Keywords

In Java, **keywords** are the reserved words that have some predefined meanings and are used by the Java compiler for some internal process or represent some predefined actions. These words cannot be used as identifiers such as **variable** names, method names, class names, or object names.

Now, let us go through a simple example before a deep dive into the article.

Example:

```java
// Java Program to demonstrate Keywords
class Demo {

    public static void main(String[] args)
    {
        // Using final and int keyword
        final int x = 10;

        // Using if and else keywords
        if(x > 10){
          System.out.println("Failed");
        }
        else {
          System.out.println("Successful demonstration"
                             +" of keywords.");
        }
    }
}
```

Output

```
Successful demonstration of keywords.
```

Java Keywords List

Java contains a list of keywords or reserved words which are also highlighted with different colors be it an IDE or editor in order to segregate the differences between flexible words and reserved words. As of **Java 21,** there are **53 keywords** defined in Java. They are listed below in the table with the primary action associated with them.

Keywords	Usage
abstract	Specifies that a class or method will be implemented later, in a subclass
assert	Assert describes a predicate placed in a Java program to indicate that the developer thinks that the predicate is always true at that place.
boolean	A data type that can hold True and False values only
break	A control statement for breaking out of loops.
byte	A data type that can hold 8-bit data values
case	Used in switch statements to mark blocks of text
catch	Catches exceptions generated by try statements
char	A data type that can hold unsigned 16-bit Unicode characters
class	Declares a new class
const	Reserved but not used
continue	Sends control back outside a loop
default	Specifies the default block of code in a switch statement
do	Starts a do-while loop
double	A data type that can hold 64-bit floating-point numbers

Keywords	Usage
else	Indicates alternative branches in an if statement
enum	A Java keyword is used to declare an enumerated type. Enumerations extend the base class.
extends	Indicates that a class is derived from another class or interface
final	Indicates that a variable holds a constant value or that a method will not be overridden
finally	Indicates a block of code in a try-catch structure that will always be executed
float	A data type that holds a 32-bit floating-point number
for	Used to start a for loop
goto	Reserved but not used
if	Tests a true/false expression and branches accordingly
implements	Specifies that a class implements an interface
import	References other classes
instanceof	Indicates whether an object is an instance of a specific class or implements an interface
int	A data type that can hold a 32-bit signed integer
interface	Declares an interface

Keywords	Usage
long	A data type that holds a 64-bit integer
native	Specifies that a method is implemented with native (platform-specific) code
new	Creates new objects
null	This indicates that a reference does not refer to anything
package	Declares a Java package
private	An access specifier indicating that a method or variable may be accessed only in the class it's declared in
protected	An access specifier indicating that a method or variable may only be accessed in the class it's declared in (or a subclass of the class it's declared in or other classes in the same package)
public	An access specifier used for classes, interfaces, methods, and variables indicating that an item is accessible throughout the application (or where the class that defines it is accessible)
return	Sends control and possibly a return value back from a called method
short	A data type that can hold a 16-bit integer
static	Indicates that a variable or method is a class method (rather than being limited to one particular object)

Keywords	Usage
strictfp	A Java keyword is used to restrict the precision and rounding of floating-point calculations to ensure portability.
super	Refers to a class's base class (used in a method or class constructor)
switch	A statement that executes code based on a test value
synchronized	Specifies critical sections or methods in multithreaded code
this	Refers to the current object in a method or constructor
throw	Creates an exception
throws	Indicates what exceptions may be thrown by a method
transient	Specifies that a variable is not part of an object's persistent state
try	Starts a block of code that will be tested for exceptions
void	Specifies that a method does not have a return value
volatile	This indicates that a variable may change asynchronously
while	Starts a while loop

Keywords	Usage
sealed	The sealed keyword is used to declare a class as "sealed," meaning it restricts which classes can extend it.
permits	The permits keyword is used within a sealed class declaration to specify the subclasses that are permitted to extend it.

Example: Using a keyword as a variable name would give error as shown below.

```java
// Java Program to illustrate what if
// we use the keywords as the variable name
class Demo
{
    public static void main(String[] args)
    {
        // Note "this" is a reserved
        // word in java
        String this = "Hello World!";
        System.out.println(this);
    }
}
```

Output:

```
./Geeks.java:9: error: not a statement
        String this = "Hello World!";
        ^
./Geeks.java:9: error: ';' expected
        String this = "Hello World!";
              ^
2 errors
```

Important Points:

- The keywords **const** and **goto** are reserved, even though they are not currently used in Java.
- **true, false,** and **null** look like keywords, but in actuality they are **literals**. However, they still can't be used as identifiers in a program.
- In Java, keywords are case-sensitive, and writing Java keywords in upper case (like IF instead of if) will throw an error.