

# **ЖОЛДЫҚ АЙНЫМАЛЫЛАР**

## Char құрылымы

UTF-16 кодының элементі ретіндегі символ.

Атаулар кеңістігі: [System](#)

C# тілінде символдармен жұмыс жасауға арналған арнайы char класы қолданылады, онда символдарды көрсетудің екі байттық жүйесі және осы символдармен жұмыс жасаудың көптеген әдістер жиыны қолданылады.

IsLower әдісін қолдану мысалы.

```
using System;
public class IsLowerSample {
    public static void Main() {
        char ch = 'a';
        Console.WriteLine(Char.IsLower(ch));

        Console.WriteLine(Char.IsLower("upperCase",
5));
    }
}
// Output: "True"
// Output: "False"
```

## Char.IsWhiteSpace (Char) әдісі

Егер с символы бос орын болса, онда true мәні қайтарылады.

```
using System;public class IsWhiteSpaceSample
{
    public static void Main()
    {
        string str = "black matter";
        Console.WriteLine(Char.IsWhiteSpace('A'));
        Console.WriteLine(Char.IsWhiteSpace(str, 5));
    }
}
// Output: "False"
// Output: "True"
```

Мысал,

`M:System.Char.GetUnicodeCategory(System.Char)` әдісі - string жолындағы әрбір символдың Unicode категориясын шығарады .

```
using System;
using System.Globalization;

class Example
{
    public static void Main()
    {
        // Define a string with a variety of character categories.
        String s = "The car drove down the narrow, secluded road.";
        // Determine the category of each character.
        foreach (var ch in s)
            Console.WriteLine("{0}: {1}", ch, Char.GetUnicodeCategory(ch));
    }
}
```

## Нәтиже:

```
// The example displays the following output:  
// 'T': UppercaseLetter  
// 'h': LowercaseLetter  
// 'e': LowercaseLetter  
// ' ': SpaceSeparator  
// 'r': LowercaseLetter  
// 'e': LowercaseLetter  
// 'd': LowercaseLetter  
// ' ': SpaceSeparator  
// 'c': LowercaseLetter  
// 'a': LowercaseLetter  
// 'r': LowercaseLetter  
// ' ': SpaceSeparator  
// 'd': LowercaseLetter  
// 'r': LowercaseLetter
```

# Жолдық айнымалы ұғымы

## String классы

UTF-16 кодындағы мәтін

Атаулар кеңістігі: System

C# тілінде жолдармен жұмыс істеу үшін (мәтіндермен, символдардың кез келген бірігуі) сәйкес класымен анықталатын арнайы string типі қолданылады. String класы сілтемелік типке жатады және кез келген ұзындықтағы жолдық айнымалыларды анықтауға көмектеседі.

## Конструкторлар (толық ақпаратты анықтамадан қараңыз)

Имя	Описание
<a href="#">String(Char*)</a>	Инициализирует новый экземпляр класса String значением, определенным указателем на массив знаков Юникода.
<a href="#">String(Char*, Int32, Int32)</a>	Инициализирует новый экземпляр класса String значением, которое определяется заданным указателем на массив знаков Юникода, начальной позицией знака в этом массиве и длиной.
<a href="#">String(Char, Int32)</a>	Инициализирует новый экземпляр класса String значением, полученным путем повторения заданного знака Юникода указанное число раз.
<a href="#">String(Char[])</a>	Инициализирует новый экземпляр класса String значением, заданным в виде массива знаков Юникода.
<a href="#">String(Char[], Int32, Int32)</a>	Инициализирует новый экземпляр класса String значением, заданным массивом знаков Юникода, начальной позицией знака в пределах данного массива и длиной.



## Конструкторлар (толық ақпаратты анықтамадан қараңыз)

Имя	Описание
<a href="#">String(SByte*)</a>	Инициализирует новый экземпляр класса String значением, определенным указателем на массив 8-разрядных целых чисел со знаком.
<a href="#">String(SByte*, Int32, Int32)</a>	Инициализирует новый экземпляр класса String значением, определяемым заданным указателем на массив 8-разрядных целых чисел со знаком, начальной позицией в пределах массива и длиной.
<a href="#">String(SByte*, Int32, Int32, Encoding)</a>	Инициализирует новый экземпляр класса String значением, определенным заданным указателем на массив 8-разрядных целых чисел со знаком, начальной позицией в пределах этого массива, длиной и объектом Encoding.

