

C # Hashtable con ejemplos



¿Qué es Hashtable en C #?

Una tabla hash es una colección especial que se utiliza para almacenar elementos clave-valor. Entonces, en lugar de almacenar solo un valor como la pila, la lista de matrices y la cola, la tabla hash almacena 2 valores. Estos 2 valores forman un elemento de la tabla hash.

A continuación se muestran algunos ejemplos de cómo se verían los valores de una tabla hash.

```
{"001", ".Net"}
{"002", ".C #"}
{"003", "ASP.Net"}
```

Arriba tenemos 3 pares clave-valor. Las claves de cada elemento son 001, 002 y 003 respectivamente. Los valores de cada par clave-valor son ".Net", "C #" y "ASP.Net" respectivamente.

Veamos las operaciones disponibles para la colección Hashtable con más detalle.

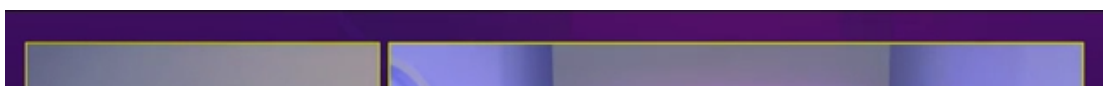
Declaración de Hashtable

La declaración de una tabla hash se muestra a continuación. Una tabla hash se crea con la ayuda del tipo de datos Hashtable. La palabra clave "nueva" se utiliza para crear un objeto de Hashtable. A continuación, el objeto se asigna a la variable ht.

```
Hashtable ht = new Hashtable ()
```

Agregar elementos a la tabla hash

El método Add se usa para agregar un elemento a la cola. La sintaxis general de la declaración se da a continuación.





REPRODUCCI...



```
HashTable.add ("clave", "valor")
```

Ejemplo 1:

Recuerde que cada elemento de la tabla hash consta de 2 valores, uno es la clave y el otro es el valor.

Ahora, veamos cómo funciona esto a nivel de código. Todo el código mencionado a continuación se escribirá en nuestra aplicación de consola.

El código se escribirá en nuestro archivo Program.cs. En el siguiente programa, escribiremos el código para ver cómo podemos usar los métodos mencionados anteriormente.

Por ahora, en nuestro ejemplo, solo veremos cómo podemos crear una tabla hash, agregar elementos a la tabla hash y mostrarlos en consecuencia.

```

{
    static void Main(string[] args)
    {
        Hashtable ht = new Hashtable();
        ht.Add("001", ".Net");
        ht.Add("002", "C#");
        ht.Add("003", "ASP.Net");
        ICollection keys = ht.Keys;
        foreach (String k in keys){
            Console.WriteLine(ht[k]);
        }
        Console.ReadKey();
    }
}

```

Creating a hashtable variable

Getting the keys collection

Adding elements to the hashtable

Displaying the value for each key

(./images/c-sharp-net/052616_1306_CCollection13.png)



```

usando el sistema;
usando System.Collections;
utilizando System.Collections.Generic;
utilizando System.Linq;
usando System.Text;
usando System.Threading.Tasks;

DemoApplication de espacio de nombres
{
    programa de clase
    {
        static void Main (cadena [] argumentos)
        {
            Hashtable ht = new Hashtable ();
            ht.Add ("001", ". Neto");
            ht.Add ("002", "C #");
            ht.Add ("003", "ASP.Net");

            ICollection keys = ht.Keys;

            foreach (Cadena k en claves)
            {
                Console.WriteLine (ht [k]);
            }
            Console.ReadKey ();
        }
    }
}

```

Explicación del código: -

1. Primero, declaramos la variable hashtable usando el tipo de datos Hashtable usando la palabra clave "New". El nombre de la variable que define es 'ht'.
2. Luego agregamos elementos a la tabla hash usando el método Add. Recuerde que necesitamos agregar un elemento clave y un valor cuando agregamos algo a la tabla hash.
3. No existe una forma directa de mostrar los elementos de una tabla hash.
 - Para mostrar la tabla hash, primero necesitamos obtener la lista de claves (001, 002 y 003) de la tabla hash.
 - Esto se hace a través de la interfaz ICollection. Este es un tipo de datos especial que se puede utilizar para almacenar las claves de una colección de tablas hash. Luego asignamos las claves de la colección de tablas hash a la variable 'claves'.
4. A continuación, para cada valor de clave, obtenemos el valor asociado en la tabla hash mediante el uso de la declaración ht [k].

