**C# String Metotları Nedir?**

String metotları, metinsel tanımlamalarımız üzerinde arama yapma, karşılaştırma, değiştirme, kopyalama gibi çeşitli işlemleri yapmamıza imkan sağlar. Kullanabileceğimiz bu metot türlerini çeşitli alt başlıklar halinde örneklendirerek göreceğiz.

**Genel String Metotları**

En çok kullanacağımız ve genel olarak herkes tarafından sıklıkla kullanılan metotlara yer vereceğim bu başlık altında ilk bölüm içerisinde göreceğimiz metot türleri şu şekildedir:

* **Length()**: Dizi yapılarında da kullandığımız bu metodu değişkenimizden sonra belirttiğimiz zaman verimizin kaç karakterden oluştuğunu gösterir.
* **Substring(Parametre)**: Parametre kısmında belirteceğimiz indis numarasından sonra bulunan karakterlerin işleme alınmasını sağlar. Bir diğer kullanım örneği olarak **Substring(Parametre1, Parametre2)** olarak tanımlama yapabiliriz, bu tanımlama sonrasında verilen değerler arasında bulunan indis elemanlarında işlemini gerçekleştirir.
* **ToLower()**: İsminden de anlaşılacağı üzere metinsel verimizin karakterlerini küçük harflere dönüştürür.
* **ToUpper()**: Metinsel karakterlerimizin içerisinde bulunan harfleri büyük harfe dönüştürür.

Bu tanımları aklımızda daha kolay canlandırabilmek için basit bir kullanış örneği oluşturarak kod çıktısına göz atarsak daha yararlı olacaktır.

string myName = **null**;

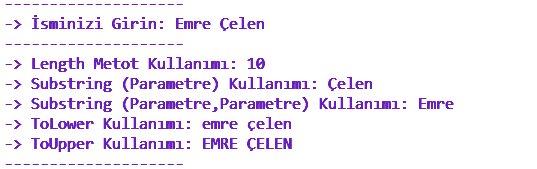
string ayrac = new string('-', 20); // İstenilen karakteri verilen değer kadar oluşturuyoruz.

Console.Write("{0}\n-> İsminizi Girin: ", ayrac);

myName = Console.ReadLine(); // Klavyeden Değer Alma

Console.WriteLine("{0}\n-> Length Metot Kullanımı: {1}\n-> Substring (Parametre) Kullanımı: {2}\n-> Substring (Parametre,Parametre) Kullanımı: {3}\n-> ToLower Kullanımı: {4}\n-> ToUpper Kullanımı:{5}\n{6}",ayrac,myName.Length,myName.Substring(5),myName.Substring(0,4),myName.ToLower(),myName.ToUpper(),ayrac);

**Kod Çıktısı:**

C# String Metotları #1

C# String metotları konumuza devam ederek genel yapıda kullanacağımız metot yapılarını listeleyelim.

* **Insert()**: Değişkenimiz içerisinde belirli bir indis numarasından itibaren başına, ortasına veya metin yapımızın sonuna veri eklemek istediğimiz zaman kullanırız.
* **Replace()**: Türkçe karşılığı olarak değiştirmek diyebileceğimiz bu metot, kullanımda belirteceğimiz iki parametrenin ilkini metin içerisinde arayarak varsa ikinci parametredeki değer ile değiştirir.
* **Remove()**: Metodun isminden de anlaşılacağı gibi belirtilen indisten itibaren veya indis aralığında silme işlemi gerçekleştirir.
* **Clone()**: Veri yapımız içerisinde bulunan değeri başka bir değişkene aktarmak istediğimiz zaman kullanırız.
* **Trim()**: Metinsel verimiz içerisinde başlangıcında ve sonunda boşluk var ise kaldırmak için kullanılır.

string myText = " Buralar Gidecek Buralar Kalacak ";

string myCloneText = **null**;

string ayrac = new string('-', 20); // İstenilen karakteri verilen değer kadar oluşturuyoruz.

