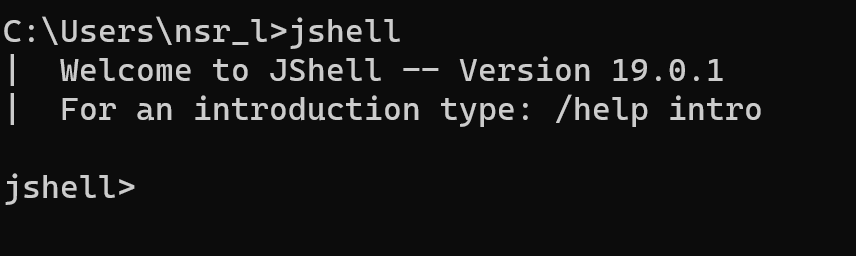
**Java 17 Masterclass: Start Coding in 2023**

**Section 1: Getting Started**

**Section 2: Programming Tools**

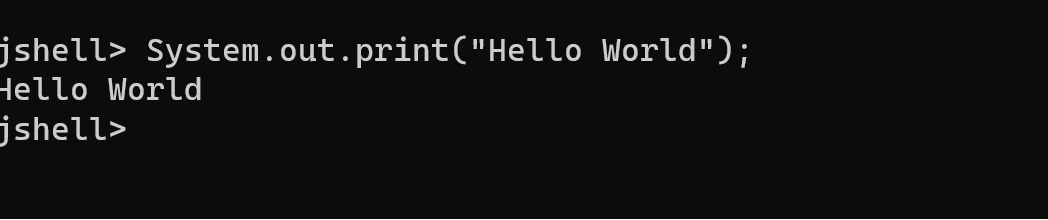
**Section 3: First Steps**



**=>Hello World**

System.out.print(“Hello World”);

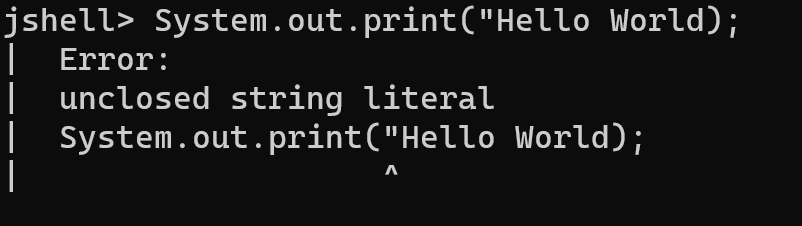
This is a command to print some information to the screen.



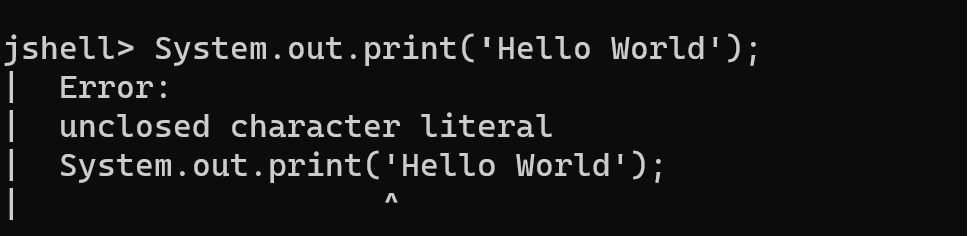
To modify existing line you can press the up arrow key-remember that in Jshell. You can see history of the lines you’ve previously typed using up arrows and down arrows.

**Errors:**

**1.Unclosed string literal**

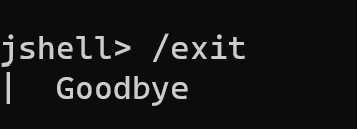
****

**2.Unclosed character literal**



Typing forward slash and the word ‘exit’, or forward slash with the shortcut text ex, will end your Jshell session,if you get stuck.

