

Arrays in Java

Review

What is an Array?

An array is a collection of identical types of data. For example, all integers, or all strings.

An array is a data structure, meaning it will always have a consistent and predictable form. For example, in Java all arrays are indexed started at zero, and the index is used to access each element of the array.

An array is an object in memory. The array variable points to a location in memory.

Creating an Array

Declare an array variable by specifying the data type and giving the array variable a name:

```
double[] prices;  
int[] ages;  
String[] names;
```

To allocate memory to an array, an additional step is required. For example:

```
double[] prices = new double[5];  
int[] ages = new int[10];
```

Initializing an Array

If an array is created in memory, default values will be used (similar to creating an object).

An array can be initialized to other values when created using an assignment statement. This isn't always a practical option.

```
String[] names = {"Bob", "Mary"};
```

Initializing an Array

If an array is created in memory, default values will be used (similar to creating an object): numbers are zero, boolean are false, and Strings are empty ("").

An array can be initialized to other values when created using an assignment statement. This isn't always a practical option.

```
String[] names = {"Bob", "Mary"};
```

Traversing an Array

Traversing an array generally means visiting each element of the array, one at a time, in a systematic way. This is accomplished using a loop.

```
for (int i = 0; i < array.length(); i++)  
{  
    array[i] = value;  
}
```

Arrays as Objects

Just like other objects, an array is actually made up of two different parts:

- (1) the named array variable
- (2) the array data stored in memory

This distinction becomes very important when using arrays under the following conditions:

- (1) output array data
- (2) comparing arrays to each other
- (3) passing arrays as parameters to methods

Output an Array

Since the array variable is a memory location, it is not simple to output the contents of an array.

```
System.out.println(array) ;
```

This will print the address of the array in memory.

Instead, the array must be traversed, and each element output or processed individually.

```
for (int i = 0; i < array.length(); i++)  
{  
    System.out.println(array[i]) ;  
}
```


Comparing Arrays

```
boolean same = array1 == array2;
```

This compares the addresses of each array.

Instead, the array must be traversed, and each element compared individually.

```
int length = min(array1.length(), array2.length());  
for (int i = 0; i < length; i++)  
{  
    same = same && (array1[i] == array2[i]);  
}
```