JAVA.UTIL.ARRAYS.SORT METHOD

Description

The **java.util.Arrays.sort*int[]a, intfromIndex, inttoIndex*** method sorts the specified range of the specified array of ints into ascending numerical order. The range to be sorted extends from index *fromIndex*, inclusive, to index *toIndex*, exclusive.

Declaration

Following is the declaration for **java.util.Arrays.sort** method

```
public static void sort(int[] a, int fromIndex, int toIndex)
```

Parameters

- **a** -- This is the array to be sorted.
- **fromIndex** -- This is the index of the first element *inclusive* to be sorted.
- **toIndex** -- This is the index of the last element *exclusive* to be sorted .

Return Value

This method does not return any value.

Exception

- **IllegalArgumentException** -- if fromIndex > toIndex
- **ArrayIndexOutOfBoundsException** -- if fromIndex < 0 or toIndex > a.length

Example

The following example shows the usage of java.util.Arrays.sort method.

```
package com.tutorialspoint;

import java.util.Arrays;

public class ArrayDemo {

   public static void main(String[] args) {

   // initializing unsorted int array
   int iArr[] = {3, 1, 2, 18, 10};

   // let us print all the elements available in list
   for (int number : iArr) {
   System.out.println("Number = " + number);
   }

   // sorting array from index 0 to 3
   Arrays.sort(iArr, 0, 3);

   // let us print all the elements available in list
   System.out.println("int array with some sorted values(0 to 3) is:");
   for (int number : iArr) {
   System.out.println("Number = " + number);
   }
   }
}
```

Let us compile and run the above program, this will produce the following result:

```
Number = 3
Number = 1
Number = 2
Number = 18
Number = 10
int array with some sorted values(0 to 3) is:
Number = 1
Number = 2
Number = 3
Number = 18
Number = 10
```