An example would be /exit or /ex



**=>Variables**

In programming, a variable is a named storage location that holds data, and its value can be changed during the execution of a program. Variables are fundamental to programming as they allow you to manipulate and store data in memory. Here are some key points about variables:

1. **Declaration:**
   * Before using a variable, you need to declare it. The declaration includes the data type of the variable and its name.
   * Example: **int age;**
2. **Initialization:**
   * After declaring a variable, you typically initialize it by assigning an initial value.
   * Example: **age = 25;**
3. **Assignment:**
   * You can change the value of a variable through assignment statements.
   * Example: **age = 30;**
4. **Data Types:**
   * Variables have data types that define the type of data they can hold (e.g., **int**, **double**, **String**).
   * Example: **double salary = 50000.50;**
5. **Naming Rules:**
   * Variable names must follow certain rules, such as starting with a letter, being case-sensitive, and not using reserved keywords.
   * Example: **String firstName;**
6. **Scope:**
   * The scope of a variable defines where in the code it can be accessed. For example, local variables are only accessible within the method or block where they are declared.
   * Example:

javaCopy code

void exampleMethod() { int localVar = 10; // localVar is only accessible within this method }

1. **Constants:**
   * You can use the **final** keyword to declare constants, which are variables whose values cannot be changed.
   * Example: **final double PI = 3.14159;**
2. **Instance and Class Variables:**
   * In object-oriented programming, variables can be instance variables (associated with an instance of a class) or class variables (associated with a class itself).
   * Example:

javaCopy code

public class MyClass { int instanceVar; // instance variable static int classVar; // class variable }

1. **Local Variables:**
   * Variables declared inside a method, constructor, or block are called local variables.
   * Example:

javaCopy code

void exampleMethod() { int localVar = 10; // local variable }

1. **Null and Default Values:**
   * Variables that are not explicitly initialized might have default values (e.g., **0** for numeric types, **false** for boolean, **null** for reference types).

These concepts are general to many programming languages, including Java. Understanding how to use variables is fundamental to writing effective and dynamic programs.

In Java, variables are used to store and manipulate data. They are essentially named storage locations that can hold values of a particular data type. Here are the basic types of variables in Java:

1. **Primitive Data Types:**
   * **byte:** 8-bit integer.
   * **short:** 16-bit integer.
   * **int:** 32-bit integer.
   * **long:** 64-bit integer.
   * **float:** 32-bit floating-point.
   * **double:** 64-bit floating-point.
   * **char:** 16-bit Unicode character.
   * **boolean:** Represents true or false.

Example:

javaCopy code

int age = 25; double salary = 50000.50; char grade = 'A'; boolean isStudent = true;

1. **Reference Data Types:**
   * **Objects:** Instances of classes.
   * **Arrays:** Homogeneous collections of elements.

Example:

javaCopy code

String name = "John"; Scanner scanner = new Scanner(System.in); int[] numbers = {1, 2, 3, 4, 5};

1. **Constants:**
   * Variables declared with the **final** keyword are constants, meaning their values cannot be changed once assigned.

Example:

javaCopy code

final double PI = 3.14159;

1. **Local Variables:**
   * Declared within a method, constructor, or block.
   * They must be initialized before use.

Example:

javaCopy code

void exampleMethod() { int localVar = 10; // Do something with localVar }

1. **Instance Variables (Non-Static Fields):**
   * Declared within a class but outside any method, constructor, or block.
   * Each instance of the class has its own copy of the instance variable.

Example:

javaCopy code

public class MyClass { int instanceVar = 5; }

1. **Class Variables (Static Fields):**
   * Declared with the **static** keyword.
   * Belongs to the class, not to instances of the class.

Example:

javaCopy code

public class MyClass { static int classVar = 10; }

It's important to note that Java is a statically-typed language, meaning that you must declare the type of a variable before using it. Additionally, Java follows the camelCase naming convention for variables, where the first word starts with a lowercase letter, and subsequent concatenated words begin with an uppercase letter.

