Tuple vs ValueTuple in C#

A Tuple or ValueTuple is a way to group multiple values into a single object without creating a custom class.

# What is a Tuple in C#?

A Tuple is a data structure introduced in .NET Framework 4.0. It is a reference type, stored on the heap, and is immutable. Values are accessed using Item1, Item2, etc.

# What is a ValueTuple in C#?

A ValueTuple is a newer, more efficient version of Tuple introduced in C# 7.0. It is a value type (stored on stack), supports named fields, and offers better performance and readability.

# Use of Tuple and ValueTuple

Use Tuples or ValueTuples to return multiple values from a method, or temporarily group values without creating a custom class. Prefer ValueTuple for most modern C# code.

# Comparison

Feature Tuple ValueTuple  
Type Reference Value  
Performance Slower Faster  
Syntax Tuple.Create("A", 1) ("A", 1)  
Access .Item1, .Item2 .Name, .ID  
Mutability Immutable Mutable  
Readability Less readable More readable

# Example: Tuple

using System;  
  
class Program  
{  
 static void Main()  
 {  
 Tuple<string, int> employee = Tuple.Create("John", 101);  
 Console.WriteLine("Name: " + employee.Item1);  
 Console.WriteLine("ID: " + employee.Item2);  
 }  
}

# Example: ValueTuple

using System;  
  
class Program  
{  
 static void Main()  
 {  
 (string Name, int ID) employee = ("Alice", 102);  
 Console.WriteLine("Name: " + employee.Name);  
 Console.WriteLine("ID: " + employee.ID);  
 }  
}

# Example: Returning ValueTuple from Method

(string Name, int ID) GetEmployee()  
{  
 return ("Bob", 103);  
}  
  
void Main()  
{  
 var employee = GetEmployee();  
 Console.WriteLine($"Name: {employee.Name}, ID: {employee.ID}");  
}