# Конструкторлар мысалдары

```
// string бнициализациясының мысалдары
char[] characterArray =
    { 'b', 'i', 'r', 't', 'h', ' ', 'd', 'a', 'y' } ;
string originalString = "Welcome to C# programming!";
string string1 originalString;
string string2 newstring( characterArray );
string string3 newstring( characterArray, 6, 3 );
string string4 newstring( 'C', 5 );
```

## Нәтижесі:

"Welcome to C# programming!"

"birth day"

"day"

"CCCCC"

## Жолдық айнымалы ұғымы

Мысал,

```
string string1 = "Today is " + DateTime.Now.ToString("D") + ".";
Console.WriteLine(string1);

string string2 = "This is one sentence. " + "This is a second. ";
string2 += "This is a third sentence.";
Console.WriteLine(string2);
// The example displays output like the following:
// Today is Tuesday, July 06, 2011.
// This is one sentence. This is a second. This is a third sentence.
```

## string индекаторы

string индекаторы, жолдағы кез-келген символды таңдап алуға көмектеседі:

```
// символдарды индекстері қолданып кері тіртіпте шығару
Console.Write( "The string reversed is: " );
for ( int i = string1.Length - 1; i >= 0; --i)
Console.Write( string1[i] );
```

## Length қасиеті,

- жолдың ұзындығын қайтарады.

```
Console.WriteLine( "Length of string1: "+ string.Length );
```

## CopyTo әдісі

- символдарды көшіреді.

```
string string1 = "hello there";  
// Копирование символов из string1 в characterArray  
string1.CopyTo( 0, characterArray, 0,  
characterArray.Length );  
Console.WriteLine( "\nThe character array is: " )
```

**Нәтижесі:**

**The character array is: hello**

## @ және \ символдарын қолдану

Жолда символдарды жазылған күйінде шығару үшін @ символы литерал алдында жазылады.

Мына берілген жолды

**"C: \MyFolder\MySubFolder\MyFile. txt"**

келесі түрде жазуға болады:

**1) string file = "C:\\MyFolder\\MySubFolder\\MyFile.txt";**

2) немесе @ символын қолданып, жазуға болады:

**string file = @"C:\MyFolder\MySubFolder\MyFile.txt";**

# Жолдарды салыстыруу

String классы жолдарды салыстырудун бирнеше әдістерін ұсынады. Жолдарды салыстырудун бирнеше әдістері бар: **Equals**, **CompareTo** және **==** операторы

**Equals** әдісін және **==** операторын қолдану мысалы

```
// Проверка равенства с методом Equals
if ( string1.Equals( "hello" ) )
    Console.WriteLine( "string1 equals \"hello\"" );
else
    Console.WriteLine( "string1 does not equal \"hello\"" );

// Проверка равенства с оператором ==
if ( string1 == "hello" )
```

**CompareTo** әдісін қолдану мысалы

```
string1.CompareTo( string2)
```

# StartsWith және EndsWith әдістері

Жолдарда бастапқы немесе соңғы символдарын тексеру - ол үшін **StartsWith** және **EndsWith** әдістері қолданылады.

```
string[] strings = { "started", "starting", "ended", "ending" };  
// Проверяем, начинается ли каждая строка в массиве с "st"  
for ( int i = 0; i < strings.Length; i++)  
{  
if ( strings[ i ].StartsWith( "st" ) )  
Console.WriteLine( "\"" + strings[ i ] + "\"" + " starts with \"st\"");  
  
if ( strings[ i ].EndsWith( "ed" ) )  
Console.WriteLine( "\"" + strings[ i ] + "\"" + " ends with \"ed\"");  
}  
.. . . .
```

**Нәтижесі:**

"started" starts with "st"



