# Console

## System.Console

|  |  |  |
| --- | --- | --- |
| **API** | **Description** | **Return values** |
| **void WriteLine**(**string** *inValue*)  Overloads:   1. (**object** *inValue*) | Write *inValue*, following by a line terminator, to the standard output stream. |  |
| **string ReadLine**() | Read a line of characters from the standard input stream. |  |

# Data Types

## string (alias of System.String)

|  |  |  |
| --- | --- | --- |
| **API** | **Description** | **Return values** |
| **bool** **Equals**(  **string** *inValue*) | Compare the *inValue* with the specified string. | true: string == inValue  false: string != inValue |
| **bool** **IsNullOrEmpty**(  **string** *inValue*) | Check whether the *inValue* is null or empty. | true: null or empty string  false: otherwise |
| **string** **Format**(  **string** *inFormat*,  **object** *inAr0g*)  More overloads, more uses: [here](https://www.c-sharpcorner.com/UploadFile/mahesh/format-string-in-C-Sharp/). | Replace one or more format items in *inFormat* with the string representation of the *inArg*.  Note: Format items are denoted as “{0}, {1}, …”  Ex:  *string format = @"{0}\test.ini};*  *string arg0 = @"Preferences";*  *string output = string.Format(format, arg0);*  // *output* will be *@"Preferences\test.ini"* |  |
| **string** **ToLower**(  **string** *inValue*) | Convert the *inValue* to lowercase. |  |
| **string** **ToUpper**(  **string** *inValue*) | Convert the *inValue* to uppercase. |  |
| **bool** **Contains**(  **string** *inValue*) | Check whether the *inValue* is a substring of the specified string. |  |

## Array

|  |  |  |
| --- | --- | --- |
| **API** | **Description** | **Return values** |
| **int** **Length**() | Get total of elements in all dimensions of the array. |  |
|  |  |  |
|  |  |  |
|  |  |  |

# IO

## System.IO.Directory

|  |  |  |
| --- | --- | --- |
| **API** | **Description** | **Return values** |
| **bool** **Exists**(  **string** *inDirPath*) | Check whether *inDirPath* exists. | true: exist  false: otherwise |
| **string[]** **GetFiles**(  **string** *inDirPath*) | Get list of file names (including file paths) in the *inDir*. |  |
| **DirectoryInfo CreateDirectory**(  **string** *inPath*) | Create a new directory unless the directory in *inPath* already exists. |  |

## System.IO.DirectoryInfo

|  |  |  |
| --- | --- | --- |
| **API** | **Description** | **Return values** |
| **void** **DirectoryInfo**(  **string** *inPath*) | Get info of the directory specified in *inPath* | void |
| **FileInfo[]** **GetFiles**(  **string** *inSearchPattern*) | Get a list of files from the current dir matching *inSearchPattern (eg. “\*.zip”)* | File(s) exist: Array of file(s)  File not exist: Empty array of files |
| **DirectoryInfo[]** **GetDirectories**(  **string** *inSearchPattern*,  **SearchOption** *inSearchOption*) | Get a list of sub-directories in the current dir matching *inSearchPattern* and *inSearchOption*.  Note: For all sub-direction, *inSearchPattern = “\*”* |  |

## System.IO.File

|  |  |  |
| --- | --- | --- |
| **API** | **Description** | **Return values** |
| **bool** **Exists**(  **string** *inFilePath*) | Check whether *inFilePath* exists. | true: exist  false: otherwise |
| **void** **Copy**(  **string** *srcFileName*,  **string** *destFileName*,  **bool** *is*O*verride*) | Copy an existing file (*srcFileName*) to a new file (*destFileName*). |  |

## System.IO.FileInfo

|  |  |  |
| --- | --- | --- |
| **API** | **Description** | **Return values** |
| **string** **Name**() | Get name of the file. |  |

## System.IO.Path

|  |  |  |
| --- | --- | --- |
| **API** | **Description** | **Return values** |
| **string GetDirectoryName**(  **string** *inFilePath*) | Get directory path from file path.  Ex: If *inFilePath = "D:\Test\test.exe"*, then the API’s return value is *"D:\Test*" |  |
| **string** **GetFileNameWithoutExtension**(  **string** *inFilePath*) | Get name of the file, without the file extension. |  |
| **string** **Combine**(  **string** *inPath1*,  **string** *inPath2*) | Combine two strings into a path. |  |

# List

## System.Collections.Generic.List

# Time

## System.DateTime

Represent an instant in time, expressed as date and time.

Eg:

var date = **DateTime**.**Now**;

Console.WriteLine(date);

Console.WriteLine($"On {date:d} at {date:t}!");

Output:

6/29/2019 8:39:37 PM

On 6/29/2019 at 8:39 PM!

# Globalization

## System.Globalization.CultureInfo

Provides information about a specific culture (called a locale for unmanaged code development), including name, language, writing system, calendar, sort order of strings, formatting for dates and numbers, etc.

// Detect system locale

string langFolder = "English";

CultureInfo ci = CultureInfo.CurrentCulture;

if (ci.Name.Equals("ja-JP"))

{

langFolder = "Japanese";

}