Introduction to Variables in Java

Variables are one of the most fundamental concepts in any programming language. In Java, variables allow you to store data and use it throughout your program.

What is a Variable?

A **variable** is a container that holds data that can be changed during the execution of a program. Each variable in Java has a **name** (identifier), a **type**, and a **value**.

Key Elements of a Variable:

1. **Type**: Defines the kind of data the variable can hold (e.g., int for integers, double for floating-point numbers).
2. **Name**: The identifier you use to refer to the variable in your code.
3. **Value**: The data stored in the variable.

Declaring a Variable

In Java, to create a variable, you need to declare its **type** and **name**. The syntax for declaring a variable is:

Datatype variableName;

For example:

int number;

Here, int is the type, and number is the variable name.

Assigning a Value to a Variable

Once declared, you can assign a value to a variable using the assignment operator =:

number = 10;

Or you can combine declaration and assignment in one step:

int number = 10;

Types of Variables

Java supports different types of variables, classified based on their data type and scope.

1. Primitive Data Types

Primitive types are the most basic data types in Java.

Data Type	Description	Example
int	Stores integers (whole numbers)	int x = 5;
double	Stores decimal numbers	double y = 3.14;
char	Stores single characters	char c = 'A';
boolean	Stores true or false values	boolean b = true;

2. Reference Data Types

These refer to objects and are used to store the address (reference) of the object in memory. Common reference types include classes and arrays.

String name = "John"; // String is a reference type

Variable Scope

The scope of a variable refers to the part of the program where the variable can be accessed.

1. Local Variables

- Declared inside a method.
- Accessible only within the method where they are defined.

```
public void myMethod() {
   int localVar = 10;  // local variable
}
```

2. Instance Variables

- Declared inside a class but outside any method.
- Accessible by all methods of the class.

```
public class MyClass {
   int instanceVar = 20;  // instance variable
}
```

3. Static Variables (Class Variables)

- Declared using the static keyword.
- Shared among all instances of the class.

```
public class MyClass {
   static int staticVar = 30;  // static variable
```

}

Rules for Naming Variables

- **Letters, numbers, underscore (_) and dollar sign ($)** are allowed.
- The variable name must **start with a letter** (or underscore/dollar sign).
- **No spaces** are allowed.
- Variable names are **case-sensitive** (myVar and MyVar are different).
- Cannot use **reserved words** (e.g., int, class).

Example of valid variable names:

```
int myNumber;
double piValue;
boolean isAvailable;
```

Example of Variable Declaration and Use

```
public class Main {
   public static void main(String[] args) {
      // Declare and initialize a variable
      int number = 100;

      // Print the value of the variable
      System.out.println("The number is: " + number);

      // Change the value of the variable
      number = 200;

      // Print the new value
      System.out.println("The new number is: " + number);
   }
}
```