fonksiyonlardır.

Konsol ekranında kod satırları..

```
namespace ConsoleApp1
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            // Kod Satırları
        }
    }
}
```

Bir C# programı Main metodunu çağırarak başlar.

....blokları arasına yazılır

Write ve WriteLine Metotları

Konsol ekranında yazdırma işlemleri için **Write** ve **WriteLine** metotları kullanılır.

Write Metodu, kendisine gönderilen değeri konsol ekranında aynı satıra yazdırmak için kullanılan bir metottur.

WriteLine metodu ise, kendisine gönderilen değeri satır sonu karakterini de ekleyerek konsol ekranında aynı satıra yazıp imleci bir alt satıra konumlandırmak için kullanılır.

```
Console.Write("Bilgisayar");
Console.Write("Programcılığı");

Console.WriteLine("Bilgisayar");
Console.WriteLine("Programcılığı");

Console.WriteLine("Bilgisayar");
Console.WriteLine("Programcılığı");
```

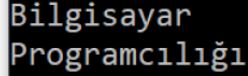
Kaçış Karakterleri

C# dili içerisinde \ karakterinin ardından kullanılarak belli bir eylem yerine getiren karakterlere Escape (kaçış) karakterleri ismi verilir.


Başlıca Kaçış karakterleri;

Karakter	Açıklama	Karakter	Açıklama
\a	Bip sesi çıkartır	\t	Yatay Tab
\b	BackSpace	\v	Dikey Tab
\f	Sonraki Sayfa	\'	Tek Tırnak
\n	Alt satıra iner	\"	Çift Tırnak
\r	Satır Başı	\\	? (Ters Bölü İşareti)

```
Console.Write("Bilgisayar\nProgramcılığı");
```



```
Console.Write("Bilgisayar\tProgramcılığı");
```



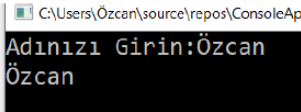
Read ve ReadLine Metotları

Konsol ekranında değer okuma işlemleri için **Read** ve **ReadLine** metotları kullanılır.

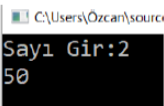
ReadLine Metodu, konsol ekranında değer okumak için kullanılan bir metottur. Geriye string değişken türünde değer döndürdüğünden dolayı string değişken tiplerindeki değerleri okumak için kullanılır.

Read metodu ise, değerlerden ilk karakterin ASCII karakter kodunu geriye döndürür.

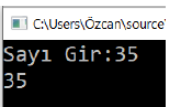
```
string ad;  
Console.Write("Adınızı Girin:");  
ad = Console.ReadLine();  
Console.WriteLine(ad);
```



```
int sayi;  
Console.Write("Sayı Gir:");  
sayi = Console.Read();  
Console.WriteLine(sayi);
```



```
int sayi;  
Console.Write("Sayı Gir:");  
sayi = Convert.ToInt32(Console.ReadLine());  
Console.WriteLine(sayi);
```



Örnek

```
static void Main(string[] args)
{
    String numara1, numara2;
    Console.Write("1. Numarayı Girin: ");
    numara1 = Console.ReadLine();
    Console.Write("2. Numarayı Girin: ");
    numara2 = Console.ReadLine();
    Console.WriteLine();
    Console.WriteLine("Girilen Numaralar {0} ve {1}", numara1, numara2);
    Console.WriteLine();
    Console.WriteLine("Girilen Numaralar " + numara1 + " ve " + numara2);
    Console.ReadLine();
}
```

C:\Users\Özcan\source\repos\ConsoleApp1\ConsoleApp1

```
1. Numarayı Girin: 5
2. Numarayı Girin: 6

Girilen Numaralar 5 ve 6

Girilen Numaralar 5 ve 6
```

Örnek

```
static void Main(string[] args)
{
    int sayi1, sayi2;
    Console.Write("Birinci sayıyı girin : ");
    sayi1 = Convert.ToInt32(Console.ReadLine());
    Console.Write("İkinci sayıyı girin : ");
    sayi2 = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Sayıların Toplamı = " + (sayi1 + sayi2));
    Console.ReadLine();
}
```

C:\Users\Özcan\source\repos\ConsoleApp1\ConsoleApp1\b

```
Birinci sayıyı girin : 10
İkinci sayıyı girin : 20
Sayıların Toplamı = 30
```

ReadKey Metodu

Konsol uygulamalarında gerekli metotlar kullanılmadığı sürece, uygulamamız çalıştırıldığında konsol ekranında sonuçlar görüntülenip, uygulama penceresi hızlıca kapanmaktadır. Sonuçların kullanıcılar tarafından görünmesi ve uygulama penceresini istenildiği zaman kapatılabilmesinin sağlanması gerekmektedir. Bunun için Read, ReadLine veya ReadKey metotlarını kullanabiliriz.

ReadKey metodu, konsol ekranında basılan herhangi bir tuşu okumak amacıyla kullanılır. Herhangi bir tuşa basılıncaya kadar konsol ekranı bekletir. Ardından bir sonraki kodu çalıştırır.

```
Console.ReadKey();
```

ConsoleKeyInfo Yapısı

Konsol ekranında Read metodunu kullanarak basılan karakteri okuyabilir ve char metodu ile dönüşüm yaparak hangi karaktere basıldığı bilgisine erişebiliriz.

```
static void Main(string[] args)
{
    char karakter;
    Console.Write("Bir tuşa basarak ardından Enter'a basınız:");
    karakter = Convert.ToChar(Console.ReadLine());
    Console.WriteLine("Basılan Tuş : " + karakter);
    Console.ReadKey();
}
```

Ancak her ne kadar Read metodu ile konsol ekranında basılan karakterleri okuyabilirsek de Enter, F1, Spacebar gibi tuşları okuyamayız. Bunun için ReadKey ve ConsoleKeyInfo yapısını kullanacağız.

ReadKey metodu, konsol ekranında basılan herhangi bir tuşu okumak amacıyla kullanılırken, ConsoleKeyInfo yapısı ise konsol ekranında basılan tuş ile ilgili bilgi almak amacı ile kullanılır.

Örnek

```
static void Main(string[] args)
{
    Console.WriteLine("Bir Tuşa Basınız!!!");
    ConsoleKeyInfo Tus = Console.ReadKey(true);
    Console.WriteLine(Tus.Key.ToString() + " tuşuna bastınız.");
    Console.ReadKey();
}
```

C:\Users\Özcan\source\repos\ConsoleApp1\ConsoleA

```
Bir Tuşa Basınız!!!
Enter tuşuna bastınız.
```

Örnek

```
static void Main(string[] args)
{
    string ad,soyad;
    Console.Write("Adınızı Girin : ");
    ad=Console.ReadLine();
    Console.Write("Soyadınızı Girin : ");
    soyad = Console.ReadLine();
    Console.Clear();
    Console.WriteLine("Ekranı yazdırmak için F2, Çıkmak için ESC tuşuna basınız");
    ConsoleKeyInfo Tusbilgi = Console.ReadKey(true);
    if (Tusbilgi.Key == ConsoleKey.Escape)
    {
        Environment.Exit(0);
    }
    else if (Tusbilgi.Key == ConsoleKey.F2)
    {
        Console.Write(ad + " " + soyad);
        Console.ReadKey();
    }
}
```