Java Variables

Variable

A variable is the name of a reserved area allocated in memory.

 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 in [Java](https://www.javatpoint.com/java-tutorial):

* local variable
* instance variable
* static variable

# Local Variables:

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.

 a Local variable can only be defined inside a method, the scope for the local variable is within the block inside which it gets defined.

Outside the block, we cannot even access the variable because we do not know in real whether it exists or not for us.

Local variables are useful when we need some temporary variable to hold the values for us inside the block

Variable\_type varibale\_name = value\_to\_hold;

String demo = ""my variable !!;

They act as temporary variables for us to use hold the vale and vanish when the particular piece of code stops or finishes its execution

In java, we cannot have a static local variable declaring inside a method that is not static here. Because static variables are directly associated with the class level.

In java, we can define the final local variable inside the method. We just need to have the final keyword associated with it. For reference, see the below code;

# 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.javatpoint.com/static-keyword-in-java).

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

Instances in the Java program are created whenever an object is called in a class or function, which uses a variable for the purpose of allocating a memory storage unit for every time an object is called, and an instance is created. Any changes or operations performed on the instances will reflect automatically in the instance variable and the corresponding memory unit. This type of variables are supported by several data types, namely int, char, long, short, boolean, float, double, byte and object.

**Syntax:**

<datatype> <variable\_name>;

<datatype> <variable\_name> = <initializing\_value>;

Instance variables are declared/defined in a class but outside the constructor (s), method(s) and block(s). Instance variables are called so because they’re instance(object) specific and are not shared among instances (objects)s, and changes done to variables of a particular instance will not reflect on others. When memory is allocated for an object in a heap, a slot for each of the instance variables is created. Instance variables of an object are created when the object is created with ‘new’ and are destroyed when the object is destroyed. Objects use them to preserve the states.

Instance variables have default values; therefore, they can be declared without initialization, unlike local variables. The default values for the variables depend on their data types.

|  |  |
| --- | --- |
| **Datatype** | **Default Value** |
| boolean | false |
| byte | 0 |
| short | 0 |
| int | 0 |
| long | 0L |
| char | u0000 |
| float | 0.0f |
| double | 0.0d |
| Object | null |

### Characteristics of Instance Variable in Java

* Instance variables are visible to the constructor, method(s) and block(s). It is usually recommended to declare them as private, but the users can change the level of visibility by declaring it with various access modifiers suitable for their program.
* Instance variables of a base class can be used by the subclasses depending on the access level given to the variable.
* Instance variables can be declared final to protect the constness of a declared variable. final instance variables are initialized while declaration and can’t be modified. They can also be declared and initialized in block and constructor.
* The instance variable can use the default access modifier.

# 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.

**static** **int** m=100;//static variable