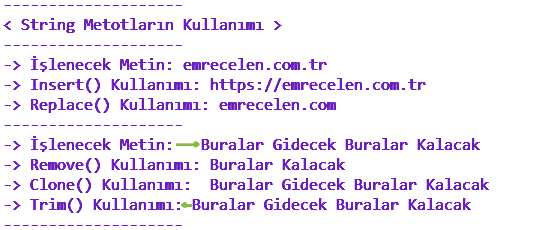
Console.WriteLine("{0}\n< String Metotların Kullanımı >\n{1}", ayrac, ayrac);

Console.WriteLine("-> İşlenecek Metin: {0}\n-> Insert() Kullanımı: {1}\n-> Replace() Kullanımı: {2}\n{3}",myBlog,myBlog.Insert(0,"https://"),myBlog.Replace(".tr",""),ayrac);

myCloneText = (string)myText.Clone();

Console.WriteLine("-> İşlenecek Metin: {0}\n-> Remove() Kullanımı: {1}\n-> Clone() Kullanımı:{2}\n-> Trim() Kullanımı: {3}\n{4}",myText,myText.Remove(0,18),myCloneText,myText.Trim(),ayrac);

**Kod Çıktısı:**

C# String Metotları #2

**Not:** Trim metodunda sadece belirli bir alanın boşluğunu kaldırmak istediğimiz zaman bu alan başlangıç kısmı ise **TrimStart()** metodunu son alanı ise **TrimEnd()** metodunu kullanırız.

**String Değerleri Karşılaştırma Metotları**

String veri tipi içerisinde bulunan veriyi herhangi bir değer veya değişken ile karşılaştırmak istediğimiz zaman kullanabileceğimiz çeşitli metotlar yer almaktadır. Bu metotlar şu şekildedir:

* **IndexOf(Parametre)**: Belirteceğimiz parametre, arayacağımız metinimiz içerisinde kaçıncı karakterinde ilk kez kullanıldığını gösterir.
* **LastIndexOf(Parametre)**: IndexOf metodunda olduğu gibi arama işlemini gerçekleştirir fakat bu sefer aranan parametrenin en son bulunduğu yeri gösterir.
* **Equals(Parametre)**: Parametre olarak vereceğimiz değer ile metodu kullandığımız değerin eşit olup olmadığını karşılaştırmak için kullanılır.
* **StartWith(Parametre)**: Belirtilen parametre içerisinde bulunan değer ile karşılaştırma yaptığımız metinin başlayıp başlamadığı kontrol edilir.
* **EndsWith(Parametre)**: StartWith metodunda olduğu gibi işlemleri aynı şekilde gerçekleştirir fakat bu sefer belirtilen parametre ile bitip bitmediğini kontrol eder.
* **Contains(Parametre)**: Bu metodu intihal programları gibi düşünebilirsiniz. Parametre olarak verilen metinin, sorguladığımız metin içerisinde bulunup bulunmadığını kontrol eder.

string myName = **null**;

string userName, userPass;

string ayrac = new string('-', 20); // İstenilen karakteri verilen değer kadar oluşturuyoruz.

Console.Write("{0}\n< Sisteme Giriş Yap >\n{1}\n-> Kullanıcı Adını Gir: ",ayrac,ayrac);

userName = Console.ReadLine();

Console.Write("-> Kullanıcı Şifresini Gir: ");

userPass = Console.ReadLine();

Console.WriteLine(ayrac);

**if**(userName.Equals("root") && userPass.Equals("admin")){

myName = "EMRE ÇELEN";

Console.WriteLine("-> Giriş Onaylandı\n{0}",ayrac);

Console.WriteLine("-> İşlenecek Metin: {0}\n{1}\n-> IndexOf('E') Kullanımı: {2}\n-> LastIndexOf('E') Kullanımı: {3} ",myName,ayrac,myName.IndexOf("E"),myName.LastIndexOf("E"));

Console.WriteLine("-> StartWith('RE') Kullanımı: {0}\n-> EndsWith('EN') Kullanımı: {1}\n-> Contains('ÇELEN') Kullanımı: {2}\n{3}", myName.StartsWith("RE"), myName.EndsWith("EN"),myName.Contains("ÇELEN"),ayrac);

}

**else**

Console.WriteLine("-> Hatalı Giriş Yaptınız\n{0}", ayrac);

**Kod Çıktısı:**

