Maydm

Arrays

Maydm

Java Arrays

Arrays store one or more values in a single variable. This saves programmers from needing to declare separate variables for each value.

Arrays are declared with [] square brackets

Think of Arrays as a Collection or Grouping

An Array of Students

An Array of Coders

Maydm

Arrays are a Group of Objects

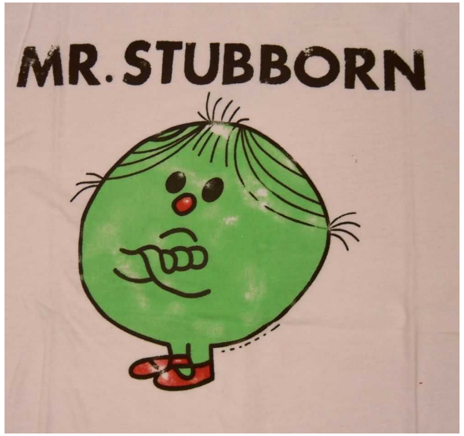
An Array of Keys

An Array of Pokemon
(and a boy)

Maydm

An Immutable Container Object

Arrays hold an unchangeable number of values of a single type. Arrays, like Strings, are immutable. Once established the number of containers in the array object cannot be changed. Remember, this is called immutability.



Maydm

All About Arrays

An array is a container object that holds a fixed number of values for a single type. So an array cannot hold both String and integer values or floats and integers. Only one type per array.

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

Creates an Array of Integers with 10 indices.

```
String[] anArrayOfStrings;
anArrayOfStrings = new String[5];
```

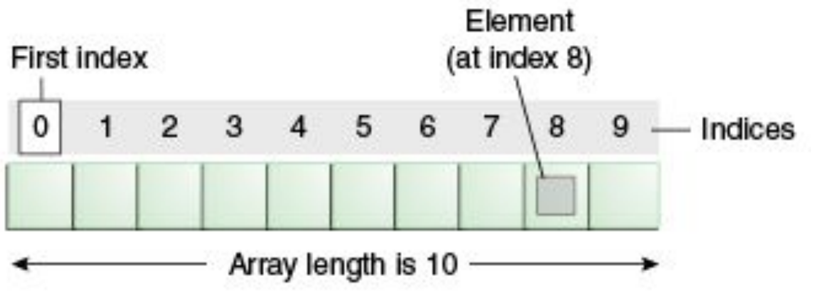
Creates an Array of Strings with 5 indices.

Maydm

Arrays, Index, Indices

If you've ever read the back of a reference book then you know an index is a list, or a catalog of items. Arrays can be thought of as an index as well.

Indices is simply the plural form of index.

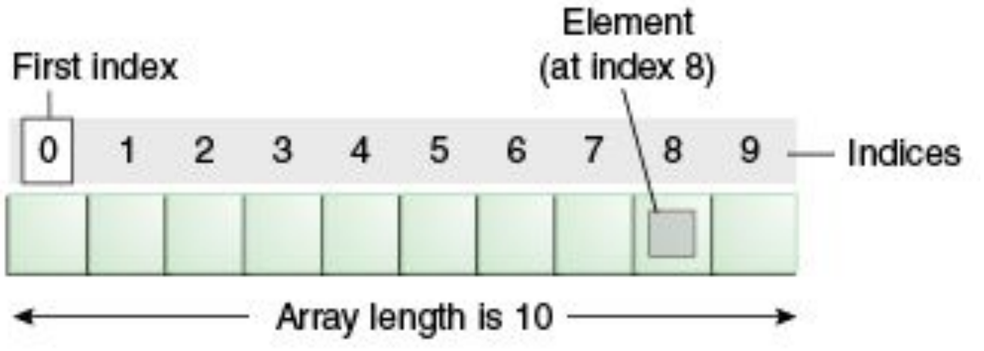


Maydm

Start at Zero

Did you notice the index image in the last slide? Did you see that the index starts at zero?

So an array of ten objects will have indices ranging from 0 to 9.



Maydm

Immutable Length and Type

Arrays are immutable. Remember, that means that once an array's type and length is set it cannot be changed. Specific values within the array can be mutated, however.

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

This array will always be a collection of 10 integers. No Strings or decimals allowed. The indices will be numbered 0 through 9

```
String[] anArrayOfStrings;
anArrayOfStrings = new String[5];
```

This array can only be a collection of 5 Strings. The indices will range from 0 to 4.

Maydm

Changing an element’s value

We can change a specific element within an array by assigning it a new value.

```
String[] plants = {“sunflower”, “daisy”, “buckthorn”};
plants[2] = “wintergreen”;
```

Practice Arrays

Open *initial-array-practice.txt* **in the repository and practice writing arrays a couple of different ways.**

More Practice with Arrays

Let’s create an array and give it values. Open ArrayPractice.txt to begin

```
Class ArrayPractice {
    public static void main(String[] args) {
        // creates array of integers
        int[] anArrayOfIntegers;
        // reserves memory for 10 integers
        anArrayOfIntegers = new int[10];
        // Instantiate first element (element #0)
        anArrayOfIntegers[0] = 50;
        // Initialize second element (element #1)
        anArrayOfIntegers[1] = 100;
        // So on and so forth
        anArrayOfIntegers[2] = 150;
```

All the types of an Array

An array can be declared as any of the primitive types or String.

```
type[] nameOfArray;
byte[] anArrayOfBytes;
short[] anArrayOfShorts;
Int[] anArrayOfInts;
long[] anArrayOfLongs;
float[] anArrayOfFloats;
double[] anArrayOfDoubles;
char[] anArrayOfChars;
String[] anArrayOfStrings;
```

Maydm

Anatomy of an Array

Array Declaration

```
type[] nameOfArray;
```

Array name

A declaration like does NOT create an array!

