# Dictionaries & Hashtables

A Dictionary is a collection that stores key-value pairs, they are generic types that allow static tpying and use without boxing. They are preferred over Hashtables in C# because they offer better performance and compile-time verification.

Hashtables are specific types of dictionary classes that use an integer value (called a hash) to aid in the store of its keys. They use the hash to speed up the searching for a specific key in the collection.

One subtle but important difference between the two is that Hashtables support multiple reader threads with a single writer thread, while Dictionaries offer no thread safety.

|  |  |
| --- | --- |
| **Hashtable** | **Dictionary** |
| Hashtable is included in the Sysm.Collections namespace. | Dictionary is included in the System.Collections.Generic namespace. |
| Hashtable is a loosely typed (non-generic) collection, this means it stores key-value pairs of any data types. | Dictionary is a generic collection. So it can store key-value pairs of specific data types. |
| Hashtable is thread safe. | Only public static members are thread safe in Dictionary. |
| Hashtable returns null if we try to find a key which does not exist. | Dictionary throws an exception if we try to find a key which does not exist. |
| Data retrieval is slower than the dictionary collection because of boxing-unboxing. | Data retrieval is faster than Hashtable because it is type safe, so no need for boxing-unboxing. |