CODEGYM

Java Developer in 12 months

Mentor-supported training program from CodeGym

MODULE 1. JAVA SYNTAX

Lesson 6 Arrays

CODEGYM

Lesson plan

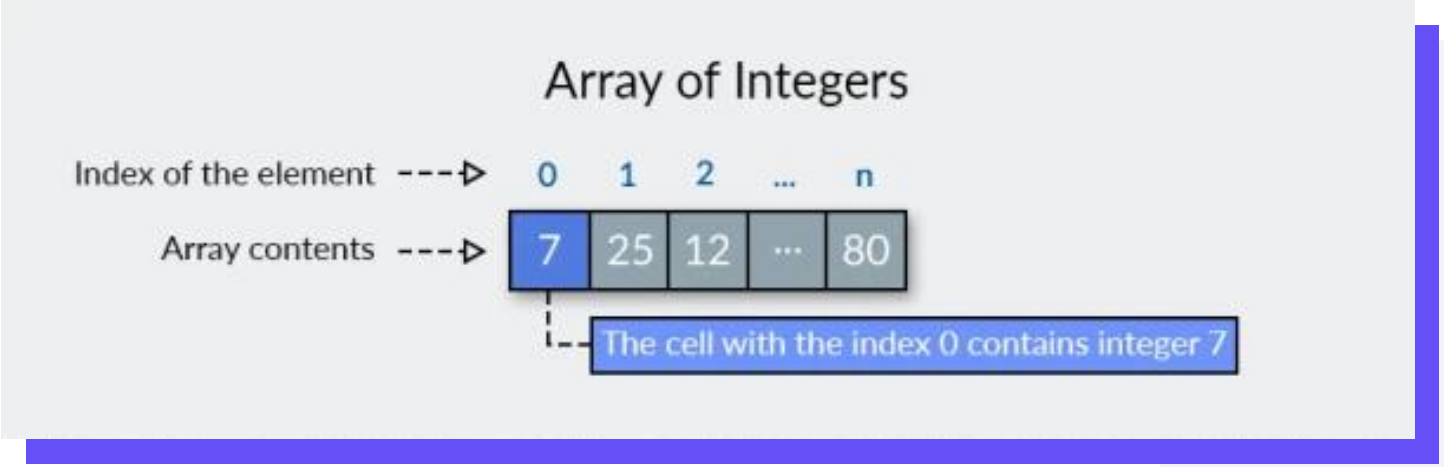
- Arrays
- Working with the cells of an array
- Array length
- Fast array initialization in Java

CODEGYM

Array

An array is a data structure that stores elements of the same type. You can think of it as a set of numbered cells. You can put some data in each cell.

A specific cell is accessed using its number. An element's number in the array is also called an index.



CODEGYM

Creating an array

You can create an array in Java using the following operator: `new`

```
new ArrayType[length];
```

Examples:

`int[] array;`
 	
`array = new int[100];`
	Declare an array
	
	Create (allocate memory for) an array of 100 ints

`int[] array = new int[10];`	Declare the variable and allocate memory

You cannot change the size of an array after it is created. You can create a new one, but the length of the existing container cannot be changed.

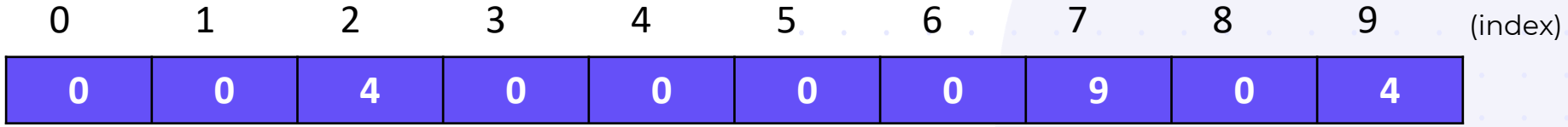
After an array is created using the new operator, its cells contain default values. For numeric types, the default value is 0, for the boolean type, it is false, and for reference types, it is null.

CODEGYM

Working with the cells of an array

The numbering of cells in an array always starts from zero.
If we have an array of 10 elements, then the numbers (indices) of its cells are 0..9 inclusive. If the array contains 200 elements, then the indices are 0..199 inclusive.

