Online Training
C# ASP.NET Online Training Program
C# ASP.NET Core Training
Microservices Online Training Program
Microsoft Azure Training

Introduction & Environment Setup
Intro
Setup Computer Works
C# Introduction to Programming Languages
Free Computer Programs Utilities
Different Types of Applications
Programming Methodologies
Algorithm, Flowcharts, Programs, and Flowcharts
Introduction to .NET Framework
.NET Framework Architecture and Components
Introduction to C#
Introduction to Programming
How to Download and Install Visual Studio on Windows
Creating First Console Application using Visual Studio
.NET Developer Roadmap for 2025
Coding Standard Best Practices

C# .NET Basics
Basic Structure of C# Programs
Methods and Properties of Console Class in C#
Data Types in C#
Literals in C#
Type Casting C#
Variables in C#
Operators in C#
Control Flow Statements in C#
If Else Statements in C#
Switch Statements in C#
Loops in C#
While Loop in C#
Do While Loop in C#
For Loop in C#
Break Statement in C#
Continue Statement in C#
Goto Statement in C#
Jump Statements in C#
Array Data Structure in C#
Call By Value and Call By Reference in C#
Recursion in C#
Command Line Arguments in C#
Command Line Arguments in C#
String in C#
Static Keyword in C#
Static Class and Static Members in C#
Const and Read Only in C#
Properties in C#
Why we Should Override ToString Method in C#
Difference Equals Method in C# and Equality Operator
Object Initializers and Collection Initializers
Tolstring Method in C#
Checked and Unchecked Keywords in C#
Boxing and Unboxing in C#

OOPs in C#
Object Oriented Programming (OOPs) in C#
Class and Objects in C#
Constructor in C#
Types of Constructors in C#
Why We Need Constructors in C#
Static vs Non-Static Constructors
Private Constructors in C#
Destructors in C#
Garbage Collection in .NET Framework
Differences Between Finalize and Dispose in C#
Inheritance in C#
Encapsulation in C#
Abstraction in C#
Interfaces in C#
Types of Inheritance in C#
Interface Inheritance
Application Development
Call and Hook Relationship in C#
Generalization and Specialization in C#
Method Class and Method Methods in C#
Method Class and Method Methods Interview Questions and Answers
How to Use Method Classes and Methods in C# Application
Interface in C#
Interface Interview Questions and Answers
Interface Runtime Examples in C#
Multiple Inheritance in C#
Multiple Interface Runtime Examples in C#
Polymorphism in C#
Method Overloading in C#
Operator Overloading in C#
Method Overriding in C#
Method Hiding in C#
Sealed Class and Sealed Methods in C#
Sealed Class and Sealed Methods in C#
Extension Methods in C#
Partial Class in C#
Variable Methods and Instance of a Class in C#

Advanced Real Time Examples
Real time Examples on Encapsulation Principle in C#
Real Time Examples of Abstraction Principle in C#
Real Time Examples of Inheritance Principle in C#
Real Time Examples of Polymorphism Principle in C#
Real Time Examples of Interface in C#
Real Time Examples of Method Class in C#

Exception Handling
C# Exception Handling in C#
Multiple Catch Blocks in C#
Finally Block in C#
How to Create Custom Exceptions in C#
Throw Exception in C#
Exception Handling Abuse in C#

Events, Delegates and Lambda Expressions in C#
Course Structure of Events, Delegates and Lambda Expressions
Basics of Events, Delegates and Event Handler in C#
Delegates in C#
Multicast Delegates in C#
Delegates Real Time Example in C#
Generic Delegates in C#
Anonymous Method in C#
Lambda Expressions in C#
Events in C# with Examples

