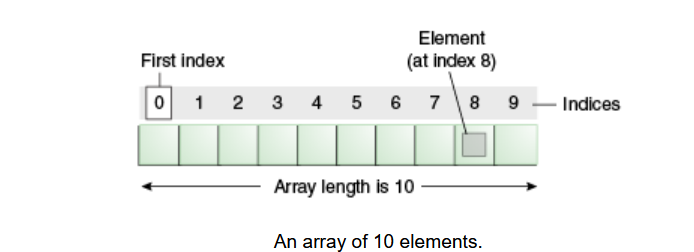
# Java

## Data Types

* byte
* short
* int
* long
* float
* double
* boolean
* char
* String

## Arrays

### Definition

* Is a container object -> holds a fixed number of values of a single type.
* Length of the array -> decided during the array creation. Once created -> length is fixed.
* 
* Element: Is each item in the array, is accessed by its numerical index.
* Refer an example code (DeclaringAndInitializingAnArray) in the repository [LearningJava](https://github.com/Phanindra08/LearningJava) under the arrays package. In Real world programming, will use one of the supported looping constructs to iterate through each element in the array rather than writing each line individually as demonstrated in the above example.

### Declaration

* Two components in the array declaration: array’s type and array’s name.
  + Array’s type is written as type[] -> type is datatype of the elements in the array, [] -> special symbol for indicating this variable holds an array. The size of the array is not part of its type; hence brackets are empty.
  + Array’s name -> Can be anything as long as it follows the rules of the naming convention.
* The declaration will not create an array (i.e., it is not same as declaring variables of other types). Only tells the compiler, the variable holds an array of specified type.
* Can declare an array of any data type.
* Can also place the brackets ([]) after the array’s name but this form is discouraged as the [] identify array type, hence should appear with the type designation.

### Creating an Array

1. Can use ‘new’ operator. The statement ‘anArray = new int[10];’ -> Allocates an array with memory to store 10 integer elements and assigns it to ‘anArray’ variable.
   1. If the array is not initialized with ‘new’ keyword like the above step then the compilation will fail with the error message as below printed by the compiler.
2. The below is a way to create and initialize an array. Length of the array -> determined by the number of values provided between the braces and separated by commas.



### Multidimensional Arrays

1. Can also declare array of arrays. Use two or more sets of brackets for multidimensional arrays.
2. Ex: String[][] names
3. Unlike C or Fortran languages, multidimensional arrays in Java are an array whose components are arrays. Consequence of this is rows are allowed to vary in length.
4. Refer an example code (MultiDimArrayDemo) in the repository [LearningJava](https://github.com/Phanindra08/LearningJava) under the arrays package.

### Methods and Properties

* Length: Is a built-in property. Used to determine size of any array.

### Copying Arrays

* Using the ‘arraycopy’ method from System class.
* Refer an example code (ArrayCopyDemo) in the repository [LearningJava](https://github.com/Phanindra08/LearningJava) under the arrays package.

### Array Manipulations

* Java provides methods to perform some of the most common manipulations related to arrays using methods. This enables the developer to use just one line of the code to call the method.
* Java provided several methods for performing array manipulations in the ‘java.util.Arrays‘ class. The common tasks are copying, sorting, searching etc., in the arrays.
* Can use ‘copyOfRange’ method instead of ‘arraycopy’ method. The difference is ‘copyOfRange’ method does not require us to create the destination array before calling the method as the destination array is returned by the method.

## Classes

### System class

* **‘arraycopy’ method:**
  + Used to efficiently copy data from one array into another.
  + **Definition:** public static void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)
  + Two Object arguments specify the array to copy from and the array to copy to.
  + Three int arguments specify the starting position in the source array, the starting position in the destination array and the number of array elements to copy.

### Arrays class

* **‘copyOfRange’ method:**
  + Can use ‘copyOfRange’ method instead of ‘arraycopy’ method. The difference is ‘copyOfRange’ method does not require us to create the destination array before calling the method as the destination array is returned by the method.
  + **Definition:**
  + Second parameter is the initial index of the range to be copied, inclusively.
  + Third parameter is the final index of the range to be copied, exclusively.
* **‘binarySearch’ method:**
  + Searching an array for a specific value to get the index at which it is placed.
* **‘equals’ method:**
  + Comparing two arrays to determine if they are equal or not.
* **‘fill’ method:**
  + Filling an array to place a specific value at each index.
* **‘sort’ method:**
  + Sorting an array into ascending order sequentially.
* **‘parallelSort’ method:**
  + Sorting an array into ascending order concurrently. Introduced in Java 8.
  + Using parallel sort on large arrays on multiprocessor system is faster than the sequential array sorting.
* **‘stream’ method:**

**To Refer Later:**

* <https://docs.oracle.com/javase/tutorial/collections/streams/index.html>