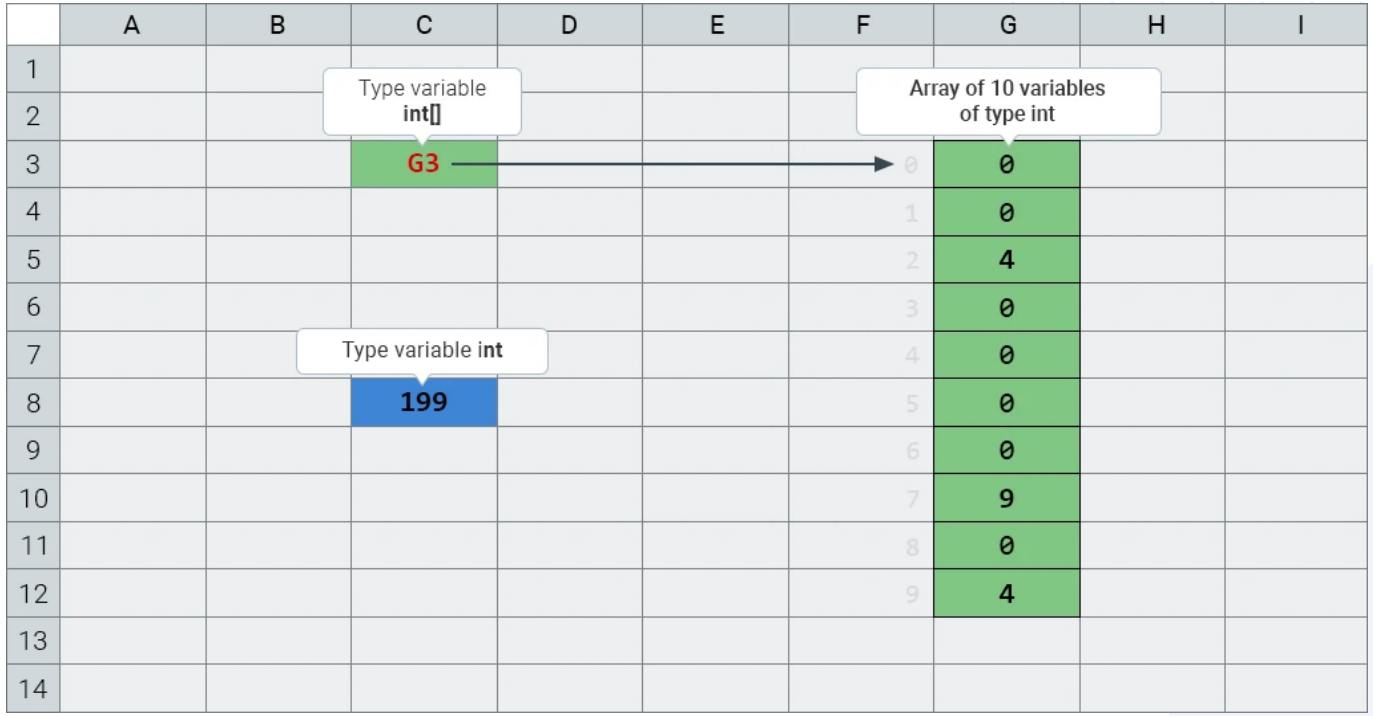
Code	Explanation
`int[] a = new int[10];`	Create an array of 10 ints.
`a[2] = 4;`	In the cell with the index 2, write the value 4.
`a[7] = 9;`	In the cell with the index 7, write the value 9.
`a[9] = a[2] + a[5];`	In the cell with index 9, write the sum of the values that are stored in cells 2 (which stores the value 4) and 5 (which stores the value 0).



CODEGYM

Arrays in memory

When creating arrays (as when creating strings), two separate blocks of memory are allocated: one for storing the array (container) itself and a second block for the variable that stores its address.



CODEGYM

Array length

Find the length of an array using this expression:

```
array.length;
```

array is the array variable's name
length is the name of the array's property

Code	Explanation

`int[] array;`
`if (a < 10)`
`  array = new int[10];`
`else`
`  array = new int[20];`
`for (int i = 0; i < array.length; i++){`
`  System.out.println(array[i]);`
`}`
	Create an int[] array
	If the a variable is less than 10,
	then create an array of 10 elements.
	Otherwise,
	create an array of 20 elements
	Loop over all the elements of the array: from 0 to length (array.length - 1)

CODEGYM

String array

Like any programming language, Java lets you create arrays of strings.
The type of any string in Java is String.

A type of a one-dimensional array is String[].

```
String[] name = new String[size];
```

name is the name of the array variable
size is the size of the array (number of strings)

After an array is created using the new operator, its cells contain default values.
For reference types, including String, the default value is null.

CODEGYM

Fast array initialization in Java

Array initialization is the process of filling an array with specific values (other than the default).

Suppose we need to store the lengths of each month in an array.
This is what the code might look like:

```
int[] months = new int[12];
months[0] = 31; // January
months[1] = 28; // February
months[2] = 31; // March
months[3] = 30; // April
months[4] = 31; // May
months[5] = 30; // June
months[6] = 31; // July
months[7] = 31; // August
months[8] = 30; // September
months[9] = 31; // October
months[10] = 30; // November
months[11] = 31; // December
```

```
// Lengths of months of the year
int[] months = new int[] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
```

This works for types other than int:

```
// Names of months of the year
String[] months = { "January", "February", "March", "April", "May", "June", "July",
"August", "September", "October", "November", "December"};
```

CODEGYM

Homework

MODULE 1. JAVA SYNTAX

Complete Level 7

CODEGYM

Answers to questions

MODULE 1. JAVA SYNTAX