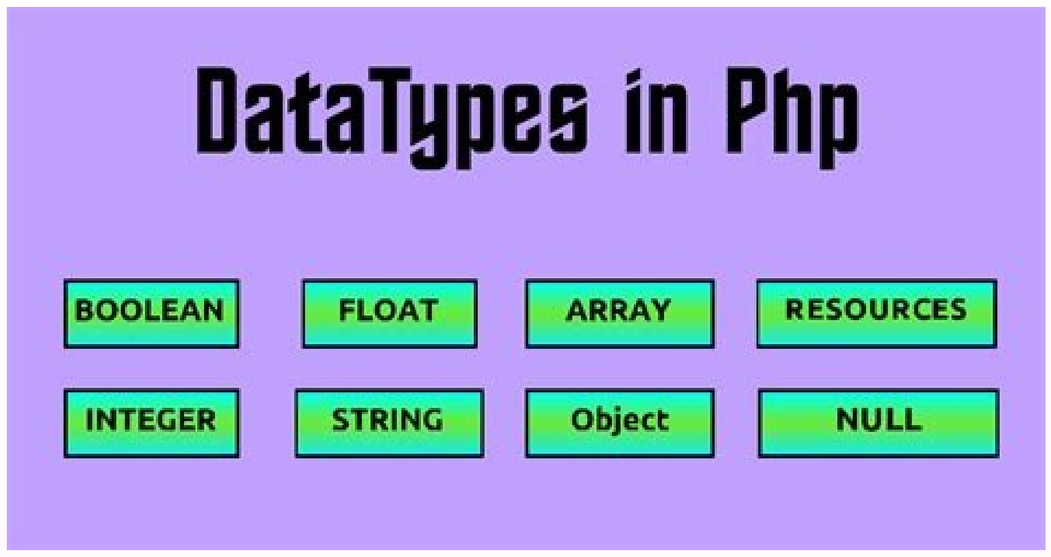
**Array in java notes pdf**

I'm not robot!




```c
/**
 * C program to check whether a number is
 * positive, negative or zero
 */

#include <stdio.h>

int main() {
    int number;
    /*
     * Take a number as input from user
     */
    printf("Enter a Number\n");
    scanf("%d", &number);

    if(number > 0) {
        printf("%d is Positive Number", number);
    } else if (number < 0) {
        printf("%d is Negative Number", number);
    } else {
        printf("Input Number is Zero");
    }

    return 0;
}

//Source: BTechGeeks.com
```

```java
// Java Program to Print Left Triangle Star Pattern

import java.util.*;
public class Main
{
   public static void main(String args[])
   {
   // taking variable for loop iteration and row .
   int i,row,c;
   //creating object
   Scanner s = new Scanner(System.in);
   // entering the number of row
   System.out.print("Enter row : ");
   row = s.nextInt();
   //iteration for no. of row
      for (i=0; i<row; i++)
      {
          //iteration for printing space
          for (c=2*(row-i); c>=0; c--)
              System.out.print(" ");
          //iteration for printing stars
          for (c=0; c<=i; c++ )
              System.out.print("* ");
          // printing new line .
          System.out.println();
      }
   }
}

// Source: BTechGeeks.com
```

Array in java lecture notes. Array in java class 10 icse notes. Array count in java. Array in java notes pdf. What is array in java with example. Array in java and its types. Arraylist in java notes. Write notes in javascript array.

