Java Tokens | A java program is basically made up of a group of classes and methods.
A class is a container that contains a set of declaration statements and methods containing executable statements.

A statement consists of variables, constants, operators, keywords, comments, identifiers, punctuators, etc. When the program is run, comments are stripped and executable statements are executed.

The output is called **translation unit**. A translation unit is a sequence of tokens: symbols, numbers, and words.

What are tokens in Java?

Tokens are the various elements in the java program that are identified by Java compiler. A token is the smallest individual element (unit) in a program that is meaningful to the compiler.
In simple words, a java program is a group of tokens, comments, and white spaces. For example, consider the below java statements:

```
final double p = 3.14 // A constant.



x = a + b; // An expression.



v = Math.pow(10, 1); // An inbuilt java function.
```

Let us consider the first statement, which is made up of six tokens: “final”, “double”, “p”, “=”, “3.14”, and “;”.

Similarly, second statement consists of six tokens: “x”, “=”, “a”, “+”, “b”, and “;”.

Types of Tokens

Java language contains five types of tokens that are as follows:

- Reserved Keywords
- Identifiers
- Literals,
- Operators
- Separators

Keywords

Keywords in Java are predefined or reserved words that have special meaning to the Java compiler. Each keyword is assigned a special task or function and cannot be changed by the user. You cannot use keywords as variables or identifiers as they are a part of Java syntax itself. A keyword should always be written in lowercase as Java is a case sensitive language. Java supports various keywords, some of them are listed in ppt

Identifier

Java Identifiers are the user-defined names of variables, methods, classes, arrays, packages, and interfaces. Once you assign an identifier in the Java program, you can use it to refer the value associated with that identifier in later statements. There are some de facto standards which you must follow while naming the identifiers such as:

- Identifiers must begin with a letter, dollar sign or underscore.
- Apart from the first character, an identifier can have any combination of characters.
- Identifiers in Java are case sensitive.
- Java Identifiers can be of any length.
- Identifier name cannot contain white spaces.
- Any identifier name must not begin with a digit but can contain digits within.
- Most importantly, **keywords** can’t be used as identifiers in Java.

Literals

Literals are syntactically representation of fixed value from the source.

. These literals are of immutable ,that means once declared its not changed

1. Integer
2. Floating Point
3. Character
4. String
5. Boolean

Operators

An operator in Java is a special symbol that signifies the compiler to perform some specific mathematical or non-mathematical operations on one or more operands.

Special Symbols

Special symbols in Java are a few characters which have special meaning known to Java compiler and cannot be used for any other purpose.

Symbol	Description
Brackets []	These are used as an array element reference and also indicates single and multidimensional subscripts
Parentheses()	These indicate a function call along with function parameters
Braces{}	The opening and ending curly braces indicate the beginning and end of a block of code having more than one statement
Comma (,)	This helps in separating more than one statement in an expression

Semi-Colon (;)	This is used to invoke an initialization list
Asterisk (*)	This is used to create a pointer variable in Java