

KEYWORD IN JAVA

Java keywords are also known as reserved words. Keywords are particular words that act as a key to a code. These are predefined words by Java so they cannot be used as a variable or object name or class name.

Example:

```
// Java Program to Illustrate What If We use the keywords as
// the variable name

// Driver Class
public class HelloWorld {
    // Main Function
    public static void main(String[] args)
    {
        // Note "this" is a reserved
        // word in java
        String this = "Hello World!";
        System.out.println(this);
    }
}
// Code Contributed by @Shubham Jain
```

Output:

```
HelloWorld.java:11: error: not a statement
    String this = "Hello World!";
    ^
HelloWorld.java:11: error: ';' expected
    String this = "Hello World!";
    ^
2 errors
```

Here's the complete list of all keywords in Java programming.

- **abstract:** used to declare a class or method as abstract. An abstract class is a class that cannot be instantiated, and an abstract method is a method without a body that must be implemented in a subclass.
- **assert:** used to perform assertion testing in Java. An assertion is a statement that should always be true, and if it is false, then an **AssertionError** is thrown.
- **boolean:** used to declare a boolean variable, which can only have two values: true or false.
- **break:** used to break out of a loop or switch statement.
- **byte:** used to declare a byte variable, which is a data type that can store values from -128 to 127.
- **case:** used in a switch statement to define a case label.
- **catch:** used to catch and handle exceptions in Java.
- **char:** used to declare a char variable, which is a data type that can store a single character.
- **class:** used to declare a class in Java.
- **const:** a keyword that was reserved but never implemented in Java.
- **continue:** used to skip the current iteration of a loop and continue to the next iteration.
- **default:** used in a switch statement to define a default case.
- **do:** used to start a do-while loop.
- **double:** used to declare a double variable, which is a data type that can store decimal values.
- **else:** used in an if statement to define an alternative block of code to execute if the condition is false.

- **enum:** used to declare an enumeration, which is a type that consists of a set of named constants.
- **extends:** used to extend a class in Java.
- **final:** used to declare a variable or method as final, which means that its value or implementation cannot be changed.
- **finally:** used in a try-catch block to define a block of code that will always be executed, regardless of whether an exception is thrown or not.
- **float:** used to declare a float variable, which is a data type that can store decimal values with less precision than double.
- **for:** used to start a for loop.
- **goto:** a keyword that was reserved but never implemented in Java.
- **if:** used to define a conditional statement in Java.
- **implements:** used to implement an interface in Java.
- **import:** used to import a package or class into a Java program.
- **instanceof:** used to check if an object is an instance of a particular class or interface.
- **int:** used to declare an int variable, which is a data type that can store whole numbers.
- **interface:** used to declare an interface in Java.
- **long:** used to declare a long variable, which is a data type that can store larger whole numbers than int.
- **native:** used to declare a method as native, which means that its implementation is provided by the underlying platform, rather than in Java code.
- **new:** used to create a new object in Java.
- **package:** used to define a package in Java.

- **private:** used to declare a variable or method as private, which means that it can only be accessed within the same class.
- **protected:** used to declare a variable or method as protected, which means that it can be accessed within the same class or any subclass.
- **public:** used to declare a variable or method as public, which means that it can be accessed from anywhere in the Java program.
- **return:** used to return a value from a method or exit a method without returning a value.
- **short:** used to declare a short variable, which is a data type that can store smaller whole numbers than int.
- **static:** used to declare a variable or method as static, which means that it belongs to the class rather than to individual objects of the class.
- **strictfp:** used to enforce strict floating-point precision in Java.
- **super:** used to call a method or constructor in the superclass.
- **switch:** used to start a switch statement in Java.
- **synchronized:** used to ensure that only one thread can access a block of code or object at a time in Java.
- **this:** used to refer to the current object in Java.
- **throw:** used to throw an exception in Java.
- **throws:** used to declare that a method may throw an exception in Java.
- **transient:** used to declare a variable as transient, which means that it will not be serialized when the object is written to a file or transmitted over a network.
- **try:** used to start a try-catch block in Java.
- **void:** used to declare a method that does not return a value.

- **volatile:** used to declare a variable as volatile, which means that it is subject to optimization by the Java Virtual Machine.

Keywords in Java:

abstarct	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while