Arrays

By: Megan Avery
Updated November 2018

Before we begin

- Make sure you have Java & Java JDK downloaded on your computer
  - can run `java -version` in terminal to check
- Make sure you have IntelliJ downloaded on your computer

- *Suggested:* Watch previous Java tutorials

What is an array?

An array is an object that stores many values of the same type. All the elements of an array must be of the same type.
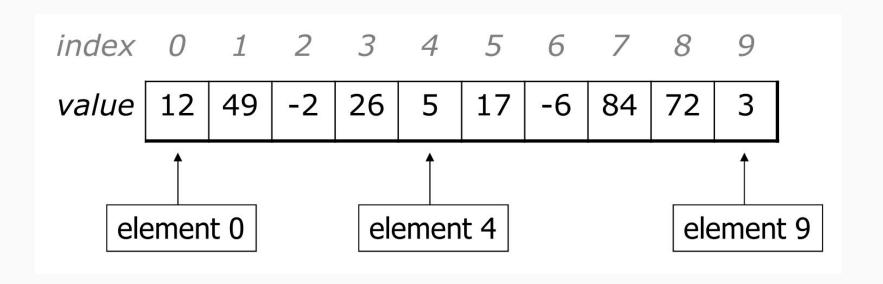
Vocab Alert!

element: one value in an array

index: a 0-based integer to access an element from an array

- Array are 0 indexed just like Strings

Index Example

Big Advantage of an Array

** Can replace many variables with just one object **

Array Code Example

```
// array declaration
int[] numbers = new int[10];

// array access
int x = numbers[3];

// array modification
numbers[2] = 4;
```

Default Array Element Values

Type	Default value
`int`	`0`
`double`	`0.0`
`boolean`	`false`
`String` or other object	`null` (means, "no object")

The length field

An array's length field stores its number of elements.

It is accessed like:

```
int[] numbers = new int[10];
int x =numbers.length; //10
```

Quick Array Initialization

You can declare an array and set all of its values at the same time

```java
int[] numbers = {12, 49, -2, 26, 5, 17, -6};
```

- Useful when you know what the array's elements will be
- The compiler determines the length by counting the values

Common Errors

Reading an index that is "out of bounds" aka NOT between 0 and array.length - 1
- Reading or writing any index outside this range will throw an ArrayIndexOutOfBoundsException

Creating an array of Objects (like Strings) which are defaulted to null for every position and then trying to call a method on an element
- Trying to call a method on a null element will throw a NullPointerException

Limitations of Arrays

- You cannot resize an existing array
- You cannot compare arrays with == or .equals
- An array does not know how to print itself it will just print its memory address if you try

The Arrays Class

Class Arrays in package java.util has useful static methods for manipulating arrays, it helps solve some of the limitations of an array

Method name	Description
`binarySearch(`**`<array>`**`, `**`<value>`**`)`	returns the index of the given value in a *sorted* array (or < 0 if not found)
`copyOf(`**`<array>`**`, `**`<length>`**`)`	returns a new copy of an array
`equals(`**`<array1>`**`, `**`<array2>`**`)`	returns `true` if the two arrays contain same elements in the same order
`fill(`**`<array>`**`, `**`<value>`**`)`	sets every element to the given value
`sort(`**`<array>`**`)`	arranges the elements into sorted order
`toString(`**`<array>`**`)`	returns a string representing the array, such as `"[10, 30, -25, 17]"`

Syntax for usage: Arrays.<methodName>(<parameters>)
Note: Must import java.util.Arrays;

Vocab Alert!

traversal: An examination of each element of an array

Arrays and for loops

It is common to use for loops to access array elements.

Example:

```java
int[] numbers = {12, 49, -2, 26, 5, 17, -6};
for (int i = 0; i < numbers.length; i++) {
    System.out.print(numbers[i] + " ");
}

System.out.println();
// output: 12 49 -2 26 5 17 -6
```

Exercises

What is output by the following code?

Time Limit: 3 minutes

```java
String[] names = new String[5];
names[1] = "Olivia";
names[3] = "Isabelle";
System.out.print(names[0].length());
```

Array Exercise #1

Time Limit: 5 minutes

Write a method that takes in a base and a power and stores the powers of the base in an array up to the given power. Then, print out the array at the end of the method.

Example: Given 2 and 4 the output would be [1, 2, 4, 8, 16]

Array Exercise #2

Time Limit: 6 minutes

Write a method that prompts the user for 4 numbers. Store the numbers they enter in an array. After they have entered the numbers print out the largest and smallest numbers in the array.

Array Exercise #3

Time Limit: 10 minutes

A fibonacci sequence is a sequence of numbers where the current number is the sum of the previous 2 numbers. Write a method that creates a fibonacci sequence that is stored in an array and prints it.

Fibonacci sequence to create: [1, 1, 2, 3, 5, 8, 13, 21]

Practice for Later

Write a program that prompts the user to enter a number of days to keep track of the weather and then prompt the user for the temperature on that many days as a whole number in degrees.

Then print:
- The average temperature
- How many days were above the average
- The temperature on the 2 coldest days
- The temperature on the 2 hottest days

The End