

Arrays



Arrays in Java

- In Java, arrays are objects
- They generally can be used like arrays are in other languages.
- One-dimensional and Multi-dimensional
- Array index starts at zero. Thus the last element of an N size array will have an index of $N - 1$.

One-Dimensional Array Declaration

General form:

```
type array-name[ ] = new type[ size ];
```

Example1:

```
int sample[ ] = new int[10];
```

Creates an array of ten integers

One-Dimensional Array Declaration

General form:

```
type array-name[ ] = new type[ size ];
```

Example2:

```
int sample[ ];  
sample = new int[10];
```

Illustrates a two-step way of creating
an array of ten integers

One-Dimensional Array Declaration with Initialization

General form:

```
type array-name[ ] = { comma separated list };
```

Example3:

```
int sample[ ] = { 23, -9, 44, 56, -600 };
```

An array of 5 integers named `sample` is created and initialized with the given values.

Initializing a one dimensional array (an example)

```
class MinMax2 {
    public static void main(String args[]) {
        int nums[] = { 99, -10, 100123, 18, -978,
                      5623, 463, -9, 287, 49 };
        int min, max;

        min = max = nums[0];
        for(int i=1; i < 10; i++) {
            if(nums[i] < min) min = nums[i];
            if(nums[i] > max) max = nums[i];
        }
        System.out.println("Min and max: " + min + " " + max);
    }
}
```

Two-Dimensional Array Declaration

General form:

```
type array-name[ ] [ ] = new type[ size ] [ size ];
```

Example:

```
int sample[ ] [ ] = new int[10] [20];
```

Two-Dimensional Array Declaration

General form:

```
type array-name[ ] [ ] = new type[ size ] [ size ];  
                                Row   Col
```

Example:

```
int sample[ ] [ ] = new int[10] [20];
```


Two dimensional arrays

(an example)

```
class TwoD {
    public static void main(String args[]) {
        int t, i;
        int table[][] = new int[3][4];

        for(t=0; t < 3; ++t) {
            for(i=0; i < 4; ++i) {
                table[t][i] = (t*4)+i+1;
                System.out.print(table[t][i] + " ");
            }
            System.out.println();
        }
    }
}
```

```
class Squares {
    public static void main(String args[]) {
        int sqrs[][] = {
            { 1, 1 },
            { 2, 4 },
            { 3, 9 },
            { 4, 16 },
            { 5, 25 },
            { 6, 36 },
            { 7, 49 },
            { 8, 64 },
            { 9, 81 },
            { 10, 100 }
        };
        int i, j;

        for(i=0; i < 10; i++) {
            for(j=0; j < 2; j++)
                System.out.print(sqrs[i][j] + " ");
            System.out.println();
        }
    }
}
```

Each row has its own
set of initializers

Finding the Length of an Array

Each array has associated with it a length method that returns the length as an integer

```
int list[] = new int[10];
int nums[] = { 1, 2, 3 };

System.out.println("length of list is " + list.length);
System.out.println("length of nums is " + nums.length);

// use length to initialize list
for(int i=0; i < list.length; i++)
    list[i] = i * i;
```

