

Java Arrays

Previous

Next >

Java Arrays

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

To declare an array, define the variable type with **square brackets**:

```
String[] cars;
```

We have now declared a variable that holds an array of strings. To insert values to it, we can use an array literal - place the values in a comma-separated list, inside curly braces:

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
```

To create an array of integers, you could write:

```
int[] myNum = {10, 20, 30, 40};
```





HTML

CSS MORE ▼

EXERCISES ▼



Access the Elements of an Array

You access an array element by referring to the index number.

This statement accesses the value of the first element in cars:

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars[0]);
// Outputs Volvo
```

Try it Yourself »

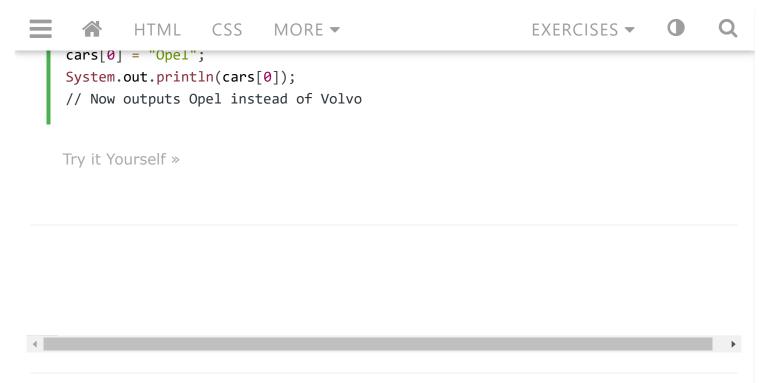
Note: Array indexes start with 0: [0] is the first element. [1] is the second element, etc.

Change an Array Element

To change the value of a specific element, refer to the index number:

Example

Example



Array Length

To find out how many elements an array has, use the length property:

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
System.out.println(cars.length);
// Outputs 4

Try it Yourself »
```

Loop Through an Array

You can loop through the array elements with the for loop, and use the length property to specify how many times the loop should run.

The following example outputs all elements in the **cars** array:





HTML

CSS MORE ▼







```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (int i = 0; i < cars.length; i++) {</pre>
  System.out.println(cars[i]);
}
```

Try it Yourself »

Loop Through an Array with For-Each

There is also a "for-each" loop, which is used exclusively to loop through elements in arrays:

Syntax

```
for (type variable : arrayname) {
```

The following example outputs all elements in the **cars** array, using a **"for-each"** loop:

Example

```
String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};
for (String i : cars) {
  System.out.println(i);
```

Try it Yourself »





HTML

CSS MORE ▼

EXERCISES ▼



If you compare the for loop and for-each loop, you will see that the for-each method is easier to write, it does not require a counter (using the length property), and it is more readable.

Multidimensional Arrays

A multidimensional array is an array containing one or more arrays.

To create a two-dimensional array, add each array within its own set of curly braces:

Example

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

myNumbers is now an array with two arrays as its elements.

To access the elements of the **myNumbers** array, specify two indexes: one for the array, and one for the element inside that array. This example accesses the third element (2) in the second array (1) of myNumbers:

Example

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
int x = myNumbers[1][2];
System.out.println(x); // Outputs 7
```

Try it Yourself »

We can also use a for loop inside another for loop to get the elements of a twodimensional array (we still have to point to the two indexes):





HTML CSS MORE ▼

EXERCISES ▼



0

```
public class Main {
   public static void main(String[] args) {
     int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
     for (int i = 0; i < myNumbers.length; ++i) {
        for(int j = 0; j < myNumbers[i].length; ++j) {
            System.out.println(myNumbers[i][j]);
        }
     }
   }
}</pre>
```

Try it Yourself »

Test Yourself With Exercises

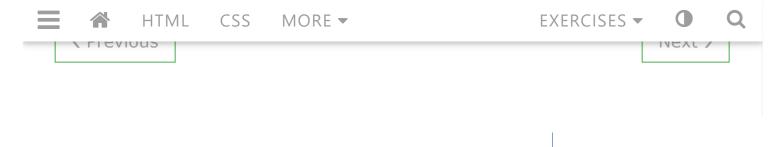
Exercise:

Create an array of type String called cars.

```
= {"Volvo", "BMW", "Ford"};
```

Submit Answer »

Start the Exercise



COLOR PICKER



LIKE US







Get certified by completing a course today!



Get started

CODE GAME





HTML CSS MORE ▼

EXERCISES ▼ **①**







Play Game

Certificates

HTML
CSS
JavaScript
Front End
Python
SQL
And more





HTML

CSS

MORE ▼

EXERCISES ▼



0

REPORT ERROR

FORUM

ABOUT

SHOP

Top Tutorials

HTML Tutorial

CSS Tutorial

JavaScript Tutorial

How To Tutorial

SQL Tutorial

Python Tutorial

W3.CSS Tutorial

Bootstrap Tutorial

PHP Tutorial

Java Tutorial

C++ Tutorial

jQuery Tutorial

Top References

HTML Reference

CSS Reference

JavaScript Reference

SQL Reference

Python Reference

W3.CSS Reference

Bootstrap Reference





HTML

CSS

MORE ▼

EXERCISES ▼





Java Kererence Angular Reference jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Web Courses

HTML Course CSS Course

JavaScript Course

Front End Course

SQL Course

Python Course

PHP Course

jQuery Course

Java Course

C++ Course

C# Course

XML Course

Get Certified »

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2021 by Refsnes Data. All Rights Reserved. W3Schools is Powered by W3.CSS.

