



Tutoriales de informática
C CMD DTD HTML Java XML XSD {+}

Tutorial de Java

- Guion del tutorial
- Qué es Java
- Instalación del JDK
- Configuración de las variables de entorno
- HolaMundo.java desde la consola
- Comentarios
- Variables
- Constantes
- Instrucciones de control en Java
- Subprogramas
- Arrays
- Clases y objetos
- Herencia

Referencia de Java

- Palabras clave del lenguaje Java
- Tipos de datos primitivos

Material extra

- Ejercicios resueltos

Más recursos

- Libros de Java

Otros tutoriales

- Lenguaje C
- Pseudocódigo
- {+}

Inicio » Tutorial de Java » Palabras clave

Palabras clave (reservadas) del lenguaje Java

En el lenguaje de programación Java se puede hacer uso de las palabras clave (*keywords*) –también llamadas palabras reservadas– mostradas en la siguiente tabla, donde se indica la funcionalidad de cada una de ellas. Dichas palabras, no pueden ser utilizadas como identificadores por los programadores para definir variables, constantes, etc.

Nota: `true`, `false` y `null` no son considerados palabras clave de Java, sino literales. Ahora bien, tampoco se pueden utilizar como identificadores.

Lista de palabras clave (<i>keywords</i>) del lenguaje Java	
Palabra	Descripción de su funcionalidad
<code>abstract</code>	Se utiliza para definir clases y métodos abstractos.
<code>assert</code>	Sirve para afirmar que una condición es cierta.
<code>boolean</code>	Tipo de dato primitivo booleano (lógico), que puede ser <code>true</code> o <code>false</code> .
<code>break</code>	Instrucción de salto que interrumpe (rompe) la ejecución de un bucle o de una instrucción de control alternativa múltiple (<code>switch</code>).
<code>byte</code>	Tipo de dato primitivo número entero (<i>integer</i>) de 8 bits.
<code>case</code>	Caso de una instrucción de control alternativa múltiple (<code>switch</code>).
<code>catch</code>	Cláusula de un bloque <code>try</code> donde se especifica una excepción.
<code>char</code>	Tipo de dato primitivo carácter (valor Unicode) de 16 bits.
<code>class</code>	Sirve para definir una clase.
<code>const</code> *	No se utiliza.
<code>continue</code>	Instrucción de salto que interrumpe (rompe) la ejecución de la iteración de un bucle. Pero, permitiendo continuar al bucle seguir realizando otras iteraciones.
<code>default</code>	Caso por defecto de una instrucción de control alternativa múltiple (<code>switch</code>).

Palabra	Descripción de su funcionalidad
do	Se usa en la sintaxis de un bucle hacer mientras (do while).
double	Tipo de dato primitivo número real en coma flotante con precisión doble (<i>double-precision floating-point</i>) de 64 bits.
else	<i>Si no</i> , en una instrucción de control alternativa doble (if else).
enum	Sirve para definir tipos de datos enumerados.
extends	Cláusula que permite indicar la clase padre de una clase.
final	Permite indicar que una variable no se puede modificar, un método no se puede redefinir o de una clase no se puede heredar.
finally	Clausula que permite especificar un bloque de código que siempre se ejecutará, se produzca o no una excepción en un bloque try .
float	Tipo de dato primitivo número real en coma flotante con precisión simple (<i>single-precision floating-point</i>) de 32 bits.
for	Instrucción de control repetitiva para.
goto *	Instrucción de salto (<i>ir a</i>). No se usa.
if	Se emplea para escribir instrucciones de control alternativas simples (if) o dobles (if else).
implements	Sirve para definir la o las interfaces de una clase.
import	Permite importar un paquete (<i>package</i>).
instanceof	Operador que permite saber si un objeto es una instancia de una clase concreta.
int	Tipo de dato primitivo número entero (<i>integer</i>) de 32 bits.
interface	Se utiliza para declarar una interfaz.
long	Tipo de dato primitivo número entero (<i>integer</i>) de 64 bits.
native	Modificador que se utiliza para indicar que un método está implementado en un lenguaje de programación (distinto a Java) dependiente de la plataforma.
new	Operador que se utiliza para crear un objeto nuevo de una clase.
package	Agrupar a un conjunto de clases.
private	Modificador de acceso para indicar que un elemento es accesible únicamente desde la clase donde se ha definido.
protected	Modificador de acceso para indicar que un elemento es accesible desde la clase donde se ha definido, subclases de ella y otras clases del mismo paquete (<i>package</i>).
public	Modificador de acceso para indicar que un elemento es accesible desde cualquier clase.
return	Se usa para indicar el valor de retorno de un método.

<i>Palabra</i>	<i>Descripción de su funcionalidad</i>
short	Tipo de dato primitivo número entero (<i>integer</i>) de 16 bits.
static	Permite especificar que un elemento es único en una clase, no pudiendo existir instancias de esa clase que contengan a dicho elemento.
strictfp	Se usa para indicar que se tienen que utilizar cálculos en coma flotante estricto (<i>strict floating point</i>).
super	Permite invocar a un método o constructor de la superclase.
switch	Instrucción de control alternativa múltiple.
synchronized	Modificador que se utiliza para indicar que un método o bloque de código es atómico.
this	Se utiliza para referenciar al objeto actual, así como para invocar a un constructor de la clase a la que pertenece dicho objeto.
throw	Permite lanzar una excepción
throws	Sirve para indicar las excepciones que un método puede lanzar.
transient	Sirve para especificar que un atributo no sea persistente.
try	Permite especificar un bloque de código donde se quieren atrapar excepciones.
void	Tipo de dato vacío (sin valor).
volatile	Modificador que se usa para indicar que el valor de un atributo que está siendo utilizado por varios hilos (<i>threads</i>) esté sincronizado.
while	Se usa para escribir bucles mientras (while) y bucles hacer mientras (do while).

(*) Con un asterisco se han marcado las palabras clave que no se utilizan.