**Java :**
### **Java Strings:**

In Java, a `String` is a sequence of characters used to represent text. Strings are widely used for storing and manipulating textual data. Java provides a rich set of methods and operations to work with strings.

Here's how you can declare and manipulate strings in Java:

```java
public class StringExample {
    public static void main(String[] args) {
        // Creating strings
        String message = "Hello, Java!";
        String name = "Alice";

        // Concatenation
        String greeting = "Hello, " + name; // Result: "Hello, Alice"

        // Length of a string
        int length = message.length(); // Result: 12

        // Accessing characters
        char firstChar = message.charAt(0); // Result: 'H'

        // Substring
        String substring = message.substring(7); // Result: "Java!"

        // String comparison
        boolean isEqual = message.equals("Hello, Java!"); // Result: true

        // String manipulation
        String upperCase = message.toUpperCase(); // Result: "HELLO, JAVA!"
        String lowerCase = message.toLowerCase(); // Result: "hello, java!"

        // String searching
        boolean containsJava = message.contains("Java"); // Result: true

        // Replace characters
        String replaced = message.replace('o', '0'); // Result: "Hell0, Java!"

        // Splitting strings
        String[] words = message.split(" "); // Result: ["Hello,", "Java!"]
    }
```

}
```

### **Java Math:**

The `java.lang.Math` class provides various static methods for mathematical operations. It includes methods for common mathematical functions, constants, and more complex calculations.

Here are some examples of using the `Math` class:

```java
public class MathExample {
    public static void main(String[] args) {
        // Basic arithmetic operations
        double sum = Math.add(5, 3); // Result: 8.0
        double difference = Math.subtract(10, 5); // Result: 5.0
        double product = Math.multiply(4, 6); // Result: 24.0
        double quotient = Math.divide(20, 4); // Result: 5.0

        // Trigonometric functions
        double sinValue = Math.sin(Math.toRadians(30)); // Result: 0.5
        double cosValue = Math.cos(Math.toRadians(60)); // Result: 0.5

        // Exponential and logarithmic functions
        double power = Math.pow(2, 3); // Result: 8.0
        double squareRoot = Math.sqrt(25); // Result: 5.0
        double logarithm = Math.log(10); // Natural logarithm

        // Constants
        double pi = Math.PI;
        double e = Math.E;
    }
}
```

### **Java Booleans:**

A `boolean` is a data type that represents a binary value, typically denoting `true` or `false`. Booleans are used in conditional statements and logic operations.

Here's how you can use booleans in Java:

```java

```
public class BooleanExample {
    public static void main(String[] args) {
        boolean isJavaFun = true;
        boolean isCodingHard = false;

        // Conditional statements
        if (isJavaFun) {
            System.out.println("Java is fun!");
        } else {
            System.out.println("Java is not fun.");
        }

        // Logical operations
        boolean resultAND = isJavaFun && isCodingHard; // Result: false
        boolean resultOR = isJavaFun || isCodingHard; // Result: true
        boolean resultNOT = !isJavaFun; // Result: false
    }
}
```

In this example, we declare and use boolean variables, demonstrate conditional statements based on boolean values, and show logical operations such as AND, OR, and NOT.