Java Full Course

By TechDive With Gaurav

Java Variables

• A variable is a container which holds the value while the Java program is executed. A variable is assigned with a data type.

• A variable is the name of a reserved area allocated in memory. In other words, it is a name of the memory location. It is a combination of "vary + able" which means its value can be changed.

• Variable is a name of memory location. There are three types of variables in java: local, instance and static.

- Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in the memory.

```
Data-type variable = [ value]

Example: - int a=10;
```

Declaring a Variable

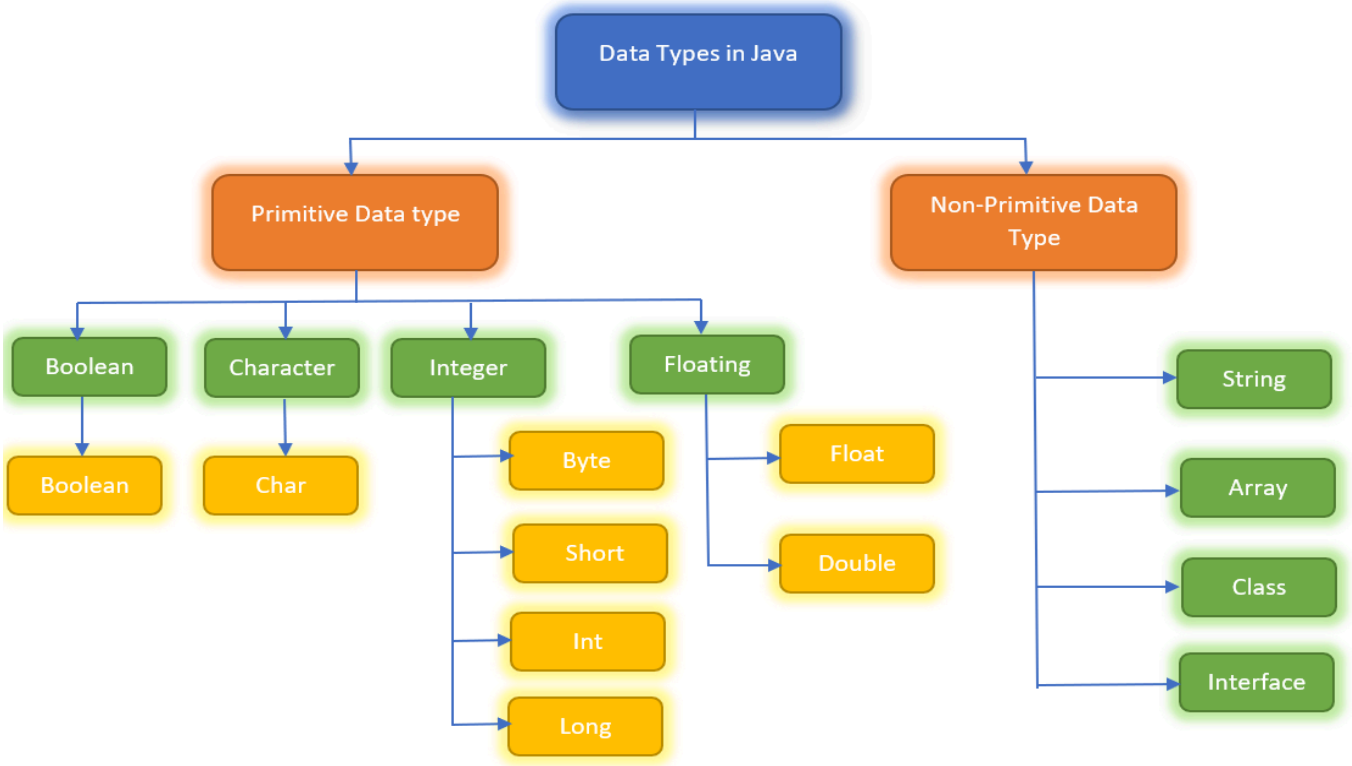
1. Must not being with a digit (1Rahul is invalid)
2. Name of variable is case sensitive (Rohit and rohit are different)
3. It shouldn't be a keyword (like Super)
4. Space is not allowed (rohit verma=10; is invalid) **(rohitVerma = 10;)**
5. It can contain alphabets, $ character, _ character and digits if they all condition will come true.