An array is a collection of similar types of data. For example, if we want to store the names of 100 people then we can create an array of the string type that can store 100 names. String[] array = new String[100]; Here, the above array cannot store more than 100 names. The number of values in a Java array is always fixed. How to declare an array in Java? In Java, here is how we can declare an array. dataType[] arrayName; For example, double[] data; Here, data is an array that can hold values of type double. But, how many elements can array this hold? Good question! To define the number of elements that an array can hold, we have to allocate memory for the array in Java. For example, // declare an array double[] data; // allocate memory data = new double[10]; Here, the array can store 10 elements. We can also say that the size or length of the array is 10. In Java, we can declare and allocate the memory of an array in one single statement. For example, double[] data = new double[10]; How to Initialize Arrays in Java? In Java, we can initialize arrays during declaration. For example, //declare and initialize and array int[] age = {12, 4, 5, 2, 5}; Here, we have created an array named age and initialized it with the values inside the curly brackets. Note that we have not provided the size of the array. In this case, the Java compiler automatically specifies the size by counting the number of elements in the array (i.e. 5). In the Java array, each memory location is associated with a number. The number is known as an array index. We can also initialize arrays in Java, using the index number. For example, // declare an array int[] age = new int[5]; // initialize array age[0] = 12; age[1] = 4; age[2] = 5; .. Java Arrays initialization Note: Array indices always start from 0. That is, the first element of an array is at index 0. If the size of an array is n, then the last element of the array will be at index n-1. How to Access Elements of an Array in Java? We can access the element of an array using the index number. Here is the syntax for accessing elements of an array, // access array elements array[index] Let's see an example of accessing array elements using index numbers. Example: Access Array Elements class Main { public static void main(String[] args) { // create an array int[] age = {12, 4, 5, 2, 5}; // access each array elements System.out.println("Accessing Elements of Array:"); System.out.println("First Element: " + age[0]); System.out.println("Second Element: " + age[1]); System.out.println("Third Element: " + age[2]); System.out.println("Fourth Element: " + age[3]); System.out.println("Fifth Element: " + age[4]); } } Output Accessing Elements of Array: First Element: 12 Second Element: 4 Third Element: 5 Fourth Element: 2 Fifth Element: 5 In the above example, notice that we are using the index number to access each element of the array. We can use loops to access all the elements of the array at once. Looping Through Array Elements In Java, we can also loop through each element of the array. For example, Example: Using For Loop class Main { public static void main(String[] args) { // create an array int[] age = {12, 4, 5}; // loop through the array // using for loop System.out.println("Using for Loop:"); for(int i = 0; i < age.length; i++) { System.out.println(age[i]); } } } Output Using for Loop: 12 4 5 In the above example, we are using the for Loop in Java to iterate through each element of the array. Notice the expression inside the loop, age.length Here, we are using the length property of the array to get the size of the array. We can also use the for-each loop to iterate through the elements of an array. For example, Example: Using the for-each Loop class Main { public static void main(String[] args) { // create an array int[] age = {12, 4, 5}; // loop through the array // using for loop System.out.println("Using for-each Loop:"); for(int a : age) { System.out.println(a); } } } Output Using for-each Loop: 12 4 5 Example: Compute Sum and Average of Array Elements class Main { public static void main(String[] args) { int[] numbers = {2, -9, 0, 5, 12, -25, 22, 9, 8, 12}; int sum = 0; Double average; // access all elements using for each loop // add each element in sum for (int number: numbers) { sum += number; } // get the total number of elements int arrayLength = numbers.length; // calculate the average // convert the average from int to double average = ((double)sum / (double)arrayLength); System.out.println("Sum = " + sum); System.out.println("Average = " + average); } } Output: Sum = 36 Average = 3.6 In the above example, we have created an array of named numbers. We have used the for...each loop to access each element of the array. Inside the loop, we are calculating the sum of each element. Notice the line, int arrayLength = number.length; Here, we are using the length attribute of the array to calculate the size of the array. We then calculate the average using: average = ((double)sum / (double)arrayLength); As you can see, we are converting the int value into double. This is called type casting in Java. To learn more about typecasting, visit Java Type Casting. Multidimensional Arrays Arrays we have mentioned till now are called one-dimensional arrays. However, we can declare multidimensional arrays in Java. A multidimensional array is an array of arrays. That is, each element of a multidimensional array is an array itself. For example, double[][] matrix = {{1.2, 4.3, 4.0}, {4.1, -1.1} }; Here, we have created a multidimensional array named matrix. It is a 2-dimensional array. To learn more, visit the Java multidimensional array. Recommended Readings One of the most popular programming languages in the world, Java is an essential part of any web and application development professional's toolkit. While there are many components and concepts to understand this powerful language, in this article, we are going to talk about arrays in Java. Arrays are a straightforward yet essential concept of Java programming. Whether you are an experienced programmer or a beginner, you will inevitably use arrays in almost all aspects of Java programming.  An array refers to a data structure that contains homogeneous elements. This means that all the elements in the array are of the same data type. Let's take an example: This is an array of seven elements. All the elements are integers and homogeneous. The green box below the array is called the index, which always starts from zero and goes up to n-1 elements. In this case, as there are seven elements, the index is from zero to six. There are three main features of an array:  Dynamic