Si el código anterior se ingresa correctamente y el programa se ejecuta, se mostrará el siguiente resultado.

Salida:



Veamos algunos métodos más disponibles para tablas hash.

Contiene clave

Este método se usa para ver si hay una clave presente en Hashtable. A continuación se muestra la sintaxis general de esta declaración. La declaración devolverá verdadero si la clave existe, de lo contrario devolverá el valor falso.

```
Hashtable.Containskey (clave)
```

Contiene valor

Este método se usa para ver si un valor está presente en la tabla hash. A continuación se muestra la sintaxis general de esta declaración. La declaración devolverá verdadero si el valor existe, de lo contrario devolverá el valor falso.

```
Hashtable.ContainsValue (valor)
```

Ejemplo 2:

Cambiamos el código en nuestra aplicación de consola para mostrar cómo podemos usar el método "Containskey" y "ContainsValue".

```

static void Main(string[] args)
{
    Hashtable ht = new Hashtable();

    ht.Add("001", ".Net");
    ht.Add("002", "C#");
    ht.Add("003", "ASP.Net");

    Console.WriteLine(ht.ContainsKey("001"));
    Console.WriteLine(ht.ContainsValue("C#"));
    Console.ReadKey();
}

```

Searching for a key

1

Searching for a value

2

(/images/c-sharp-net/052616_1306_CCollection15.png)

```

usando el sistema;
usando System.Collections;
utilizando System.Collections.Generic;
utilizando System.Linq;
usando System.Text;
usando System.Threading.Tasks;

```

```

DemoApplication de espacio de nombres
{
    programa de clase
    {
        static void Main (cadena [] argumentos)
        {
            Hashtable ht = new Hashtable ();
            ht.Add ("001", ". Neto");
            ht.Add ("002", "C #");
            ht.Add ("003", "ASP.Net");

            Console.WriteLine (ht.ContainsKey ("001"));
            Console.WriteLine (ht.ContainsValue ("C #"));
            Console.ReadKey ();
        }
    }
}

```

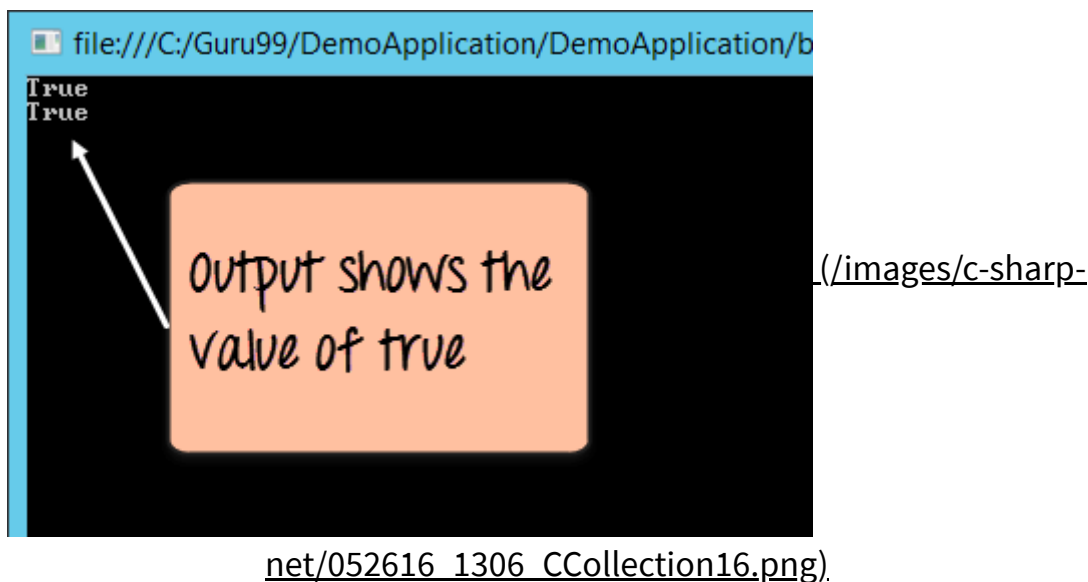
Explicación del código: -



1. Primero, usamos el método `ContainsKey` para ver si la clave está presente en la tabla hash. Este método devolverá verdadero si la clave está presente en la tabla hash. Este método debería devolver verdadero ya que la clave existe en la tabla hash.
2. Luego usamos el método `ContainsValue` para ver si el valor está presente en la tabla hash. Este método devolverá 'verdadero' ya que el valor existe en la tabla hash.