Data Types in Java

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java:

1. Primitive data types: The primitive data types include Boolean, char, byte, short, int, long, float and double.

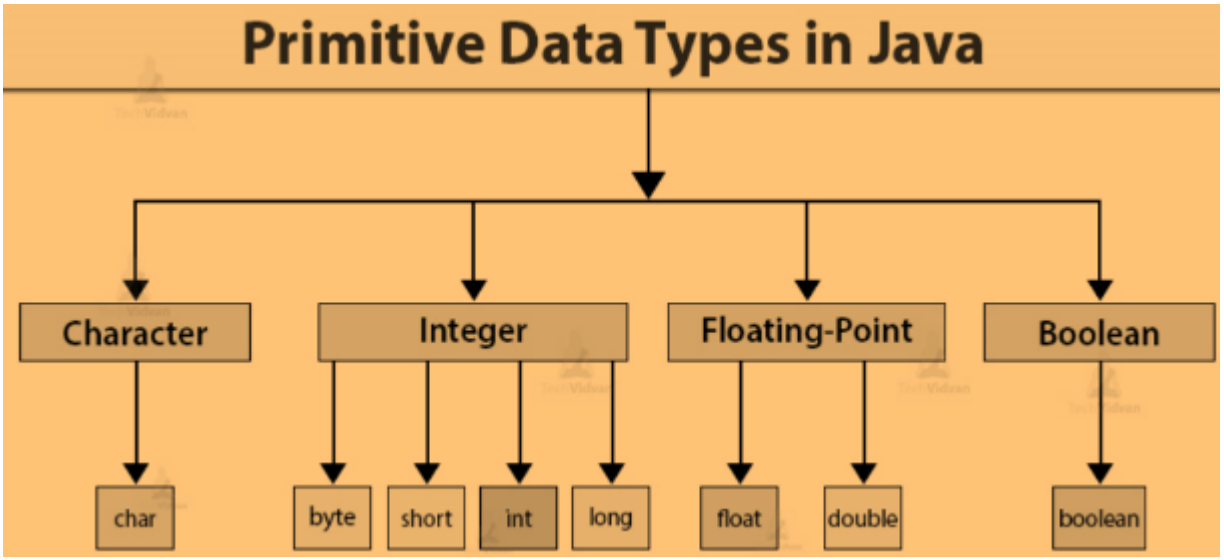
2. Non-primitive data types: The non-primitive data types include Classes, Interfaces, and Arrays.



Java Primitive Data Types

Java is a statically-typed programming language. It means, all variables must be declared before its use. That is why we need to declare variable's type and name.

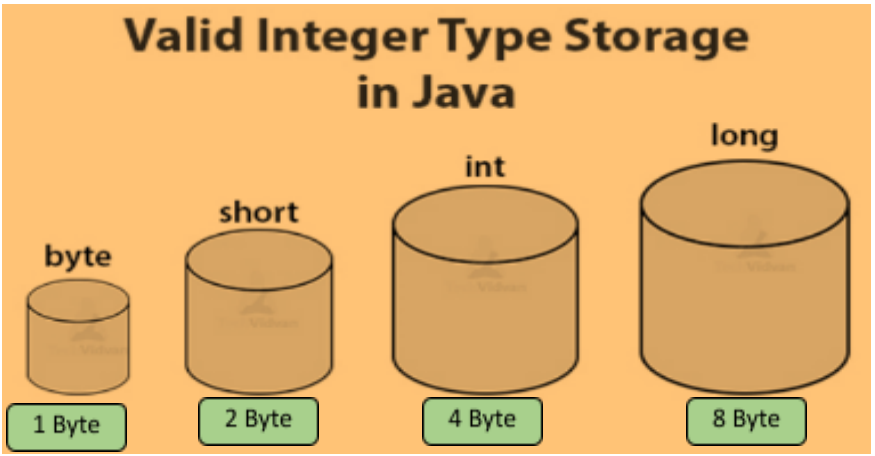
There are eight primitive datatypes supported by Java. Primitive datatypes are predefined by the language and named by a keyword.



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

Java Integers

Integer type stores whole numbers that may be positive or negative and should not contain any decimal places. Valid Integer types are – byte, short, int, and long. What type will be taken depends on the value or range of the variable.



1. Byte

Syntax:

byte myByte1 = -100;
byte myByte2 = 25;

• The byte data type is mostly used in the large arrays which need memory savings. It saves memory as it is 4 times smaller than an int (integer). Sometimes, it can also be used in place of the “int” data type, when the value is very small.

Size:	Values:	Default Value:	Range:
1 byte (8 bits)	Positive or negative whole numbers.	0	-128(-2^7) to 127 (inclusive)(2^7 -1)

```
public class ByteDataType
{
        public static void main(String args[]){
                byte myByte1 = 127;
                byte myByte2 = -48;
                System.out.println("Byte 1: " +myByte1);
                System.out.println("Byte 2: " +myByte2);
        }
}
```

OUTPUT
Byte 1: 127
Byte 2: -48

2. Short

Syntax:

short myShort = 6000;

• Similar to the byte data type, a short data type is also used to save memory in large arrays. The short data type is a 16-bit signed 2’s complement integer. It is half of an int (integer).