allocation: In arrays, the memory is created dynamically, which reduces the amount of storage required for the code.  Elements stored under a single name: All the elements are stored under one name. This name is used any time we use an array.  Occupies contiguous location: The elements in the arrays are stored at adjacent positions. This makes it easy for the user to find the locations of its elements.  Also Read: What is Java: A Beginners Guide To Java Java arrays enable you to access any element randomly with the help of indexes It is easy to store and manipulate large data sets  The size of the array cannot be increased or decreased once it is declared—arrays have a fixed size Java cannot store heterogeneous data. It can only store a single type of primitives Now that we understand what Java arrays are- let us look at how arrays in Java are declared and defined. Arrays in Java are easy to define and declare. First, we have to define the array. The syntax for it is:  Here, the type is int, String, double, or long. Var-name is the variable name of the array.  These are the two ways that you declare an array in Java. You can assign values to elements of the array like this: We have declared an array arr of type integer. The size of the array is 5, meaning that it can have five elements. The array is assigned with elements for each of the index positions. We'll run a for loop to print the elements in the array. A counter variable "i" is used to increment the index position after checking if the current index position is less than the length of the array.  After executing this program, the output that you will get using this array is as follows: Now that you know how to create and use an array in Java, let's look at the types of arrays. Get a firm foundation in Java, the most commonly used programming language in software development with the Java Certification Training Course. There are three types of arrays. We use these types of arrays as per the requirement of the program. These are: 1. One-dimensional Array Also known as a linear array, the elements are stored in a single row. For example:  In this example, we have an array of five elements. They are stored in a single line or adjacent memory locations. Look at this example in Java code. Here, the five elements are 1, 2, 3, 4, and 5. We use a for loop to print the elements of the array. The output of this is as follows:  2. Two-dimensional Array Two-dimensional arrays store the data in rows and columns: In this, the array has two rows and five columns. The index starts from 0,0 in the left-upper corner to 1,4 in the right lower corner. In this Java code, we have a two-dimensional array. We have two rows and three columns. Brackets separate the rows, and the number of elements separates the columns. For this, we use two for loops: one for rows and one for each element in the row. When we execute this program, the result will be as follows:  3. Multi-dimensional Array This is a combination of two or more arrays or nested arrays. We can even use more than two rows and columns using the following code:  Here, we are using three rows and three columns, but we are only using two for loops. Regardless of how many rows and columns are entered, the number of for loops will always be two.  Now that we know about the types of arrays, let us look at some examples below. /* Java Program to find the sum of all the elements in an array  */ The above program will add all the elements defined in my_array[] and produce the result. /* Java Program to find the sum of all the elements in an array  */ The above program will multiply all the elements defined in my_array[] and produce the result. You can copy one array to another by using Arrays.copyOf() method. The above piece of code will store the elements of the array "a" in the newly created array "b".  Java supports object cloning with the help of the clone() method to create an exact copy of an object. Learning binary search algorithms is recommended for a better understanding of arrays. All searching and sorting algorithms start with arrays. A binary search is an algorithm used to find the element in an array. It divides the array into half, and the elements in the array must be in ascending order. Then, the value of the element and the middle element of the array is compared. If the value of the element is smaller than the value of the mid element, the greater half of the array is neglected, and the search is conducted in the lower half with the same process.  For example:  We have an array of seven elements; suppose we have to find the element 11. The algorithm is applied, and we get the element with the index number.  The following is a Java code for binary search: This is the code for the binary search class.  The following is the code for the main() program. This program gets the elements of the array from the user and then asks the user to find the element in the array. The output of this program will be:  First, enter the elements in ascending order. Next, you have to put the element you want to search for and the result will be as follows:  After reading this article, you should understand the basics of arrays in Java. You learned how to declare arrays and how to assign values to them. Then you saw the three types of arrays with an example of each along with Java code. You looked at how to implement a program to add elements in an array, multiply the array elements, and how to copy and clone arrays. Finally, you got an idea to implement a binary search program that made the concept of array clear. To learn more about Java Arrays, watch this video. If you're ready to take your software development career to the next level, check out Simplilearn's Full Stack Java Developer training course. This comprehensive online Blended Learning program will teach you everything you need to know about the tools and techniques used in Java programming. Once you complete the mix of learning exercises, expert advice, and real-world industry projects, you will earn a certificate to help you land your dream job or get that promotion. What are you waiting for?