Variables are used to store data and give that data a name.

They are needed in almost every coding language to make code generic.

Widely, variables are of two types,

- Local variables

- Global variables

The variables defined inside a function or method are called local variables.

The variables defined inside a class and outside the function/method are called global variables.

Example:

*Here:*

**myVariable** is declared outside the main function/method and inside the class DataTypesTutorial and hence is a global variable which can be accessed anywhere inside this class.

**rate** is declared inside the main function/method and hence is a local variable which can be accessed anywhere inside this main function/method.

Naming Conventions:

Java naming convention is a rule to follow as you decide what to name your identifiers such as class, package, variable, constant, method, etc. But, it is not forced to follow. So, it is known as convention not rule. These conventions are suggested by several Java communities. All the classes, interfaces, packages, methods and fields of Java programming language are given according to the Java naming convention. If you fail to follow these conventions, it may generate confusion or erroneous code.

Java follows camel-case syntax for naming the class, interface, method, and variable.

If the name is combined with two words, the second word will start with uppercase letter always such as actionPerformed(), firstName, ActionEvent, ActionListener, etc.

All identifiers should begin with a letter (A to Z or a to z), currency character ($) or an underscore (\_).

After the first character, identifiers can have any combination of characters.

A keyword cannot be used as an identifier.

Most importantly, identifiers are case sensitive.

Examples of legal identifiers: age, $salary, \_value, \_\_1\_value.

Examples of illegal identifiers: 123abc, -salary.