In Java, keywords are reserved words that have special meanings and cannot be used as identifiers (such as variable names or class names). Here is a list of some of the keywords in Java:

1. **Primitive Data Type Keywords:**
   * **boolean**: Represents true or false values.
   * **byte**: 8-bit integer.
   * **short**: 16-bit integer.
   * **int**: 32-bit integer.
   * **long**: 64-bit integer.
   * **float**: 32-bit floating-point.
   * **double**: 64-bit floating-point.
   * **char**: 16-bit Unicode character.
2. **Flow Control Keywords:**
   * **if**: Conditional statement.
   * **else**: Alternative statement for if.
   * **switch**: Multi-way branch statement.
   * **case**: Defines a branch in a switch statement.
   * **default**: Defines the default branch in a switch statement.
   * **while**: Looping construct with a condition.
   * **do**: Looping construct that executes at least once.
   * **for**: Looping construct with initialization, condition, and iteration expressions.
   * **break**: Exits from a loop or a switch statement.
   * **continue**: Skips the rest of the loop and starts the next iteration.
3. **Modifiers:**
   * **public**: Access modifier indicating visibility to all classes.
   * **private**: Access modifier indicating visibility only within the declaring class.
   * **protected**: Access modifier indicating visibility to subclasses.
   * **static**: Modifier indicating that a variable or method belongs to the class, not instances of the class.
   * **final**: Modifier indicating that a variable, method, or class cannot be changed or extended.
   * **abstract**: Modifier indicating that a class or method is incomplete and must be implemented by a subclass.
   * **synchronized**: Modifier indicating that a method can be accessed by only one thread at a time.
4. **Class, Object, and Package Keywords:**
   * **class**: Declares a class.
   * **extends**: Indicates that a class is a subclass of another class.
   * **implements**: Indicates that a class implements an interface.
   * **interface**: Declares an interface.
   * **package**: Declares a package.
   * **import**: Imports a class or entire package.
5. **Exception Handling Keywords:**
   * **try**: Encloses a block of code that may throw exceptions.
   * **catch**: Catches and handles exceptions.
   * **finally**: Encloses a block of code that is executed regardless of whether an exception is thrown or not.
   * **throw**: Throws an exception manually.
   * **throws**: Specifies that a method might throw exceptions.
6. **Miscellaneous Keywords:**
   * **this**: Refers to the current instance of the class.
   * **super**: Refers to the superclass.
   * **new**: Creates a new object.
   * **return**: Returns a value from a method.
   * **void**: Indicates that a method does not return any value.
   * **instanceof**: Checks whether an object is an instance of a particular class or interface.

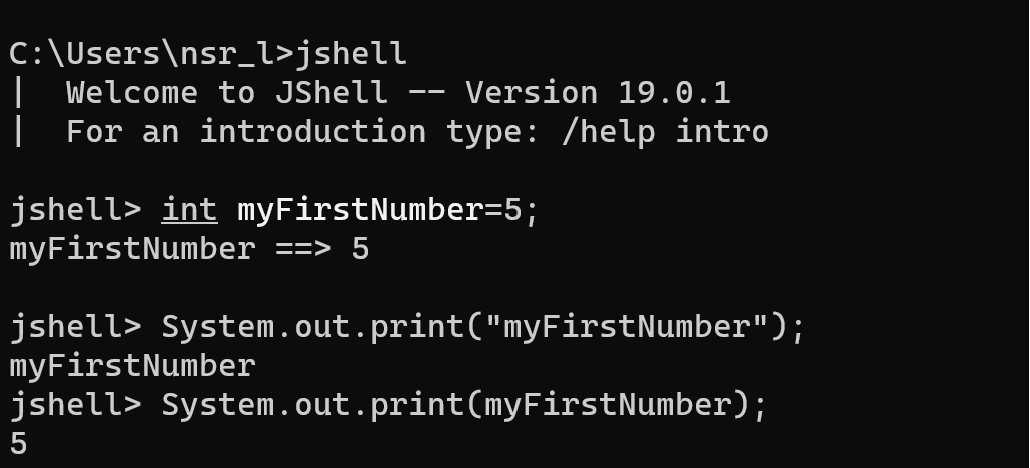
These are just some of the keywords in Java. It's essential to understand and use them correctly to write effective Java programs.