Multi-Threading
Multithreading in C#
Thread Safety in C#
How to Pass Data to Thread Function in C# Type Safe Method in C#
How to Retrieve Data from a Thread Function in C#
Join Method and Isalive Property of Thread Class in C#
Thread Synchronization in C#
Lock in C#
Monitor Class in C#
Mutex Class in C#
Semaphore Class in C#
SemaphoreSlim Class in C#
EventWaitHandle in C#
Performance Testing of a Multi-threaded Application
Thread Pooling in C#
Foreground and Background Threads in C#
AutoResetEvent and ManualResetEvent in C#
Thread.Join in C#
Thread.Sleep in C#
How to Terminate a Thread in C#
Inter Thread Communication in C#
How to Debug a Multi-threaded Application in C#

Collections in C#
Array in C#
2D Array in C#
Advantages and Disadvantages of Array in C# Type
Jagged Array in C#
Hashtable in C#
Non-Generic Stack in C#
Non-Generic Queue in C#
Non-Generic SortedList in C#
Advantages and Disadvantages of Non-Generic Collection in C#
Generic Collection in C#
Generic Collections in C#
Generics in C#
Generic Constructors in C#
Generic Interfaces in C#
How to Sort a List of Complex Type in C#
Comparison Delegates in C#
Dictionary Collection Class in C#
Comparision Between Array List and Dictionary in C#
List in Dictionary in C#
Generic Stack Collection Class in C#
Generic Queue Collection Class in C#
Generic HashSet Collection Class in C#
Generic SortedSet Collection Class in C#
Generic SortedDictionary Collection Class in C#
Generic ReadOnlyCollection Class in C#
Concurrent Collection in C#
ConcurrentDictionary Collection Class in C#
ConcurrentBag Collection Class in C#
ConcurrentStack Collection Class in C#
ConcurrentQueue Collection Class in C#
BlockingCollection in C#
Expect and Implicit Excel Data Type in C#

File Handling
C# Handling I/O
FileStream Class in C#
StreamReader and StreamWriter in C#
File Class in C#
TextWriter and TextReader in C#
BinaryWriter and BinaryReader in C#
StringWriter and StringReader in C#
Directory Class in C#
DirectoryInfo Class in C#
Expect and Implicit Excel Data in C#

Asynchronous Programming
Introduction to Concurrency Async and Await in C#
Task Class in C#
How to Return a Value from Task in C#
How to Execute Multiple Tasks in C#
How to Limit Number of Concurrent Tasks in C#
How to Cancel a Task in C# using Cancellation Token
How to Create Synchronous Method using Task in C#
Retry Pattern in C#
Task Continuation in C#
How to Control the Result of a Task in C#
Task-Based Asynchronous Programming in C#
Asynchronous Programming Continuation Tasks
How to Attached Child Tasks to a Parent Task in C#
TaskFlow in C#
Task Cancellation Cancellation Token in C#
Asynchronous Streams in C#
How to Cancel Asynchronous Stream in C#

Parallel Programming
Task Parallel Library in C#
Parallel for in C#
Parallel ForEach Loop in C#
Parallel Invoke
Maximum Degree of Parallelism in C#
Parallel Exception Operations in C#
Parallel Loop Break Safety and Stop Conditions in C#
Interrlocked in Lock in C#
Parallel LINQ in C#
Cancellation Asynchronous Programming
Parallel Programming in C#

AutoMapper
AutoMapper in C#
AutoMapper Complex Mapping in C#
AutoMapper Different Type to Member Mapping
AutoMapper in C#
AutoMapper Reverse Mapping in C#
AutoMapper Conditional Mapping in C#
AutoMapper Ignore Method in C#
Fixed and Dynamic Values in Destination Property in AutoMapper

Optional Parameter, Indexers and Enums
How to Create Optional Parameters in C#
Indexers in C#
Indexers Real Time Example in C#
Enums in C#

.NET Framework Architecture
Introduction to .NET Framework
.NET Framework Language Runtime
.NET Framework
.NET Program Execution Process
Intermediate Language (IL)/MSIL & JIT
Assembly
Common Type System in .NET Framework
Common Language Specification in .NET Framework
Managed and Unmanaged Code in .NET Framework
Assembly Dll, EXE in .NET Framework
Namespaces in .NET Framework
Strong and Weak Assemblies in .NET Framework
How to Install an Assembly in the GAC in .NET Framework
DLL Hell Problem and Solution in .NET Framework

