

Palabras reservadas

Son palabras particulares que actúan como clave de un código. Estas son palabras predefinidas, por lo que no se pueden usar como una variable o un nombre de objeto.

Lista de palabras clave de Java

1. **abstract** : se utiliza para declarar la clase **abstract**. La clase abstracta puede implementar una interfaz. Puede tener métodos abstractos y no abstractos.
2. **assert** (agregado en J2SE 1.4) Assert describe un predicado (una declaración verdadero-falso) colocado en un programa Java para indicar que el desarrollador piensa que el predicado siempre es verdadero en ese lugar. Si una aserción se evalúa como falsa en tiempo de ejecución, se produce un error de aserción, que generalmente hace que la ejecución se cancele. Opcionalmente habilitado por el método ClassLoader.
3. **boolean**: se utiliza para declarar una variable como un tipo booleano. Solo puede contener valores verdaderos y falsos.
4. **break** : se usa para romper el bucle o cambiar la declaración. Rompe el flujo actual del programa en la condición especificada.
5. **byte** : se utiliza para declarar una variable que puede contener valores de datos de 8 bits.
6. **case** : la palabra clave de **case** Java se usa con las instrucciones de cambio para marcar bloques de texto.
7. **catch** : se utiliza para capturar las excepciones generadas por las declaraciones try. Debe usarse solo después del bloque try.
8. **char** : Java se usa para declarar una variable que puede contener caracteres Unicode de 16 bits sin signo
9. **class** : se usa para declarar una clase.
10. **continue** : se usa para continuar el ciclo. Continúa el flujo actual del programa y omite el código restante en la condición especificada.
11. **default** : *Java se utiliza para especificar el bloque de código predeterminado en una instrucción de cambio.*
12. **do** : se usa en la declaración de control para declarar un bucle. Puede iterar una parte del programa varias veces.
13. **double** : se usa para declarar una variable que puede contener números de coma flotante de 64 bits.
14. **else** : se usa para indicar las ramas alternativas en una declaración if.
15. **enum** : se utiliza para definir un conjunto fijo de constantes. Los constructores Enum son siempre privados o predeterminados.
16. **extends** : se utiliza para indicar que una clase se deriva de otra clase o interfaz.
17. **final** : se usa para indicar que una variable tiene un valor constante. Se aplica con una variable. Se utiliza para restringir al usuario.
18. **False**: Un valor literal booleano.

19. **Finally** : indica un bloque de código en una estructura try-catch. Este bloque siempre se ejecuta si la excepción se maneja o no.
20. **float** : se usa para declarar una variable que puede contener un número de coma flotante de 32 bits.
21. **for** : Se utiliza para ejecutar un conjunto de instrucciones / funciones repetidamente cuando se cumplen ciertas condiciones. Si el número de iteraciones es fijo, se recomienda usar para el bucle.
22. **if** : Ejecuta el bloque if si la condición es verdadera.
23. **implements** : se utiliza para implementar una interfaz.
24. **import** : Java hace que las clases e interfaces estén disponibles y accesibles para el código fuente actual.
25. **instanceof** : se usa para probar si el objeto es una instancia de la clase especificada o si implementa una interfaz.
26. **int** : Java se utiliza para declarar una variable que puede contener un entero con signo de 32 bits.
27. **interface** : se utiliza para declarar una interfaz. Solo puede tener métodos abstractos.
28. **long** : se usa para declarar una variable que puede contener un número entero de 64 bits.
29. **native** : para especificar que un método se implementa en código nativo utilizando JNI (Java Native Interface).
30. **new** : se usa para crear nuevos objetos.
31. **null** : se utiliza para indicar que una referencia no hace referencia a nada. Elimina el valor de la basura.
32. **package** : se utiliza para declarar un paquete Java que incluye las clases.
33. **private** : Se utiliza para indicar que solo se puede acceder a un método o variable en la clase en la que se declara.
34. **protected** : es un modificador de acceso. Se puede acceder dentro del paquete y fuera del paquete, pero solo a través de la herencia. No se puede aplicar en la clase.
35. **public** : la palabra clave pública de Java es un modificador de acceso. Se utiliza para indicar que se puede acceder a un elemento desde cualquier lugar. Tiene el alcance más amplio entre todos los demás modificadores.
36. **requires** : Se usa para especificar las bibliotecas requeridas dentro de un módulo.
37. **return** : se usa para regresar de un método cuando se completa su ejecución.
38. **short** se usa para declarar una variable que puede contener un número entero de 16 bits.
39. **static** : Java se usa para indicar que una variable o método es un método de clase. La palabra clave estática en Java se utiliza principalmente para la gestión de memoria.
40. **strictlyfp** : strictlyfp se utiliza para restringir los cálculos de punto flotante para garantizar la portabilidad.
41. **super** : es una variable de referencia que se utiliza para referirse a un objeto de clase padre. Se puede utilizar para invocar el método de clase primaria inmediata.
42. **switch** : contiene una instrucción que ejecuta código basado en el valor de prueba. La instrucción switch prueba la igualdad de una variable con múltiples valores.
43. **synchronized** : se utiliza para especificar las secciones o métodos críticos en el código multiproceso.
44. **this** : se puede usar para referir el objeto actual en un método o constructor.

45. **throw** : se usa para lanzar explícitamente una excepción. La palabra clave throw se usa principalmente para lanzar una excepción personalizada. Es seguido por una instancia.
46. **throws** : se utiliza para declarar una excepción. La excepción marcada se puede propagar con lanzamientos.
47. **transient** : se utiliza en la serialización. Si define algún miembro de datos como transitorio, no se serializará.
48. **True**: un valor booleano.
49. **try** : se usa para iniciar un bloque de código que se probará en busca de excepciones. El bloque try debe ser seguido por catch o finalmente block.
50. **Var**: indicador especial que ya no se puede utilizar como tipo.
51. **void**: void se usa para especificar que un método no tiene un valor de retorno.
52. **volátil** : la palabra clave Java volátil se usa para indicar que una variable puede cambiar de forma asincrónica.
53. **while** : se usa para iniciar un ciclo while. Este ciclo itera una parte del programa varias veces. Si el número de iteraciones no es fijo, se recomienda usar el ciclo while.