Variable in Java

Variables

Variables store data for processing.
A variable is given a name (or **identifier**), such as area, age, height, and the like. The name uniquely identifies each variable, assigning a value to the variable and retrieving the value stored.

Variables have **types**. Some examples:
- **int**: for integers (whole numbers) such as 123 and -456
- **double**: for floating-point or real numbers with optional decimal points and fractional parts in fixed or scientific notations, such as 3.1416, -55.66.
- **String**: for texts such as "Hello" or "Good Morning!". Text strings are enclosed within double quotes.

You can declare a variable of a type and assign it a value. Example:

```
String name = "David";
```

This creates a variable called **name** of type **String**, and assigns it the value "David".

> It is important to note that a variable is associated with a type, and is only capable of storing values of that particular type. For example, an **int** variable can store integer values, such as 123; but it cannot store real numbers, such as 12.34, or texts, such as "Hello".

Variables

Examples of variable declarations:

```
class MyClass {
  public static void main(String[ ] args) {
    String name ="David";
    int age = 42;
    double score =15.9;
    char group = 'Z';
  }
}
```

Try It Yourself

char stands for character and holds a single character.

Another type is the Boolean type, which has only two possible values: **true** and **false**.
This data type is used for simple flags that track true/false conditions.
For example:

```
boolean online = true;
```

You can use a comma-separated list to declare more than one variable of the specified type. Example: int a = 42, b = 11;

The Math Operators

Java provides a rich set of operators to use in manipulating variables. A value used on either side of an operator is called an **operand**.
For example, in the expression below, the numbers 6 and 3 are operands of the plus operator:

```
int x = 6 + 3;
```

Java arithmetic operators:
+ addition
- subtraction
*** multiplication**
/ division
% modulo

> Arithmetic operators are used in mathematical expressions in the same way that they are used in algebraic equations.

Addition

The + operator adds together two values, such as two constants, a constant and a variable, or a variable and a variable. Here are a few examples of addition:

```
int sum1 = 50 + 10;
int sum2 = sum1 + 66;
int sum3 = sum2 + sum2;
```

Subtraction

The - operator subtracts one value from another.

```
int sum1 = 1000 - 10;
int sum2 = sum1 - 5;
int sum3 = sum1 - sum2;
```

Multiplication

The * operator multiplies two values.

```
int sum1 = 1000 * 2;
int sum2 = sum1 * 10;
int sum3 = sum1 * sum2;
```

Division

The / operator divides one value by another.

```
int sum1 = 1000 / 5;
int sum2 = sum1 / 2;
int sum3 = sum1 / sum2;
```

In the example above, the result of the division equation will be a whole number, as **int** is used as the data type. You can use **double** to retrieve a value with a decimal point.

Modulo

The **modulo** (or remainder) math operation performs an integer division of one value by another, and returns the remainder of that division.
The operator for the modulo operation is the percentage (%) character.
Example:

```
int value = 23;
int res = value % 6; // res is 5
```

Try It Yourself

Dividing 23 by 6 returns a quotient of 3, with a remainder of 5. Thus, the value of 5 is assigned to the **res** variable.