What is an array, really?

An array is a special class in Java.

It's still a class.

The array, like all other classes, ultimately inherits from java.lang.Object.

Array initialization and default element values

When you don't use an array initializer statement, all array elements get initialized to the default value for that type.

For primitive types, this is **zero** for any kind of **numeric primitive**, like int, double, or short.

For **booleans**, the default value will be **false**.

And for any **class** type, the elements will be initialized to **null**.

The Enhanced For Loop, the For Each Loop

This loop was designed to walk through elements in an array or other collection types.

It processes **one element at a time**, from the **first** element to the **last**.

Here is the syntax for the two types of for loop statements, side by side.

Enhanced For Loop	**Basic For Loop**

`for (declaration : collection) {`
`    // block of statements`
`}`
	`for (init; expression; increment) {`
	`    // block of statements`
	`}`

The enhanced for loop only has two components, versus three defined in the parentheses after the for keyword.

The Enhanced For Loop, the For Each Loop

Enhanced For Loop	**Basic For Loop**

`for (declaration : collection) {`
`    // block of statements`
`}`
	`for (init; expression; increment) {`
	`    // block of statements`
	`}`

It's important to notice that the separator character between components is a **colon** and not a semicolon for the enhanced For Loop.

The **first** part is a **declaration expression** which includes the type and a variable name. This is usually a local variable with the same type as an element in the array.

And the **second** component is the **array** or some other collection variable.

java.util.Arrays

Java's array type is very basic, it comes with very little built-in functionality.

It has a single property or field, named length.

And it inherits java.util.Object's functionality.

Java provides a helper class named java.util.Arrays, providing the common functionality you'd want for many array operations.

These are static methods on Arrays, they are class methods, not instance methods.

Printing elements in an array using Arrays.toString()

The toString method in this helper class prints out all the array elements, comma delimited, and contained in square brackets.

```
String arrayElementsInAString = Arrays.toString(newArray);
```

The output from this method is shown here conceptually.

It prints the element at index 0 first, followed by a comma, then the element at index 1 next, a comma, and so on, until all elements are printed.

```
[ e[0], e[1], e[2], e[3], ... ]
```