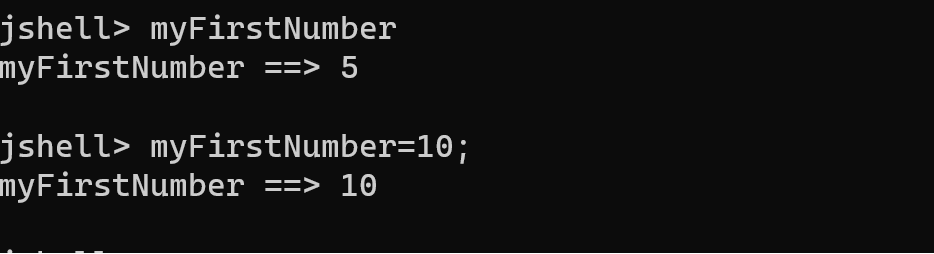
Here is a list of Java keywords:

1. **abstract**
2. **assert**
3. **boolean**
4. **break**
5. **byte**
6. **case**
7. **catch**
8. **char**
9. **class**
10. **const** (not used, reserved for future use)
11. **continue**
12. **default**
13. **do**
14. **double**
15. **else**
16. **enum**
17. **extends**
18. **final**
19. **finally**
20. **float**
21. **for**
22. **if**
23. **implements**
24. **import**
25. **instanceof**
26. **int**
27. **interface**
28. **long**
29. **native**
30. **new**
31. **package**
32. **private**
33. **protected**
34. **public**
35. **return**
36. **short**
37. **static**
38. **strictfp**
39. **super**
40. **switch**
41. **synchronized**
42. **this**
43. **throw**
44. **throws**
45. **transient**
46. **try**
47. **void**
48. **volatile**
49. **while**

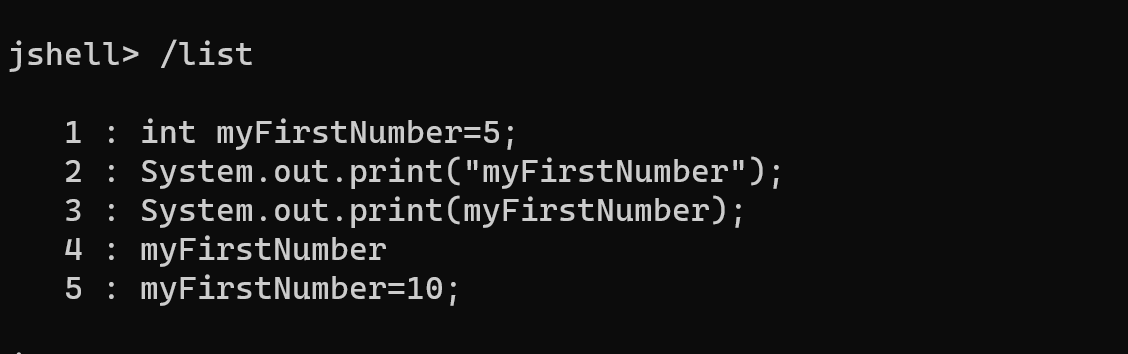
Note that **const** and **goto** are reserved keywords but not used in the language. They are reserved for potential future use. Additionally, starting from Java 9, **var** became a keyword for local variable type inference.

