**IEnumerable and List**

The main difference between IEnumerable and List in C# is that IEnumerable is an interface, while List is a concrete class. Moreover, IEnumerable is read-only, and List is not.

List represents the entire collection in memory, while IEnumerable provides an interface for getting the next item one-by-one (enumerating data).

Both IEnumerable and List are part of .NET's System.Collections namespace.

**IEnumerable is a deferred execution while List is an immediate execution.**

Difference between IEnumerable and List is clear when working with LINQ.

IEnumerable will not execute the query until you enumerate over the data, whereas List will execute the query as soon as it's called. Deferred execution makes IEnumerable faster because it only gets the data when needed.

* List is used to access an element in a specific position/index in a list.
* List is useful when you want to Add or remove items from the list. **I**Enumerable doesn't support add or remove items from the data.
* List can find out the no of elements in the collection without iterating the collection, but IEnumarable not.
* **I**Enumerable support further filtering, but List not.
* **I**Enumerable can move forward only over a collection, it can’t move backward and between the items.
* IEnumerable and List supports deferred execution.

LINQ queries are always executed when the query variable is iterated over, not when the query variable is created. This is called deferred execution. You can also force a query to execute immediately, which is useful for caching query results.