Java Collections framework is consist of the interfaces and classes

An array is a container object that holds a fixed number of values of a single type in a contiguous memory location.

It is a data structure which is used to store finite number of elements and all elements must be of similar data type.

Arrays are index based data structure so they allow random access to elements, they store. Indices start with '0'.

Array copy: using

array clone --> It creates a shallow copy of array.

SerializationUtils.clone(empArray); --> create deep copy of an array

System.arrayCopy()

Arrays.copyOf()

Arrays.copyOfRange()

Use String.join() method to create string from String array

Use split() method to split string into tokens by passing a delimiter (or regex)

Java byte[] to String Example

String str = new String( byteArray );

//or

String str = new String( byteArray , charset );

byte array to string

byte[] bytes = "hello world".getBytes();

//Convert byte[] to String

String s = new String(bytes);

//String

String string = "howtodoinjava.com";

//Base64 Decoded

byte[] bytes = Base64.getDecoder().decode(string);

byte[] bytes = "hello world".getBytes();

//Convert byte[] to String

String s = Base64.getEncoder().encodeToString(bytes);

System.out.println(s);

union between two arrays in Java using HashSet class

To get the union of two arrays, follow these steps:

1. Push first array in a HashSet instance.
2. Use **addAll()** method to add the elements of the second array into set.
3. Similarly, add all the elements of more arrays in the set, if any.

**intersection between two**[**arrays**](https://howtodoinjava.com/java-array/)**in Java** using [HashSet](https://howtodoinjava.com/java/collections/java-hashset/) class.

To get the intersection of two arrays, follow these steps:

1. Push first array in a HashSet instance.
2. Use **[retainAll()](https://howtodoinjava.com/java/collections/arraylist/arraylist-retainall-example/)** method to retain only elements which are present in second array.

**Remove duplicates from array using LinkedHashSet**

Using Java Collections, **LinkedHashSet** is one of the best approaches for removing the duplicates from an array. [LinkedHashSet](https://howtodoinjava.com/java/collections/java-linkedhashset/) achieves two objectives :

* Removes the duplicate elements, as well as
* Maintains the ordering of elements
* The [Java 8](https://howtodoinjava.com/java-8-tutorial/) [Stream.distinct()](https://howtodoinjava.com/java8/java-stream-distinct-examples/) method returns an Stream consisting of the distinct elements of a given Stream.