**Variable**

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.

**Types of Variables**

There are three types of variables:

* local variable
* instance variable
* static variable

**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 aren't even aware that the variable exists.

A local variable cannot be defined with "static" keyword.

**2) Instance Variable**

A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as [static](https://www.tpointtech.com/static-keyword-in-java).

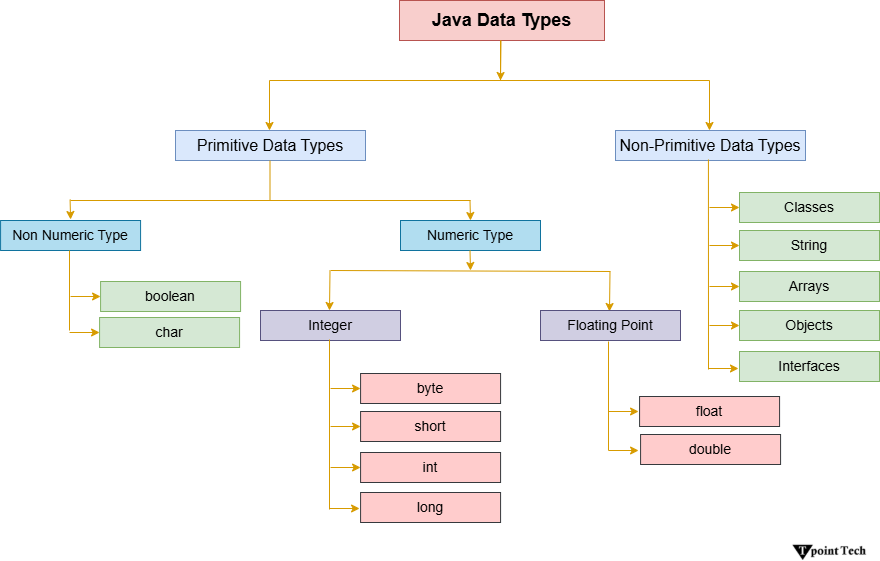
It is called an instance variable because its value is instance-specific and is not shared among instances.

**3) Static variable**

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.

**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, String, and Arrays.



**Java Primitive Data Types:**

In Java, primitive data types are the building blocks of data manipulation. These are the basic data types.

In Java, there are mainly eight primitive data types which are as follows.

1. [boolean data type](https://www.tpointtech.com/java-data-types#boolean)
2. [char data type](https://www.tpointtech.com/java-data-types#char)
3. [byte data type](https://www.tpointtech.com/java-data-types#byte)
4. [short data type](https://www.tpointtech.com/java-data-types#short)
5. [int data type](https://www.tpointtech.com/java-data-types#int)
6. [long data type](https://www.tpointtech.com/java-data-types#long)
7. [float data type](https://www.tpointtech.com/java-data-types#float)
8. [double data type](https://www.tpointtech.com/java-data-types#double)

1. Boolean Data Type

Example

Boolean a = false;

Boolean b = true;

[Compile and Run](https://www.tpointtech.com/java/tryit?program=java-data-types-1)

2. Byte Data Type

Example

byte a = 10;

byte b = -20;

3. Short Data Type

Example

short a = 10000;

short b = -5000;

4. int Data Type

Example

int a = 100000;

int b = -200000;

5. long Data Type

long num = 15000000000L;

long num = 9,223,372,036,854,775l

long a = 5000000L;

long b = -6000000L;

6. float Data Type

float num = 67;

float f = 234.5f;

7. double Data Type

double num = 75.658d;

double d = 12.3;

8. char Data Type

char alphabets = 'J';

char a = 60, b = 61, c = 62;

char c = 'A';

**Non-Primitive Data Types in Java**

In Java, non-primitive data types are also known as reference data types. It is used to store complex objects rather than simple values. Reference data types store references or memory addresses that point to the location of the object in memory. This distinction is important because it affects how these data types are stored, passed, and manipulated in Java programs.

1. Class

One common non-primitive data type in Java is the class. Classes are used to create objects, which are instances of the class. A class defines the properties and behaviours of objects, including variables (fields) and methods.

2. Interface

Interfaces are another important non-primitive data type in Java. An interface defines a contract for what a class implementing the interface must provide without specifying how it should be implemented. Interfaces are used to achieve abstraction and multiple inheritance in Java, allowing classes to be more flexible and reusable.

3. Arrays

Arrays are a fundamental non-primitive data type in Java that allows you to store multiple values of the same type in a single variable. Arrays have a fixed size, which is specified when the array is created and can be accessed using an index. Arrays are commonly used to store lists of values or to represent matrices and other multi-dimensional data structures.

4. String

In Java, a string is a sequence of characters. In simple words, we can define a string as an array of characters. The difference between a character array and a string in Java is that the string is designed to hold a sequence of characters in a single variable, whereas a character array is a collection of separate char-type entities. Note that, unlike C/C++, Java strings are not terminated with a null character.

5. enum

Java also includes other non-primitive data types, such as enums and collections. Enums are used to define a set of named constants, providing a way to represent a fixed set of values. Collections are a framework of classes and interfaces that provide dynamic data structures such as lists, sets, and maps, which can grow or shrink in size as needed.

Overall, non-primitive data types in Java are essential for creating complex and flexible programs. They allow us to create and manipulate objects, define relationships between objects, and represent complex data structures.