## Символдар мен ішкі жолдарды іздеу

- ол үшін IndexOf, IndexOfAny, LastIndexOf және LastIndexOfAny әдістері қолданылады.

```
// 1 Проверка IndexOf для поиска символа в строке  
Console.WriteLine( "First 'c' is located at index " +  
letters.IndexOf( 'c' ) );
```

```
// 2 Проверка LastIndexOf для поиска символа в строке  
Console.WriteLine( "\nlast 'c' is located at index " +  
letters.LastIndexOf( 'c' ) );
```

```
// 3 Проверка IndexOf для поиска подстроки  
Console.WriteLine( "\nFirst \"def\" is located at index " +  
letters.IndexOf( "def" ));
```

```
// 4 Проверка LastIndexOf для поиска подстроки  
Console.WriteLine( "\nlast \"def\" is located at index " +  
letters.LastIndexOf( "def" ));
```

## Ішкі жолдарды қайтару (извлечение подстроки)

- Substring әдісі қолданылады.

```
string letters = "abcdefghijklmabcdefghijklm";
```

```
// Вызов метода Substring с одним параметром
```

```
Console.WriteLine( "Substring from index 20 to end is \"" +  
letters.Substring( 20) + "\"");
```

```
// Вызов метода Substring с двумя параметрами
```

```
Console.WriteLine( "Substring from index 0 of length 6 is \"" +  
letters.Substring( 0, 6) + "\" );
```

.. . . .

**Нәтижесі:**

**Substring from index 20 to end is "hijklm"**

**Substring from index 0 of length 6 is "abcdef"**

## Жолдарды біріктіру (Конкатенация)

+ , - операторлары және Concat әдісі қолданылады.

```
1) Console.WriteLine( "string1 \"" + string1+ "\"\n" + "string2 =  
  \"" + string2 + "\"" );
```

```
2) string.Concat( string1, string2)
```



## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

Мысал. [Rank қасиеті](#) массив өлшемін көрсетеді.

```
class TestArraysClass
{ static void Main()
  {
    // Declare and initialize an array:
    int[,] theArray = new int[5, 10];
    System.Console.WriteLine("The array has {0} dimensions.",
theArray.Rank);
  }
}
// Output: The array has 2 dimensions.
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

Мысал, 4 жолдан, 2 бағанадан тұратын кі өлшемді массивті құру.

```
int[,] array = new int[4, 2];
```

Следующее объявление создает трехмерный массив с количеством элементов 4, 2 и 3.

```
int[ , , ] array1 = new int[4, 2, 3];
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

Инициализация массива.

1) Массив можно инициализировать при объявлении, как показано в следующем примере.

```
// Two-dimensional array.
```

```
int[,] array2D = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, {  
7, 8 } };
```

```
// The same array with dimensions specified.
```

```
int[,] array2Da = new int[4, 2] { { 1, 2 }, { 3, 4 }, { 5, 6 },  
{ 7, 8 } };
```

```
// A similar array with string elements.
```

```
string[,] array2Db = new string[3, 2] { { "one", "two" }, {  
"three", "four" }, { "five", "six" } };
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

```
// Accessing array elements.  
System.Console.WriteLine(array2D[0, 0]);  
System.Console.WriteLine(array2D[0, 1]);  
System.Console.WriteLine(array2D[1, 0]);  
System.Console.WriteLine(array2D[1, 1]);  
System.Console.WriteLine(array2D[3, 0]);  
System.Console.WriteLine(array2Db[1, 0]);
```



## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

2) Можно также инициализировать массив, не указывая его размерность.

```
int[,] array4 = { { 1, 2 }, { 3, 4 }, { 5, 6 },  
{ 7, 8 } };
```

3) Если нужно создать переменную массива без инициализации, то необходимо использовать оператор **new**, чтобы присвоить массив переменной. В следующем примере показано использование **new**.

```
int[,] array5; array5 = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6  
{ 7, 8 } }; // ОК - дұрыс
```

```
//array5 = {{1,2}, {3,4}, {5,6}, {7,8}}; // Error - қате
```

## КӨПӨЛШЕМДІ МАССИВТЕР

### Екі өлшемді массивтер

Следующий пример присваивает значение конкретному элементу массива.

```
array5[2, 1] = 25;
```

4) В следующем примере кода элементы массивов инициализируются значениями по умолчанию (за исключением массивов массивов).

```
int[,] array6 = new int[10, 10];
```