第10章 string型とchar型　その3

stringクラスの静的メソッドにJoinメソッドがあります。これは、文字列の配列を指定された文字列で結合します。

Join静的メソッドには2つのオーバーロードバージョンがあります。今までも、あるメソッドにはいろいろなバージョンがあることを暗黙のうちに解説していました。オーバーロードについては後の章で詳しく解説します。

```
public static string Join (
    string separator,
    string[] value
)
```

最初のバージョンです。value文字列配列をseparatorを使って連結します。

次のような文字列配列があったとします。

```
str[0] = "abc";
str[1] = "def";
str[2] = "ghi";
```

これに対して

```
string strx = string.Join("--", str);
```

とすると、文字配列strは"--"で連結されてstrxは"abc--def--ghi"となります。

次のバージョンは、

```
public static string Join (
    string separator,
    string[] value,
    int startIndex,
    int count
)
```

最初のバージョンは、配列の要素を全部結合しましたが、このバージョンでは結合する配列のスタートの要素と、結合する要素数を指定できます。

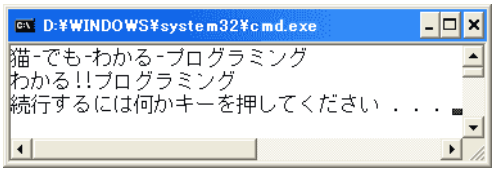
```
string strxx = string.Join("--", str, 1, 2);
```

と、するとstr[2]から2つの要素を"--"で結合します。(strxxは"def--ghi"となります)

では、簡単なサンプルを見てみましょう。

```
// join01.cs

using System;

class join01
{
    public static void Main()
    {
        string[] strarray = { "猫", "でも", "わかる", "プログラミング"};


        string str1 = string.Join("-", strarray);
        Console.WriteLine(str1);

        string str2 = string.Join("!!", strarray, 2, 2);
        Console.WriteLine(str2);
    }
}
```

実行結果は次のようになります。



Join静的メソッドのように、文字列を結合しますが、区切りの文字列を指定しないConcat静的メソッドもあります。これには、9つのバージョンがあります。よく使いそうなバージョンだけを紹介します。残りは、MSDNなどで調べてみてください。

```
public static string Concat (
```

```
    params string[] values
)
```

文字列配列valuesを連結して一つの文字列にします。

```
public static string Concat (
    string str0,
    string str1
)
```

str0とstr1を結合します。

```
public static string Concat (
    string str0,
    string str1,
    string str2
)
```

str0,str1,str2を結合します。

```
public static string Concat (
    string str0,
    string str1,
    string str2,
    string str3
)
```

str0, ...str3を結合します。

```
public static string Concat (
    Object arg0,
    Object arg1
)
```

arg0とarg1を文字列にしてそれぞれを結合した文字列を作成します。Object型はどのような型でも代入できるスーパー型です。これについても後の章で解説します。C#ではobject型、.NET型ではSystem.Object型です。

```
public static string Concat (
    Object arg0,
    Object arg1,
    Object arg2
)
public static string Concat (
    Object arg0,
    Object arg1,
    Object arg2,
    Object arg3
)
```

arg0, arg1,..をそれぞれ文字列に変換して、これらを結合した文字列を作ります。

次は、引数が1つだけのバージョンです。

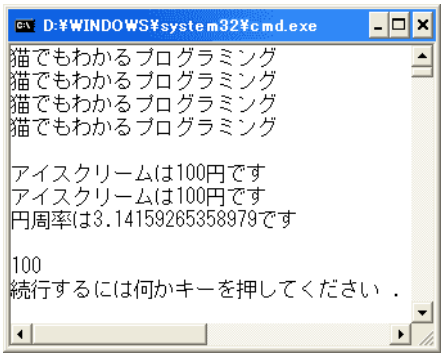
```
public static string Concat (
    Object arg0
)
```

arg0を文字列に変換します。arg0はnull参照でもかまいません。null参照とは何も参照していないという意味です。これについても後の章で解説します。

```
// concat01.cs

using System;

class concat01
{
    public static void Main()
    {
        string[] strarray;
        strarray = new string[4]{"猫", "でも", "わかる", "プログラミング"};

        string str = string.Concat(strarray);
        Console.WriteLine(str);

        string str1 = "猫でもわかる", str2 = "プログラミング";
        str = string.Concat(str1, str2);
        Console.WriteLine(str);
```

```
        string str10 = "猫でも", str20 = "わかる", str30 = "プログラミング";
        str = string.Concat(str10, str20, str30);
        Console.WriteLine(str);

        string str100 = "猫", str200 = "でも", str300 = "わかる",
        str400 = "プログラミング";

        str = string.Concat(str100, str200, str300, str400);
        Console.WriteLine(str);

        Console.WriteLine();
        string stra = "アイスクリームは", strb = "円です";
        int a = 100;
        str = string.Concat(stra, a);
        str = string.Concat(str, strb);
        Console.WriteLine(str);

        str = string.Concat(stra, a, strb);
        Console.WriteLine(str);

        double pai = Math.PI;
        string en = "円周率は", endesu = "です";

        str = string.Concat(en, pai, endesu);
        Console.WriteLine(str);

        string str0 = null;
        str = string.Concat(str0);
        Console.WriteLine(str);

        int x = 100;
        str = string.Concat(x);
        Console.WriteLine(str);

    }
}
```

実行結果は次のようになります。



PIは、Mathクラスの静的フィールドでMath.PIで円周率を表します。(もちろんPIフィールドは、定数で変更はできません。フィールドについてはクラスの章で解説します。プロパティは、フィールドに値を設定したり、フィールドから値を読み出したりするものと考えてください。これについても後に詳述します。)

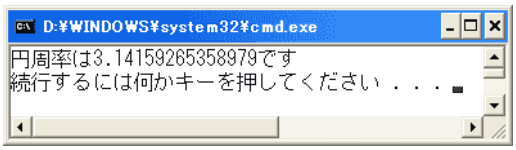
結局Concat静的メソッドは、4つ以内の引数はどんな型でも結合して文字列にしてしまうメソッドです。しかし、

```
string str0 = "アイスクリームは", str1 = "円です";
int x = 100;
str = str0 + x + str;
```

のように、「+」演算子を使っても簡単に実現できます。(しかも直感的でわかりやすいですね。ただ、Cに慣れた人には面白くないかもしれません)

```
// pulus01.cs

using System;

class pulus01
{
    public static void Main()
    {
        string str0 = "円周率は", str1 = "です";
        double pai = Math.PI;

        string str = str0 + pai + str1;
        Console.WriteLine(str);
    }
}
```

実行結果は、当然次のようになります。



[C# Index] [**総合** Index] [Previous Chapter] [Next Chapter]

Update 12/Aug/2006 By Y.Kumei