Java Array Syntax

Declaration and Initialization:

// Declaration

dataType[] arrayName;

// Initialization

dataType[] arrayName = new dataType[size];

dataType[] arrayName = {element1, element2, ...};

Example:

int[] numbers = new int[5]	// Declaration and initialization with size
int[] primes = {2, 3, 5, 7, 11}	// Declaration and initialization with values

Accessing Elements:

arrayName[index];

Example:

int thirdElement = numbers[2]; // Accessing the third element

Updating Elements:

arrayName[index] = newValue;

Example:

numbers[0] = 10; // Updating the first element

Iterating through an Array:

for (dataType element : arrayName) {

```
    // Process element
}
```

Example:

```
for (int number : numbers) {
    System.out.println(number); // Print each number in the array
}
```

Counting Array Length:

Example:

```
int size = numbers.length; // Get the length of the array
```

Sorting Arrays:

```
Arrays.sort(arrayName);
```

Example:

```
Arrays.sort(numbers); // Sorts the array 'numbers' in ascending order
```

Arrays.sort():

- Arrays.sort() is used to sort arrays of primitive data types or objects that implement the Comparable interface.
- It sorts arrays in place, modifying the original array.
- This would sort the numbers array in ascending order.

Copying Arrays:

Method 1- Using a Loop

```
// Create a new array with the same length as the original
int[] originalArray = {1, 2, 3, 4, 5};
```

```
int[] copiedArray = new int[originalArray.length];


// Copy elements one by one
for (int i = 0; i < originalArray.length; i++) {
   copiedArray[i] = originalArray[i];
}
```

Method 2- Using System.arraycopy():

```
// Create a new array with the same length as the original
int[] originalArray = {1, 2, 3, 4, 5};
int[] copiedArray = new int[originalArray.length];


// Use System.arraycopy() to copy the array
System.arraycopy(originalArray, 0, copiedArray, 0, originalArray.length);
```

Method 3- Using Arrays.copyOf():

```
// Create a new array with the same length as the original and copy its elements
int[] originalArray = {1, 2, 3, 4, 5};
int[] copiedArray = Arrays.copyOf(originalArray, originalArray.length);
```

Note - All three methods will create a copy of the original array. The choice of method depends on your preference and requirements. System.arraycopy() is generally faster for large arrays, while Arrays.copyOf() provides a more concise syntax.