Si el código anterior se ingresa correctamente y el programa se ejecuta, se mostrará el siguiente resultado.

Salida:



En la salida, puede ver claramente que tanto la clave como el valor que se busca están presentes en la tabla hash.

Resumen

- Una tabla hash se utiliza para almacenar elementos que se componen de pares de valores clave. Para acceder al valor de un elemento, necesita conocer la clave del elemento.

◀ [Anterior \(/c-sharp-queue.html\)](/c-sharp-queue.html)

[Reportar un error](#)

[próximo](#) ▶ [\(/c-sharp-windows-forms-application.html\)](/c-sharp-windows-forms-application.html)

TE PODRÍA GUSTAR:

C#

C#

C#



(/c-sharp-stack.html) (/c-sharp-stack.html)



Pila de C # con ejemplo

(/c-sharp-stack.html)

(/c-sharp-inheritance-polymorphism.html) (/c-sharp-inheritance-



polymorphism.html)

Herencia y polimorfismo de C # con ejemplos

(/c-sharp-inheritance-polymorphism.html)

(/c-sharp-arrays.html) (/c-sharp-arrays.html)



Tutorial de matriz de C #: crear, declarar, inicializar

(/c-sharp-arrays.html)

C#

(/c-sharp-arraylist.html)



(/c-sharp-arraylist.html)

Tutorial de C # ArrayList con ejemplos

(/c-sharp-arraylist.html)

C#

(/c-sharp-



serialization.html) (/c-sharp-

serialization.html)

Serialización y deserialización de C # con ejemplo

(/c-sharp-serialization.html)

C#

(/c-sharp-interview-



questions.html) (/c-sharp-interview-

questions.html)

Las 52 preguntas y respuestas principales de la entrevista de C

(/c-sharp-interview-questions.html)

Tutorial de C

[17\) Pila de C # con ejemplo \(/c-sharp-stack.html\).](#)

[18\) Cola de C # \(/c-sharp-queue.html\).](#)

[19\) C # Hashtable \(/c-sharp-hashtable.html\).](#)

[20\) Aplicación C # Windows Forms \(/c-sharp-windows-forms-application.html\).](#)

[21\) Base de datos de acceso de C # \(/c-sharp-access-database.html\).](#)

?

f (<https://www.facebook.com/guru99com/>),
t (<https://twitter.com/guru99com>), **in**
(<https://www.linkedin.com/company/guru99/>).



(<https://www.youtube.com/channel/UC19i1XD6k88KqHlET8atqFQ>).



(<https://forms.aweber.com/form/46/724807646.htm>).

Acerca de

[Sobre nosotros \(/about-us.html\)](/about-us.html).

[Anuncie con nosotros \(/advertise-us.html\)](/advertise-us.html).

[Escríbanos \(/become-an-instructor.html\)](/become-an-instructor.html).

[Contáctenos \(/contact-us.html\)](/contact-us.html).

Sugerencia de carrera

(</best-sap-module.html>) Prueba de software de la
(</software-testing-career-complete-guide.html>) herramienta de sugerencia de carrera de
SAP (</best-sap-module.html>).
[como carrera \(/software-testing-career-complete-guide.html\)](/software-testing-career-complete-guide.html).

Interesante



[eBook \(/ebook-pdf.html\)](/ebook-pdf.html).

[Blog.\(/blog/\)](/blog/).

[Quiz \(/tests.html\)](/tests.html).

[SAP eBook \(/sap-ebook-pdf.html\)](/sap-ebook-pdf.html).

Ejecutar en línea

[Ejecutar Java en línea \(/try-java-editor.html\)](/try-java-editor.html).

[Ejecutar Javascript \(/execute-javascript-online.html\)](/execute-javascript-online.html).

[Ejecutar HTML \(/execute-html-online.html\)](/execute-html-online.html).

[Ejecutar Python \(/execute-python-online.html\)](/execute-python-online.html).

© Copyright - Guru99 2020

[Política de privacidad \(/privacy-policy.html\)](/privacy-policy.html).

| [Descargo de responsabilidad de afiliados \(/affiliate-earning-disclaimer.html\)](/affiliate-earning-disclaimer.html). | [ToS \(/terms-of-service.html\)](/terms-of-service.html).

