



FREE eBook

LEARNING arrays

Free unaffiliated eBook created from
Stack Overflow contributors.

#arrays

Table of Contents

About	1
Chapter 1: Getting started with arrays	2
Remarks.....	2
Examples.....	2
Availability.....	2
Basic array in Java.....	2
Chapter 2: primitive arrays and object arrays	3
Examples.....	3
Create a basic array in Java.....	3
Java create an array with initial values.....	3
Chapter 3: Various ways of accessing arrays	4
Examples.....	4
Iterating an Array with a For Loop in C.....	4
Iterating an Array with a While Loop in C.....	4
Iterating an Array with a Do While loop in C.....	4
Iterating an Array with a For Each loop in Java.....	4
Iterating an Array using recursion in C.....	4
Applying a Function to Every Value of an Array in Javascript.....	5
Calculate Single Value from Array in Javascript.....	5
Credits	6

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [arrays](#)

It is an unofficial and free arrays ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official arrays.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with arrays

Remarks

Arrays are constructs provided by most programming languages to access a group of same objects via an index. In some languages, the types must be the same (Java), while in others (JavaScript, Python) multiple types can be in an array.

Examples

Availability

Arrays are available in most programming languages, often using square `[]` or round `()` brackets to access the elements, e.g. `Carray[6]` or `VBarray(6)`.

Basic array in Java

In Java, any object or primitive type can be an array. Array indices are accessed via `arrayName[index]`, e.g. `myArray[0]`. Values in an array are set via `myArray[0] = value`, e.g. if `myArray` is an array of type `String[]` `myArray[0] = "test";`

```
public class CreateBasicArray{
    public static void main(String[] args){

        // Creates a new array of Strings, with a length of 1
        String[] myStringArray = new String[1];
        // Sets the value at the first index of myStringArray to "Hello World!"
        myStringArray[0] = "Hello World!";
        // Prints out the value at the first index of myStringArray,
        // in this case "Hello World!"
        System.out.println(myStringArray[0]);

        // Creates a new array of ints, with a length of 1
        int[] myIntArray = new int[1];
        // Sets the value at the first index of myIntArray to 1
        myIntArray[0] = 1;
        // Prints out the value at the first index of myIntArray,
        // in this case 1
        System.out.println(myIntArray[0]);

        // Creates a new array of Objects with a length of 1
        Object[] myObjectArray = new Object[1];
        // Constructs a new Java Object, and sets the value at the first
        // index of myObjectArray to the new Object.
        myObjectArray[0] = new Object();
    }
}
```

Read Getting started with arrays online: <https://riptutorial.com/arrays/topic/1000/getting-started-with-arrays>

Chapter 2: primitive arrays and object arrays

Examples

Create a basic array in Java

```
public class CreateAnArray{
    public static void main(String[] args){
        // Creates a new array of Strings with a length of 3
        // This length cannot be changed later
        String[] myStringArray = new String[3];
        myStringArray[0] = "Hello"; // Java array indices start at 0
        myStringArray[1] = "World";
        myStringArray[2] = "!"; // The array is now full
        try{
            myStringArray[3] = "This will cause an error."; // Index 3 requires an array
            // of size 4 or greater
        }
        catch(ArrayIndexOutOfBoundsException out){
            System.out.println("Java arrays cannot be expanded.");
        }
        // Print out "Hello World!" to the console
        System.out.println(myStringArray[0]+" "+myStringArray[1]+" "+myStringArray[2]);
    }
}
```

Java create an array with initial values

```
public class CreateArrayWithValues {
    public static void main(String[] args){
        // Initializes an array of Strings with values
        String[] myArray = {"this", "array", "has", "six", "initial", "values"};
        System.out.println("myArray.length = "+myArray.length);
        // Print out each value of myArray
        for(int index = 0; index < myArray.length; index++){
            System.out.print(myArray[index]);
        }
    }
}
```

Read primitive arrays and object arrays online: <https://riptutorial.com/arrays/topic/6866/primitive-arrays-and-object-arrays>

Chapter 3: Various ways of accessing arrays

Examples

Iterating an Array with a For Loop in C

```
int arr[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
int i;
for(i = 0; i < 10; i++)
{
    printf("%d\n", arr[i]);
}
```

Iterating an Array with a While Loop in C

```
int arr[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
int i = 0;
while(i < 10)
{
    printf("%d\n", arr[i]);
    i++;
}
```

Iterating an Array with a Do While loop in C

```
int arr[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
int i = 0;
do
{
    printf("%d\n", arr[i]);
    i++;
} while (i < 10);
```

Iterating an Array with a For Each loop in Java

```
int [] arr = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
for(int value : arr) {
    System.out.print(value);
    System.out.print("\n");
}
```

*Note that the Java foreach is just a for loop with different syntax. Some languages do this and some such as C# use foreach.

Iterating an Array using recursion in C

```
int sumArrayRecursive(int * arr, int index, int arraySize)
{
```

```
if (index == (arraySize - 1))
{
    return arr[index];
}
return arr[index] + sumArrayRecursive(arr, index + 1, arraySize);
}
```

Applying a Function to Every Value of an Array in Javascript

```
var numbers = [1,2,3,4,5];
var squares = numbers.map(function(x) {
    return x*x;
});

// squares is [1,4,9,16,25]
```

Calculate Single Value from Array in Javascript

```
var arr = [1, 2, 3, 4, 5];
var sum = arr.reduce((prev, curr) => prev + curr);
console.log(sum);
// Output: 15
```

You can also specify an initial value

```
var arr = [1, 2, 3, 4, 5];
var sum = arr.reduce(function (previousValue, currentValue, currentIndex, array) {
    return previousValue + currentValue;
}, 100);
console.log(sum)
// Output: 115
```

Read Various ways of accessing arrays online: <https://riptutorial.com/arrays/topic/4260/various-ways-of-accessing-arrays>

Credits

S. No	Chapters	Contributors
1	Getting started with arrays	Bryden , Community , larsAnders , Mark Hurd
2	primitive arrays and object arrays	Bryden
3	Various ways of accessing arrays	Rainbacon