When we put something in double quotes it is **string literal**





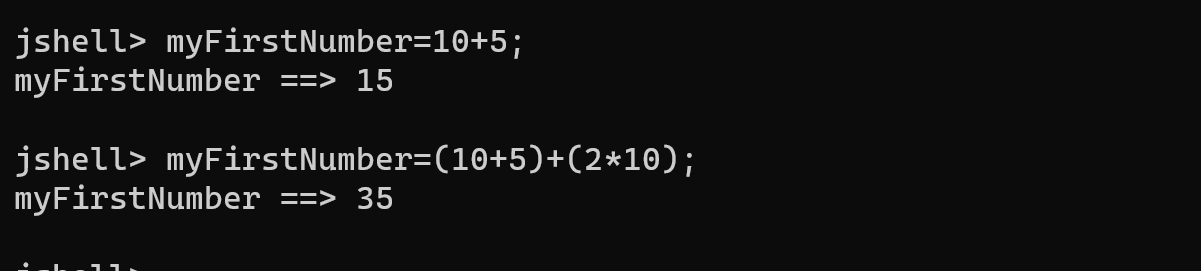
Command**: jshell>/list**



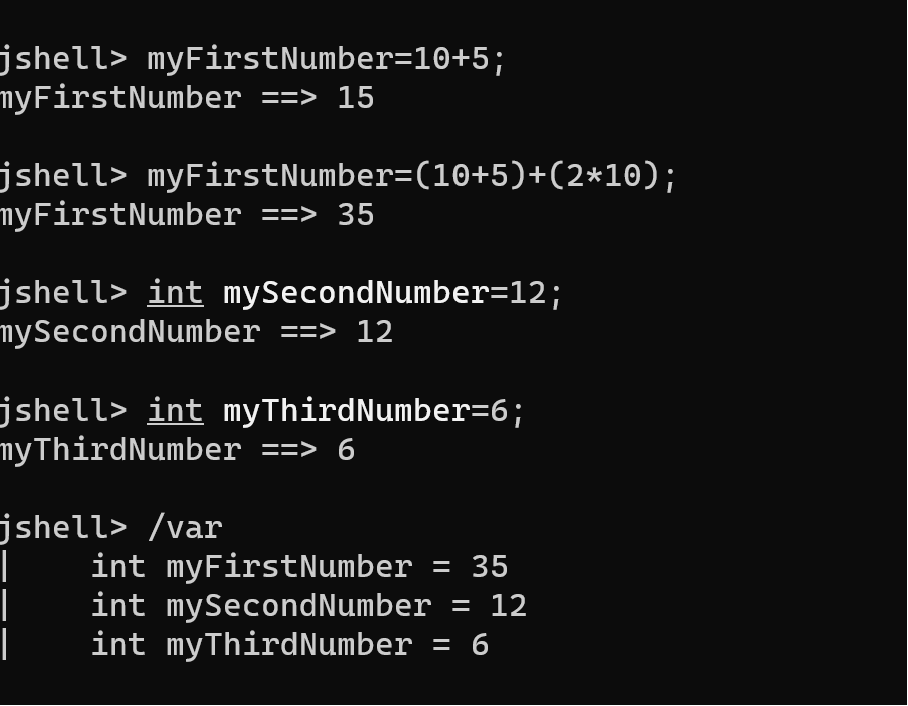
By declaring a variable again,we are efficiently re-declaring a variable,and in normal java programming,that would not be allowed ,and would throw an error

