# Arrays, Lists & ArrayLists

An array is a collection of elements of the same type that are stored in contiguous memory locations. It is a fixed-size data structure, meaning that once it is created, its size cannot be changed. You can access its elements using an index, which starts from 0. Arrays are type-safe and are the most efficient of the three, both in terms of memory and performance. Also, System.Array supports multiple dimensions (i.e. it has a Rank property) while List and ArrayList do not (though you can create a List of Lists or an ArrayList of ArrayLists).

An ArrayList is a flexible array which contains a list of objects. You can add and remove items from it and it automatically deals with allocating space. If you store value types in it, they are boxed and unboxed, which can be a bit inefficient. Also, it is not type-safe.

A List is a data structure that represents a list of objects that can be accessed by index. Like an array, but can dynamically increase/decrease in size. A List leverages generics; its is essentially a type-safe version of ArrayList. This means there is no boxing or unboxing (which improves performance) and if you attempt to add an item of the wrong type it will generate a compile-time error.

The main differences between Arrays, ArrayLists and Lists are:

* Arrays are fixed in size, while ArrayLists are Lists are dynamic and can grow or shrink as needed.
* Arrays are more efficient in terms of memory and performance, while ArrayLists and Lists are more flexible and easier to use.
* Arrays and Lists can only store elements of the same type, while ArrayLists can store elements of different types.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | Array | List | ArrayList |
| **Size** | Fixed | Dynamic | Dynamic |
| **Memory Efficiency** | High | Low | Low |
| **Type Safety** | Yes | Yes | No |
| **Element Type** | Same | Same | Different |