



APRENDERAPROGRAMAR.COM

SENTENCIA DE ASIGNACIÓN  
EN JAVA. OPERADOR DE  
ASIGNACIÓN COMPUESTA  
+= Y -=. EJEMPLO.  
(CU00635B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

**Resumen:** Entrega nº35 curso Aprender programación Java desde cero.

Autor: Alex Rodríguez

## SENTENCIA DE ASIGNACIÓN EN JAVA. OPERADOR DE ASIGNACIÓN COMPUESTA.

Vamos a ver cómo realizar asignación de contenido a variables. De momento hablaremos solo de variables, ya que el tratamiento de objetos conviene hacerlo con cautela. Además veremos el operador de asignación compuesta, un operador que es de uso opcional.



La asignación de contenido a variables se hace en Java de la siguiente manera: `variable = expresión;`

Por ejemplo `precio = precioBase + 0.35;` En toda asignación debe haber coincidencia de tipos. Si peso es una variable de tipo `int` y sobrepeso es una variable de tipo `double` ó `boolean` debemos evitar sentencias como `sobrepeso = 2 * peso;` Aunque hay ocasiones en que la falta de coincidencia de tipos será asumida por Java, debemos tratar de evitarlo en todo caso porque introduce inseguridad en la programación.

En Java una sentencia como `saldo = saldo + cantidad;` se puede escribir también así: `saldo += cantidad;`

***+= se denomina operador de asignación compuesta y sirve para sumarle una cantidad al valor de una variable. También se admite el uso del operador -= que en vez de sumar lo que hace es restar. El uso de los operadores de asignación compuesta es opcional: hay programadores que los usan y otros que no.***

Un ejemplo de asignación compuesta con -= sería: `saldo -= precioCaja * cantidadCajas;`

**Próxima entrega:** CU00636B

**Acceso al curso completo** en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=68&Itemid=188](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188)