

# C# foreach

## C# foreach

---

foreach is used to apply a set of statements for each element in an array or collection.

### C# foreach on String Array

In the following example, we use foreach to print the modified string to console for each element in the array.

#### Program.cs

```
using System;
using System.Collections.Generic;

namespace CSharpExamples {

    class Program {
        static void Main(string[] args) {
            string[] names = {"Ducati", "Honda", "Royal Enfield"};
            foreach(var name in names){
                Console.WriteLine("Buy "+name);
            }
        }
    }
}
```

#### Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
Buy Ducati
Buy Honda
Buy Royal Enfield
```

### C# foreach on List items

Following examples demonstrates the usage of foreach loop on List elements. We just print them. You can do much more on each element of the list based on your use-case requirements.

#### Program.cs

```
using System;
```

```

using System;
using System.Collections.Generic;

namespace CSharpExamples {

    class Program {
        static void Main(string[] args) {
            var names = new List<string>() {"Ducati", "Honda", "Royal Enfield"};
            foreach(var name in names){
                Console.WriteLine(name);
            }
        }
    }
}

```

### Output

```

PS D:\workspace\csharp\HelloWorld> dotnet run
Ducati
Honda
Royal Enfield

```

## C# foreach – Dictionary

You can use foreach to access each of the item in Dictionary. In the following example, we access each item of the dictionary, and then get the key and values separately.

### Programcs

```

using System;
using System.Collections.Generic;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            Dictionary<int, string> dict1 = new Dictionary<int, string>(){
                {1, "Tesla"},
                {2, "Honda"},
                {3, "Toyota"}
            };

            foreach(KeyValuePair<int, string> item in dict1) {
                Console.WriteLine(item.Key+" - "+item.Value);
            }
        }
    }
}

```

### Output

```

PS D:\workspace\csharp\HelloWorld> dotnet run
1 - Tesla
2 - Honda
3 - Toyota

```

---

## Conclusion

---

In this [C# Tutorial](#), we learned the syntax and usage of `foreach()` function, and to iterate over array, dictionary and list items using `foreach()` with example programs.

### C# Tutorial

- ◆ [C# Tutorial](#)
- ◆ [C# Basic Example](#)
- ◆ [C# Comments](#)
- ◆ [C# Variables](#)
- ◆ [C# Constants](#)
- ◆ [C# if, if-else](#)
- ◆ [C# switch](#)
- ◆ [C# while loop](#)
- ◆ [C# for loop](#)
- ⇒ **[C# foreach](#)**
- ◆ [C# break](#)
- ◆ [C# continue](#)
- ◆ [C# struct](#)
- ◆ [C# enum](#)
- ◆ [C# String](#)
- ◆ [C# Array](#)
- ◆ [C# Command Line Arguments](#)

### C# Console Operations

- ◆ [C# Write to Console](#)
- ◆ [C# Read from Console](#)

### C# Object Oriented Programming Concepts

- ◆ [C# Class & Constructors](#)
- ◆ [C# Encapsulation](#)

◆ C# Polymorphism

◆ C# Method Overloading

◆ C# Interfaces

## C# String Operations

◆ C# String Length

◆ C# Substring

## C# File Operations

◆ C# Read Text File

◆ C# Write to File

◆ C# Delete File

◆ C# Copy File

## C# Exception Handling

◆ C# try-catch

◆ C# finally

◆ C# throw

◆ C# Custom Exception

◆ C# SystemException

◆ C# DivideByZeroException

◆ C# NullReferenceException

◆ C# InvalidCastException

◆ C# IOException

◆ C# FieldAccessException

## C# Collections

◆ C# List

◆ C# SortedList

◆ C# HashSet

◆ C# SortedSet

◆ C# Stack

◆ C# Queue

◆ C# LinkedList

◆ C# LinkedList

◆ C# Dictionary

◆ C# SortedDictionary

### **C# Errors [Solved]**

◆ C# Error: Class does not contain a constructor that takes arguments