# Java Variables

## Java Variables

Variables are containers for storing data values.

In Java, there are different ****types**** of variables, for example:

* String - stores text, such as "Hello". String values are surrounded by double quotes
* int - stores integers (whole numbers), without decimals, such as 123 or -123
* float - stores floating point numbers, with decimals, such as 19.99 or -19.99
* char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
* boolean - stores values with two states: true or false

## Declaring (Creating) Variables

To create a variable, you must specify the type and assign it a value:

### Syntax

*type variable = value;*

Where type is one of Java's types (such as int or String), and variable is the name of the variable (such as ****x**** or ****name****). The **equal sign** is used to assign values to the variable.

To create a variable that should store text, look at the following example:

### Example

Create a variable called ****name**** of type String and assign it the value "****John****":

String name = "John";System.out.println(name);

To create a variable that should store a number, look at the following example:

### Example

Create a variable called ****myNum**** of type int and assign it the value ****15****:

int myNum = 15;

System.out.println(myNum);

You can also declare a variable without assigning the value, and assign the value later:

### Example

int myNum;

myNum = 15;System.out.println(myNum);

Note that if you assign a new value to an existing variable, it will overwrite the previous value:

### Example

Change the value of myNum from 15 to 20:

int myNum = 15;

myNum = 20; // myNum is now 20System.out.println(myNum);

## Final Variables

However, you can add the final keyword if you don't want others (or yourself) to overwrite existing values (this will declare the variable as "final" or "constant", which means unchangeable and read-only):

### Example

final int myNum = 15;

myNum = 20; // will generate an error: cannot assign a value to a final variable

## Other Types

A demonstration of how to declare variables of other types:

### Example

int myNum = 5;float myFloatNum = 5.99f;char myLetter = 'D';boolean myBool = true;String myText = "Hello";

## Integer Types

### Byte

The byte data type can store whole numbers from -128 to 127. This can be used instead of int or other integer types to save memory when you are certain that the value will be within -128 and 127:

### Example

byte myNum = 100;System.out.println(myNum);

### Short

The short data type can store whole numbers from -32768 to 32767:

### Example

short myNum = 5000;System.out.println(myNum);

### Int

The int data type can store whole numbers from -2147483648 to 2147483647. In general, and in our tutorial, the int data type is the preferred data type when we create variables with a numeric value.

### Example

int myNum = 100000;System.out.println(myNum);

### Long

The long data type can store whole numbers from -9223372036854775808 to 9223372036854775807. This is used when int is not large enough to store the value. Note that you should end the value with an "L":

### Example

long myNum = 15000000000L;System.out.println(myNum);

## Floating Point Types

You should use a floating point type whenever you need a number with a decimal, such as 9.99 or 3.14515.

### Float

The float data type can store fractional numbers from 3.4e−038 to 3.4e+038. Note that you should end the value with an "f":

### Example

float myNum = 5.75f;System.out.println(myNum);

### Double

The double data type can store fractional numbers from 1.7e−308 to 1.7e+308. Note that you should end the value with a "d":

### Example

double myNum = 19.99d;System.out.println(myNum);

## Java Data Types

As explained in the previous chapter, a variable in Java must be a specified data type:

### Example

int myNum = 5; // Integer (whole number)float myFloatNum = 5.99f; // Floating point numberchar myLetter = 'D'; // Characterboolean myBool = true; // BooleanString myText = "Hello"; // String

Data types are divided into two groups:

* Primitive data types - includes byte, short, int, long, float, double, boolean and char
* Non-primitive data types - such as [String](https://www.w3schools.com/java/java_strings.asp), [Arrays](https://www.w3schools.com/java/java_arrays.asp) and [Classes](https://www.w3schools.com/java/java_classes.asp) (you will learn more about these in a later chapter)

## Primitive Data Types

A primitive data type specifies the size and type of variable values, and it has no additional methods.

There are eight primitive data types in Java:

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Size** | **Description** |
| byte | 1 byte | Stores whole numbers from -128 to 127 |
| short | 2 bytes | Stores whole numbers from -32,768 to 32,767 |
| int | 4 bytes | Stores whole numbers from -2,147,483,648 to 2,147,483,647 |
| long | 8 bytes | Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| float | 4 bytes | Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits |
| double | 8 bytes | Stores fractional numbers. Sufficient for storing 15 decimal digits |
| boolean | 1 bit | Stores true or false values |
| char | 2 bytes | Stores a single character/letter or ASCII values |

## Numbers

Primitive number types are divided into two groups:

****Integer types**** stores whole numbers, positive or negative (such as 123 or -456), without decimals. Valid types are byte, short, int and long. Which type you should use, depends on the numeric value.

****Floating point types**** represents numbers with a fractional part, containing one or more decimals. There are two types: float and double.

## **Types of Non-primitive Data types in Java**

There are five types of non-primitive data types in Java. They are as follows:  
1. Class  
2. Object  
3. String  
4. Array  
5. Interface

****1. Class and objects:**** Every class is data type and it is also considered as user-defined data types. This is because a user creates a class. For more details: [Class and objects in java](https://www.scientecheasy.com/2018/07/class-obejcts-java-real-time-examples.html/)

****2. String:**** A string represents a sequence of characters like India, ABC123, etc. The simplest way to create a string object is by storing sequence of characters into string type variable like this:  
String str = “Universe”;  
Here, string type variable str contains “Universe”. A string is also a class. For more details: [String in Java](https://www.scientecheasy.com/2020/06/java-string-tutorial.html/).

****3. Array:**** An array in java is an object which is used to store multiple variables of the same type. These variables can be primitive or non-primitive data type.  
The example of declaring an array variable of primitive data type int is as follows:  
int [ ] scores;

The example of declaring an array variable of non-primitive data type is  
Student [ ] students; // Student is a name of class.  
You will learn more details in further tutorials.

****4. Interface:**** An interface is declared like a class but the only difference is that it contains only final variables and method declarations. It is a fully abstract class.

Here, we have given just basic knowledge of non-primitive data types in java. You will get more knowledge in further tutorials.