Var, Dynamic and Reflection
Reflection in C#
Dynamic Type in C#
Var Keyword in C#
Var Type in C#
Operator vs Method Hiding in C#
Variable Keyword in C#
Var vs Dyn in C#
Named Parameters in C#

C# 3.0 new Features
C# 3 New Features
Extension Methods in C#
Extensionmethod in C# Type Inference in C#
Object Initializers in C#
Type Inference in C#
Splitting Types in C# 3
Local Functions in C#
Def Returns and Ref Locals in C#
Generalized Async Return Types in C# 7
Expression Bodied Members in C#
Throwex Expression in C#
Async Main in C#

C# 8 New Features
Default Interface Methods in C#
Default Interface Methods in C#
Pattern Matching in C#
Using Declarations in C#
Static Local Functions in C#
Disposable Ref Structs in C#
Readonly Members Type in C#
Indices and Ranges in C#
Null Coalescing Assignment Operator in C#
Unmanaged Constructed Types in C#
Stackalloc in C#

Most Popular C# Books
Pro C# 10 with .NET Framework Book
Most Recommended Data Structure and Algorithms Books using C#

Conversion Between Array List and Dictionary in C#

Back to: C#.NET Tutorials For Beginners and Professionals

Conversion between Array, List, and Dictionary in C#

In this article, we will discuss how to perform **Conversion Between Array List and Dictionary in C#**. Please read our previous article where we discussed Dictionary in C# with examples. As part of the article, we will discuss the following topics.

1. Convert an array to a List – Use **ToList()** method
2. Convert a list to an array – Use **ToArray()** method
3. Convert a List to a Dictionary – Use **ToDictionary()** method
4. Convert an array to a Dictionary – Use **ToDictionary()** method
5. Convert a Dictionary to a List – Use **ToValues()** method or the **Values** Property of the dictionary object
6. Convert a Dictionary to an Array – Use the **ToArray()** method on the **Values** Property of the dictionary object

Let us understand this with an example.

The code is self-explained. Please go through the comments.

```
using System;
using System.Linq;
using System.Collections;
namespace ArrayToDictionaryExample
{
    public class Program
    {
        public static void Main(string[] args)
        {
            //Create Employee objects
            Employee emp1 = new Employee();
            emp1.ID = 101;
            emp1.Name = "Pranaya";
            emp1.Gender = "Male";
            emp1.Salary = 30000;

            Employee emp2 = new Employee();
            emp2.ID = 102;
            emp2.Name = "Priyanka";
            emp2.Gender = "Female";
            emp2.Salary = 40000;

            Employee emp3 = new Employee();
            emp3.ID = 103;
            emp3.Name = "Tripathi";
            emp3.Gender = "Male";
            emp3.Salary = 50000;

            // Create an array of employees with size 3
            // and then insert the employees into the array
            Employee[] arrayOfEmployees = new Employee[3];
            arrayOfEmployees[0] = emp1;
            arrayOfEmployees[1] = emp2;
            arrayOfEmployees[2] = emp3;

            // To convert an array to a list use ToList() method
            Console.WriteLine("Convert an array to a list using ToList() method");
            ArrayOfEmployees listOfEmployees = arrayOfEmployees.ToList();
            foreach (Employee emp in listOfEmployees)
            {
                Console.WriteLine("ID = {0}, Name = {1}, Gender =[0], Salary = {3}",
                    emp.ID, emp.Name, emp.Gender, emp.Salary);
            }
            Console.WriteLine();

            // To convert a list to an array use ToArray() method
            ListEmployee[] arrayOfEmployees = listOfEmployees.ToArray();
            foreach (Employee emp in arrayOfEmployees)
            {
                Console.WriteLine("ID = {0}, Name = {1}, Gender =[0], Salary = {3}",
                    emp.ID, emp.Name, emp.Gender, emp.Salary);
            }
            Console.WriteLine();

            // To convert a list to a Dictionary use ToDictionary() method
            Dictionary<int, Employee> dictionaryEmployeesFromArray = listOfEmployees.ToDictionary(emp => emp.ID);
            foreach (Employee emp in arrayOfEmployees)
            {
                Console.WriteLine("ID = {0}, Name = {1}, Gender =[0], Salary = {3}",
                    emp.ID, emp.Name, emp.Gender, emp.Salary);
            }
            Console.WriteLine();

            // To convert an array to a Dictionary use ToDictionary() method
            Dictionary<int, Employee> dictionaryEmployeesFromArray = arrayOfEmployees.ToDictionary(emp => emp.ID);
            foreach (Employee emp in arrayOfEmployees)
            {
                Console.WriteLine("ID = {0}, Name = {1}, Gender =[0], Salary = {3}",
                    emp.ID, emp.Name, emp.Gender, emp.Salary);
            }
            Console.WriteLine();

            // To convert a Dictionary to a List, use ToList method on the Values
            // property of the dictionary
            List<Employee> listOfEmployeesFromDictionary =
dictionaryEmployeesFromArray.Values.ToList();
            foreach (Employee emp in listOfEmployeesFromDictionary)
            {
                Console.WriteLine("ID = {0}, Name = {1}, Gender =[0], Salary = {3}",
                    emp.ID, emp.Name, emp.Gender, emp.Salary);
            }

            Console.ReadKey();
        }
    }
    class Employee
    {
        public int ID = 0;
        public string Name = null;
        public string Gender = null;
        public int Salary = 0;
    }
}
```