Size:	Values:	Default Value:	Range:
2 bytes (16 bits)	Positive or negative whole numbers.	0	-32768(-2^15) to 32767 (inclusive) (2^15 -1)

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

                short myShort = 6000;
                System.out.println("myShort: " + myShort);

        }

}
```

OUTPUT

myShort:
6000

3. int

Syntax:

int myNum = 700000 ;

• Generally, we prefer to use int data type for integral values unless if there is no issue with memory. The int data type is a 32-bit integer.

Size:	Values:	Default Value:	Range:
4 bytes (32 bits)	Positive or negative whole numbers.	0	– 2,147,483,648 (-2^31) to – 2,147,483,648 (inclusive) (2^31 -1)

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

                int myNum1 = -7000;
                int myNum2 = 90000;
                System.out.println("Number 1: "
+myNum1);
                System.out.println("Numb
+myNum2);
        }
}
```

OUTPUT

Number 1:
-7000
Number 2:
90000

4. long

Syntax:

long myLong = 11000000000L;

• The long data type is a 64-bit signed 2's complement integers. It is used when int data type cannot hold a value. It is 2 times larger than int(integer). We must use L at the end of the value.

Size:	Values:	Default Value:	Range:
8 bytes (64 bits)	Positive or negative whole numbers.	0	-9,223,372,036,854,775,808(-2^63) to 9,223,372,036,854,775,807 (inclusive)(2^63 -1)

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

                myLong1 = 100000000L;
                System.out.println("Number 1: "
+myLong1);
        }

}
```

OUTPUT

Number 1:
100000000

5. Float

Syntax:

float f = 334.334f;

• The float data type is a single-precision 32-bit IEEE 754 floating point. Its value range is unlimited. It is recommended to use a float (instead of double) if you need to save memory in large arrays of floating pointnumbers. The float data type should never be used for precise values, such as currency. Its default value is 0.0F.

Size:	Values:	Default Value:	Range:
4 bytes (32 bits)	Positive or negative whole numbers.	0.0f	3.40282347 x 10^{38}, 1.40239846 x 10^{45} Accuracy: - about 7 decimal digit

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

                float myNum = 5.7335f;
                System.out.println("Float number
:"+myNum);
        }
}
```

OUTPUT

myShort:
6000

OUTPUT

myShort:
6000

6. Double

Syntax:

double d = 334.3343433;

• The double data type is a double-precision 64-bit IEEE 754 floating point. Its value range is unlimited. The double data type is generally used for decimal values just like float. The double data type also should never be used for precise values, such as currency.

Size:	Values:	Default Value:	Range:
8 bytes (64 bits)	Positive or negative whole numbers.	0.0d	1.7E+308 to +1.7E+308 Accuracy:- about 16 decimal digit

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

                double myNum = 5.73353434;
                System.out.println("Double number
:"+myNum);
        }
}
```

7. char

Syntax:

char ch = ‘d’;

• The char data type is a single 16-bit Unicode character. Its value-range lies between '\u0000' (or 0) to '\uffff' (or 65,535 inclusive).The char data type is used to store characters.

Size:	Values:	Default Value:	Range:
2 bytes (16 bits)	One character (or number in char form).	0	From \u0000(0) to \uffff(65535)

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

        char ch = ‘f’;
        System.out.println(“Value of character
variable:”+ch);
        }
}
```

OUTPUT

Value of character variable : f

8. Boolean

Syntax:

boolean val = true;

• The Boolean data type is used to store only two possible values: true and false. This data type is used for simple flags that track true/false conditions.

• The Boolean data type specifies one bit of information, but its "size" can't be defined precisely

Size:	Values:	Default Value:	Range:
1 bit	True or False	False	True or False

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

                boolean b = true;
                System.out.println(“Value of boolean
variable:”+b);
        }
}
```

OUTPUT

Value of boolean variable : true

Non-Primitive Data Types

There are five types of non-primitive data types in Java.

• **Class** : A class in Java is a user defined data type i.e. it is created by the user. It acts a template to the data which consists of member variables and methods.

• **Object** : An object is the variable of the class, which can access the elements of class i.e. methods and variables.

• **String** : It is used to store string type values(sequence of characters).

• **Array** : An array is a data type which can store multiple homogenous variables i.e., variables of same type in a sequence

• **Interface** : An interface is similar to a class however the only difference is that its methods are abstract by default i.e. they do not have body. An interface has only the final variables and method declarations.

Types of Variables

There are three types of variables in Java:

•local variable

•instance variable (Non-static Variables)

• static variable (Class Variables)

Types of Variables

1) Local Variable

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class are not even aware that the variable exists. A local variable cannot be defined with "static" keyword.

2) Instance Variable (Non-static Variables)

A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as static. It is called an instance variable because its value is instance-specific and is not shared among instances.

3) Static variable (Class Variables)

A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory.