Difference between Hashtable and Dictionary in C#

geeksforgeeks.org/difference-between-hashtable-and-dictionary-in-c-sharp

February 25, 2019

In C#, Dictionary is a generic collection which is generally used to store key/value pairs. Dictionary is defined under *System.Collection.Generics* namespace. It is dynamic in nature means the size of the dictionary is growing according to the need.

Example:

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections.Generic;
class GFG {
static public void Main()
{
Dictionary< string , string > My_dict =
new Dictionary< string , string >();
My_dict.Add( "a.01" , "C" );
My_dict.Add( "a.02" , "C++" );
My_dict.Add( "a.03" , "C#" );
foreach (KeyValuePair< string , string > element in My_dict)
{
Console.WriteLine( "Key:- {0} and Value:- {1}" ,
element.Key, element.Value);
}
}
}
```

Output:

```
Key:- a.01 and Value:- C
Key:- a.02 and Value:- C++
Key:- a.03 and Value:- C#
```

A Hashtable is a collection of key/value pairs that are arranged based on the hash code of the key. Or in other words, a Hashtable is used to create a collection which uses a hash table for storage. It is the non-generic type of collection which is defined in *System.Collections* namespace. In Hashtable, key objects must be immutable as long as

they are used as keys in the Hashtable.

Example:

filter_none
edit

play_arrow

brightness_4

```
using System;
using System.Collections;
class GFG {
static public void Main()
{
Hashtable my hashtable = new Hashtable();
mv hashtable.Add( "A1" . "Welcome" );
mv hashtable.Add( "A2" . "to" ):
my_hashtable.Add( "A3" , "GeeksforGeeks" );
foreach (DictionaryEntry element in my_hashtable)
{
Console.WriteLine( "Kev:- {0} and Value:- {1} " ,
element.Key, element.Value);
}
}
}
```

Output:

Key:- A3 and Value:- GeeksforGeeks
Key:- A2 and Value:- to
Key:- A1 and Value:- Welcome

Hashtable Vs Dictionary

Hashtable	Dictionary
A Hashtable is a non-generic collection.	A Dictionary is a generic collection.
Hashtable is defined under System.Collections namespace.	Dictionary is defined under System.Collections.Generic namespace.
In Hashtable, you can store key/value pairs of the same type or of the different type.	In Dictionary, you can store key/value pairs of same type.

In Hashtable, there is no need to specify the type of the key and value.	In Dictionary, you must specify the type of key and value.
The data retrieval is slower than Dictionary due to boxing/ unboxing.	The data retrieval is faster than Hashtable due to no boxing/ unboxing.
In Hashtable, if you try to access a key that doesn't present in the given Hashtable, then it will give null values.	In Dictionary, if you try to access a key that doesn't present in the given Dictionary, then it will give error.
It is thread safe.	It is also thread safe but only for public static members.
It doesn't maintain the order of stored values.	It always maintain the order of stored values.

Recommended Posts:

ankita_saini
Check out this Author's contributed articles.

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.