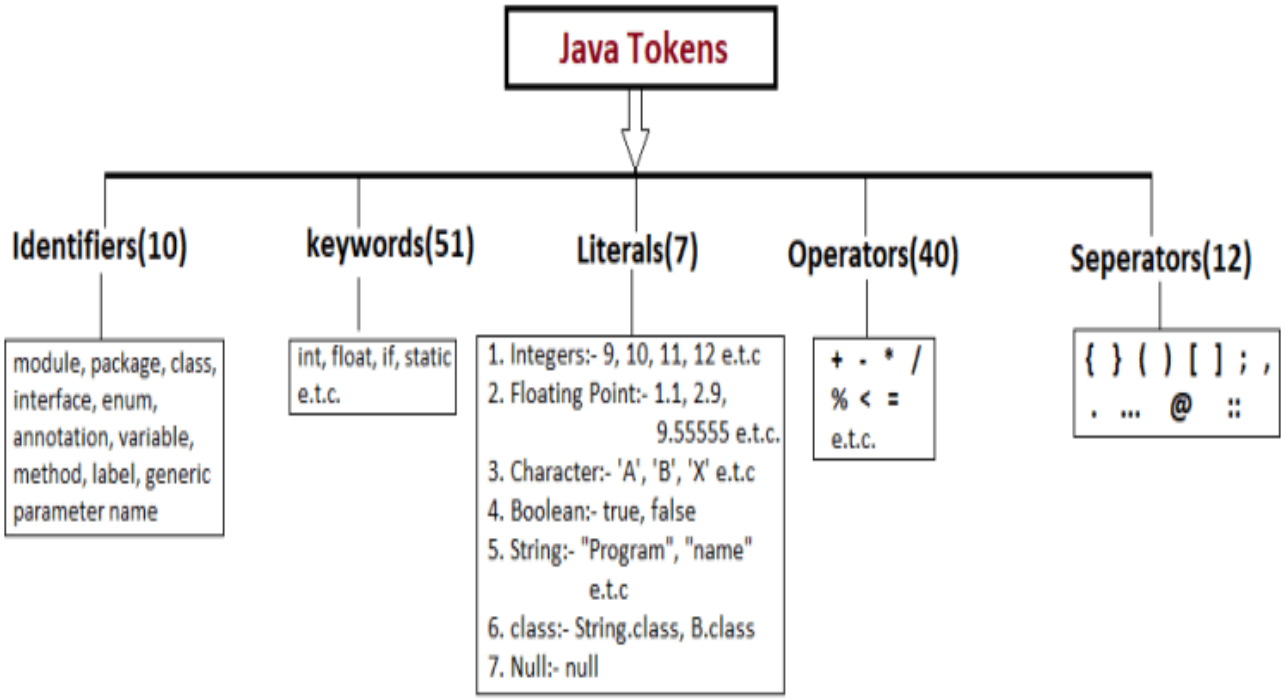
Java Tokens

A **token** is basically smallest element that can identify by the compiler. Java expressions and statements are constructed using java tokens.

Tokens can be classified as follows :->

1. Keywords
2. Identifiers
3. Constants
4. Special Symbols
5. Operators



1. **Keyword:** Keywords are pre-defined or reserved words in a programming language. Each keyword is meant to perform a specific function in a program. Since keywords are referred names for a compiler, they can't be used as variable names because by doing so,

we are trying to assign a new meaning to the keyword which is not allowed. **Java** language supports following keywords such as public,final,static,float, int etc.

2.**Identifiers:** Identifiers are used as the general terminology for naming of variables, functions and arrays. These are user-defined names consisting of an arbitrarily long sequence of letters and digits with either a letter or the underscore(_) as a first character. Identifier names must differ in spelling and case from any keywords. You cannot use keywords as identifiers; they are reserved for special use.
.
Examples of valid identifiers : abc, abc3, x1, _ab.

Examples of invalid identifiers :3abc, 1x ,@x, ~a.

3**.Constants/Literals:** Constants are also like normal variables. But, the only difference is, their values cannot be modified by the program once they are defined. Constants refer to fixed values. They are also called as literals.
Constants may belong to any of the data type.

Syntax:
final <data-type> <variable_name>=<value>;
ex:-
final int age=30;

4**.Special Symbols:** The following special symbols are used in Java having some special meaning and thus, cannot be used for some other purpose.

- **Brackets[]:** Opening and closing brackets are used as array element reference. These indicate single and multidimensional subscripts.
- **Parentheses():** These special symbols are used to indicate function calls and function parameters.
- **Braces{}:** These opening and ending curly braces marks the start and end of a block of code containing more than one executable statement.
- **comma (,):** It is used to separate more than one statements like for separating parameters in function calls.
- **semicolon :** It is an operator that essentially invokes something called an initialization list.
- **asterisk (*):** It is used to create pointer variable.
- **assignment operator:** It is used to assign values.

5.**Operators:** Java provides many types of operators which can be used according to the need. They are classified based on the functionality they provide. Some of the types are-

- Arithmetic Operators
- Unary Operators
- Assignment Operator
- Relational Operators
- Logical Operators
- Ternary Operator
- Bitwise Operators
- Shift Operators.