It tells the complier that an array of that specific type will be held inside of that variable

Create an Array

Var name

```
myArray = new int[10]
```

Array declaration

This is how we create an array. An array must be declared with the same type as it was declared to be.

Maydm

Practice writing each type of array

Open *write-one-of-each-array.txt*

Another way of writing arrays

If you know the values of the array indices then you can insert values with an array literal - a list of values, separated by commas, inside curly braces.

```
int [] numbers = {1, 2, 3, 4};
String[] biomes = {forest, ocean, desert}
```

Practice writing your own arrays with *write-array-literals.txt*

How long is that array?

Remember how we can apply different methods to strings to learn more about their contents? We can do the same with arrays. One of the most common Array properties is length.

To find the length of an array, invoke the array variable followed by `.length`**.**

```
String[] people = {“DeShawn”, “Alexander”, “Winnie”};
people.length;
```

How long is that array?

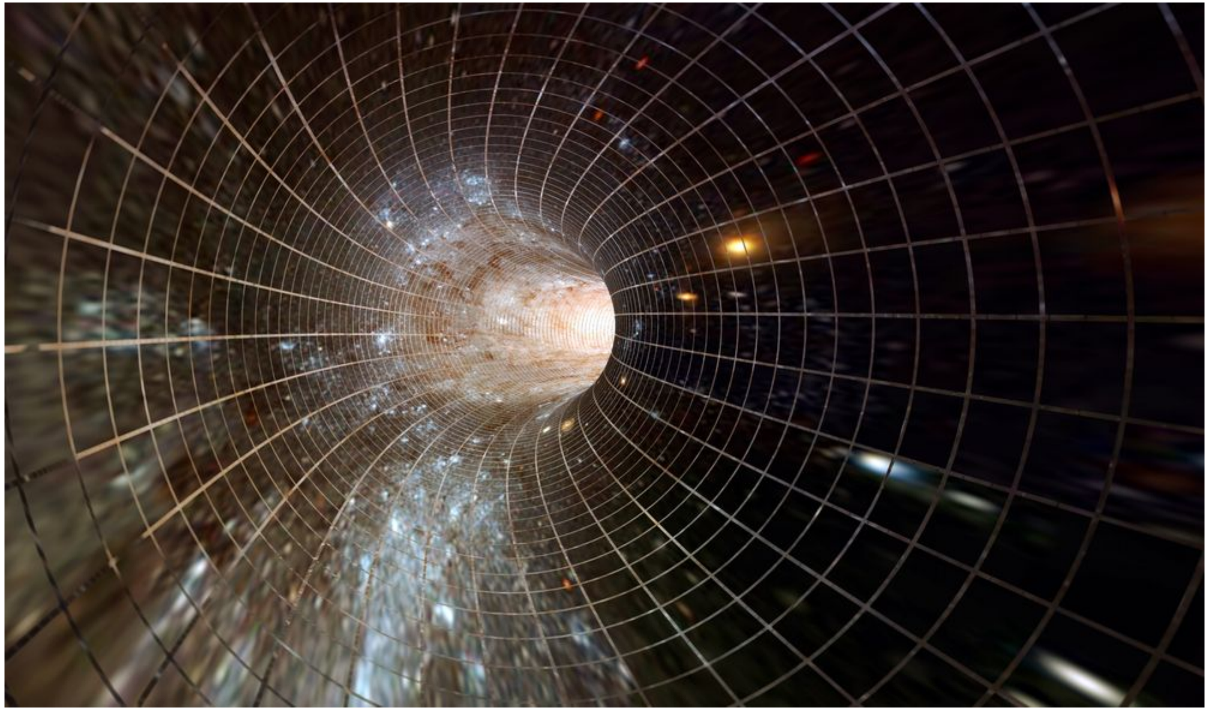
Let's practice using the length property to discover how long a few arrays are.

Open *TheLengthOfArrays.java* and solve the questions.

If you need help writing the length property then try Googling "Java Array length w3schools"

Maydm

Multidimensional Arrays


This is not what we're talking about

Multidimensional Arrays

Multidimensional arrays are not nearly as complicated as you might expect.

They are simply arrays that contain more than one array. Let's look at an example of a two-dimensional and three-dimensional array.

```
int[][] 2DNumbers = {{1, 2, 3}, {4, 5, 6, 7}};

int[][][] 3DNumbers = {{{1,2}, {3,4}}, {{5,6}}};
```

As you can see, multidimensional arrays separated by commas and curly braces. Open *multi-dimensional-arrays.txt* to write your own.

Maydm

That's all for now

That's all we'll say about arrays for now, but they are so common in code that we'll return to arrays once we learn about loops!

Let's take a moment for an array of questions

TYPE FORM