Output:

```
Convert an array to a list using ToList() method
ID = 101, Name = Pranaya, Gender =Male, Salary = 30000
ID = 102, Name = Priyanka, Gender =Female, Salary = 40000
ID = 103, Name = Tripathi, Gender =Male, Salary = 50000

ID = 101, Name = Pranaya, Gender =Male, Salary = 30000
ID = 102, Name = Priyanka, Gender =Female, Salary = 40000
ID = 103, Name = Tripathi, Gender =Male, Salary = 50000

ID = 101, Name = Pranaya, Gender =Male, Salary = 30000
ID = 102, Name = Priyanka, Gender =Female, Salary = 40000
ID = 103, Name = Tripathi, Gender =Male, Salary = 50000

ID = 101, Name = Pranaya, Gender =Male, Salary = 30000
ID = 102, Name = Priyanka, Gender =Female, Salary = 40000
ID = 103, Name = Tripathi, Gender =Male, Salary = 50000

ID = 101, Name = Pranaya, Gender =Male, Salary = 30000
ID = 102, Name = Priyanka, Gender =Female, Salary = 40000
ID = 103, Name = Tripathi, Gender =Male, Salary = 50000

ID = 101, Name = Pranaya, Gender =Male, Salary = 30000
ID = 102, Name = Priyanka, Gender =Female, Salary = 40000
ID = 103, Name = Tripathi, Gender =Male, Salary = 50000
```

In the next article, I am going to discuss List in Dictionary in C# with examples. Here, in this article, I try to explain the **Conversion Between Array List and Dictionary in C#** with an example. I hope this article will help you with your need. I would like to have your feedback, your valuable feedback, question, or comments about this article.

DOT NET TUTORIALS
Dot Net Tutorials
About the Author: Pranaya Rout
Pranaya Rout has published more than 3100 articles in 22 year career. Pranaya Rout has very good experience with Microsoft Technologies, including C#, VB, ASP.NET MVC, ASP.NET Web API, EF, EF Core, ADO.NET, LINQ, SQL Server, MYSQL, Entertainment, ASP.NET Core, Cloud Computing, Microservices, Design Patterns and still learning new things.

Previous Lesson
Dictionary Collection Class in C#

Next Lesson
List in Dictionary in C#

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name*
Email*
Website

Post Comment