**Redeclaring a variable in java is not allowed**

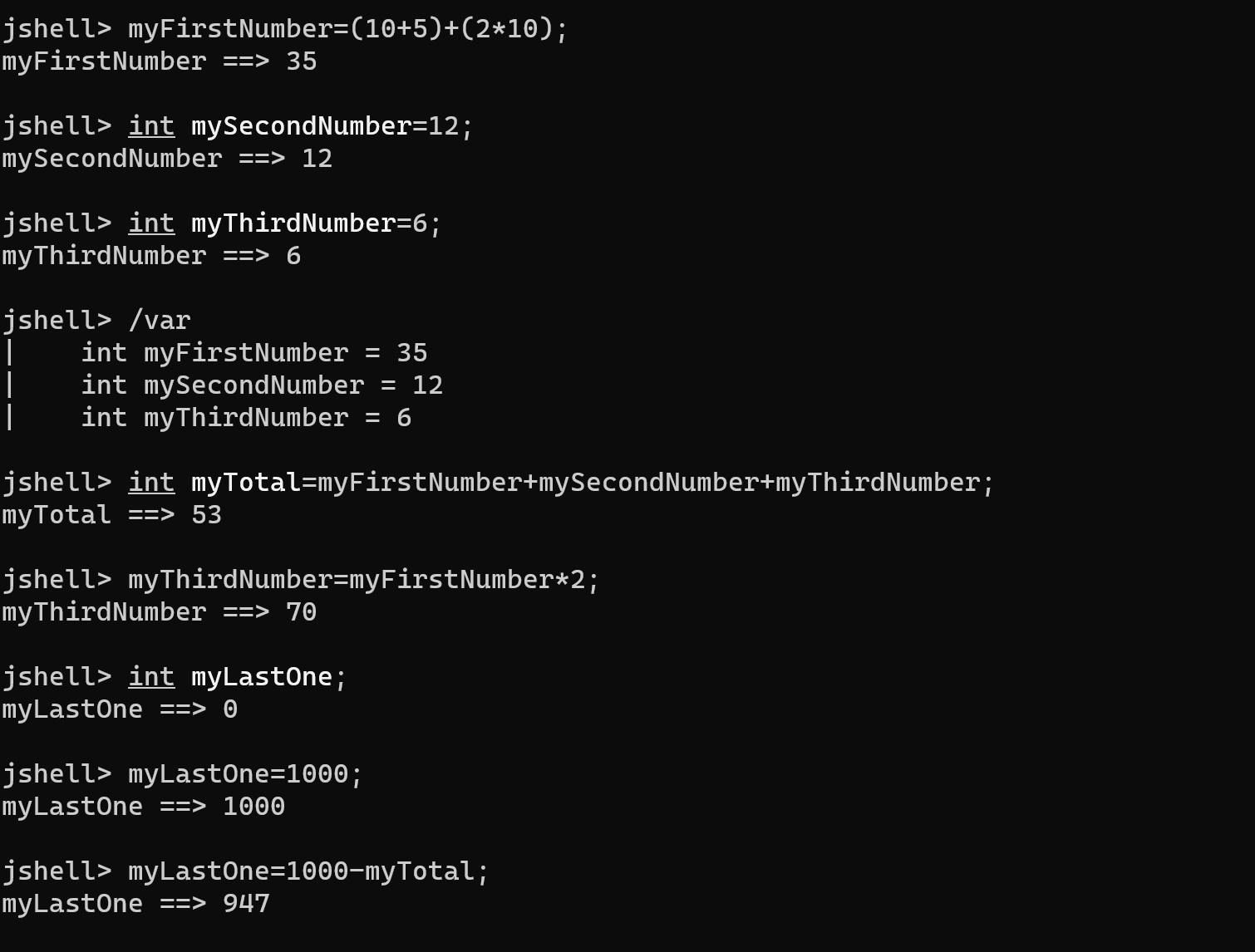
**Operators**



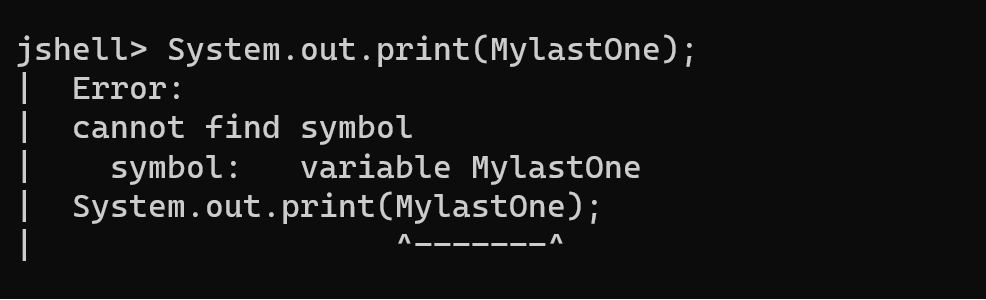
**=>Starting out with expressions**



Command:jshell>/var



Error:**3.cannot find symbol**



**Java code is case sensitive**

Int in lowercase,is not the same as Int with first letter capital or INT all in uppercase

Keywords need to be in lowercase

The **/vars** command in Jshell can you to identify any misspellings you have made.