**String Pool (String Intern Pool / String Constant Pool)**

**String pool** is a storage area **in** [**Java heap**](https://www.javatpoint.com/java-heap)where string literals store. MAintained by Java String class and by default is empty and private.

**What does it do?**

When a string literal is created JVM first check the string pool to see if the string already exists. If it does it simply returns the a reference to that string. If it doesn't exists a new String object is created in the pool. Its like object allocation. This is to increase performance and decrease memory load.

**Creation of Strings**

1. Using String Literal (These are in the string pool)
   1. String str = “Hello World”
2. Using “new” keyword (These are in the heap but not in the String pool)
   1. String str = new String(“Hello World”);

In this instance if pool has the string 1 instance will be created. If not one in heap and one in pool will be created making 2 instances in total.

**.intern() → Manual interning**

String.intern() -> Puts the string to the pool or refers to another object in the pool that has the same value. Determines if it exists or not with the String.equals**(String)** method.

Question: Is String object in the heap deleted after using the intern method?

**String pool is an example of the Flyweight Design Pattern.**

**ArrayList**

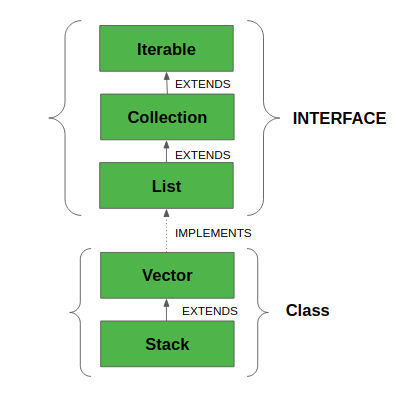
Resizable array. PRovides dynamic arrays. MAy be slower than normal arrays.

Real Life examples:

**Stack**

Last in first out (LIFO). Some operations:

1. Pop
2. Push
3. Empty
4. Peek
5. search



All implemented interfaces:

1. Serializable: must implement if they are to be serialized and deserialized.
2. Cloneable: to allow its objects to be cloned.
3. Iterable: collection of objects which is iterable
4. Collection: The Collection interface is used to pass around collections of objects where maximum generality is desired.
5. List: The List interface provides a way to store the ordered collection.
6. RandomAccess: indicate that they support fast (generally constant time) random access.

Real Life examples: Pringles, Stack of files to be read, pile of plates.

**Queue**

First In First Out (FIFO). Most common cases are PriorityQueue and LinkedLists. It is an interface so cannot create an object of Queue directly.

Real Life examples: Bank line, line in the traffic lights, line in the cafeteria.