

Java Variable Types

Variable Types in Java

Variable Types in Java :- In this [Java Tutorial](#), we shall learn about Variable Types in Java, their scope etc.

What is a Variable?

Variable is a memory location with a name and can be manipulated by programs.

Following are the four kinds of variables in java :

- Non-static
- Static
- Local
- Parameters

Each variable in java has to be given a datatype during its declaration. The datatype of variable decides on :

- how much memory to be allocated for the variable
- range of values, the variable can hold
- default value to the variable

Types of Variable in Java

Let us understand different variable types in detail.

Static or Class variables

- These variables are declared inside the class and outside methods, **with the modifier “static”**.
- These are also called class variables, because all the instances of a class share the same variable (all instances of a class point to the same memory location of the static variable).
- If an instance modifies the static variable, the change is reflected in all other instances of the class. Hence the scope of these type of variables is the class itself.
- If it is desired that the value of a static variable should be kept constant, making instances incapable of changing the value, “**final**” modifier should be used during variable declaration.
- Any attempt to change the value of final static variable, results in compile error.

Non-static or Instance variables

- These variables are declared inside the class and outside methods, **without the modifier “static”**.
- These are also called instance variables, because the variable value could differ from one instance to another instance of the same class. Hence, the variable value is unique to an instance of class, not the class itself, unlike class variables.
- Any change in the value of non-static variable, the change is confined only to that instance in which the change happened. It doesn't affect other instances of the class. Hence the scope of these type of variables is the entire instance of a class.
- As these type of variable values are unique to an instance, they define the state of the object/instance of the class.

Local variables

- These variables are declared inside methods.
- Even inside a method, the scope of local variable could be confined to a specific block.

Parameters

- These variables are declared in the signature of a method or constructor or even in exception handlers(catch blocks).

Let us understand about the different variable types and their behaviors with an example.

Example – Java Variables

Following is an example demonstrating the scope and behavior of variable types with the help of classes: VariableTypesExample.java and Car.java

Car.java

```
package com.tutorialkart.java;
/**
 * @author tutorialkart.com
 */
class Car{
    final static int noOfWheels = 4;    // static variable with modifier final
    static String ownerName = "Tutorialkart";    // static variable
    String carName; // non-static variable
    int topSpeed;    // non-static variable
    // carName and topSpeed together define the state of an instance of this Car class
    public Car(int speed, String name){ // speed and name are parameters
        topSpeed = speed;
        carName = name;
    }
    public void horn(){
        int frontCarDistance=5; // scope of frontCarDistance variable is horn() method
        { // scope of noOfHorns is this first block
            int noOfHorns = 2;
        }
    }
}
```

```

        int noOfHorns = 2;
        System.out.println(carName + " horns "+noOfHorns+" times.");
    }
    { // scope of noOfHorns is this second block
        int noOfHorns = 3;
        System.out.println(carName + " horns "+noOfHorns+" times.");
    }
    System.out.println("For "+carName+", frontCarDistance value is :"+frontCarDistance);
}
}

```

Example.java

```

package com.tutorialkart.java;
/**
 * @author tutorialkart.com
 */
public class Example {
    static int var = 10;
    public static void main(String[] args) {
        Car carOne = new Car(120, "carOne");
        Car carTwo = new Car(185, "carTwo");

        System.out.println("-----carOne's details-----");
        System.out.println(carOne.carName + "'s top speed : "+carOne.topSpeed);
        carOne.horn();
        System.out.println(carOne.carName + " has "+carOne.noOfWheels+" wheels.");
        System.out.println(carOne.carName + "'s owner is "+carOne.ownerName);
        carOne.ownerName = "Tutorialkart Modified";
        System.out.println("-----\n");

        System.out.println("-----carTwo's details-----");
        System.out.println(carTwo.carName + "'s top speed : "+carTwo.topSpeed);
        carTwo.horn();
        System.out.println(carTwo.carName + " has "+carTwo.noOfWheels+" wheels.");
        // ownerName is changed in the instance of carOne, the change reflects in all the instances
        System.out.println(carTwo.carName + "'s owner is "+carTwo.ownerName);
        System.out.println("-----");
    }
}

```

When the above example program is run, the output to console is

```

-----carOne's details-----
carOne's top speed : 120
carOne horns 2 times.
carOne horns 3 times.
For carOne, frontCarDistance value is :5
carOne has 4 wheels.
carOne's owner is Tutorialkart
-----

-----carTwo's details-----
carTwo's top speed : 185
carTwo horns 2 times.
carTwo horns 3 times.
For carTwo, frontCarDistance value is :5

```

```
For carTwo, frontCarDistance value is 1.5  
carTwo has 4 wheels.  
carTwo's owner is Tutorialkart Modified  
-----
```

Conclusion

In this [Java Tutorial](#), we have learnt about Types of Variables in Java, in-detail with the help of an example. In our next tutorial, we shall learn [Access Modifiers in Java](#).

Java Tutorial

- ◆ [Java Tutorial](#)
- ◆ [Java Introduction](#)
- ◆ [Java Installation](#)
- ◆ [IDEs for Java Development](#)
- ◆ [Java HelloWorld Program](#)
- ◆ [Java Program Structure](#)
- ◆ [Java Datatypes](#)
- ⇒ **[Java Variable Types](#)**
- ◆ [Java Access Modifiers](#)
- ◆ [Java Operators](#)
- ◆ [Java Decision Making](#)
- ◆ [Java Loops](#)
- ◆ [Java Array](#)
- ◆ [Java OOPs](#)
- ◆ [Java String](#)
- ◆ [Java Exception Handling](#)
- ◆ [Java File Operations](#)
- ◆ [Java Date & Time](#)
- ◆ [Java MySQL](#)
- ◆ [Java Random](#)
- ◆ [Java Math](#)