Retrieve an item from a Dictionary object

A dictionary<TKey, TValue> is a collection that stores key-value pairs, where each unique key is associated with one value and is optimized for fast lookups, making it ideal for scenarios where quick retrieval of values by a key is required. Dictionaries use hashing to organize and access data, allowing O(1) average time complexity for lookup, insertion, and deletion operations. Keys must be unique, and both keys and values can be of any specified type.

Using the Indexer

Example Implementation

You can directly access a value using its key. The indexer will throw an exception if the key does not exist.

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

class Program
{
  static void Main()
  {
    var dictionary = new Dictionary<string, int>
    {
      { "apple", 1 },
      { "banana", 2 },
      { "cherry", 3 }
    };

    // get item using the indexer
    int value = dictionary["banana"];
    Console.WriteLine($"The value for 'banana' is: {value}");
  }
}
```

Using TryGetValue

Example Implementation

This method is safer because it checks if the key exists before trying to access it, preventing potential KeyNotFoundException. TryGetValue returns true if the key exists and sets the output parameter to the corresponding value, allowing you to handle the absence of a key gracefully.

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

class Program
{
  static void Main()
  {
    var dictionary = new Dictionary<string, int>
    {
      { "apple", 1 },
      { "banana", 2 },
      { "cherry", 3 }
    };

    // try to get the item
    if (dictionary.TryGetValue("banana", out int value))
    {
      Console.WriteLine($"The value for 'banana' is: {value}");
    }
    else
    {
      Console.WriteLine("'banana' not found in the dictionary.");
    }
  }
}
```