

# Literales Conversión de tipos

## Literales

### Concepto y consideraciones

Llamamos literal a un valor concreto, como 345, 24.6 o true. Veamos algunas consideraciones sobre el uso de literales en Java que debes de tener en cuenta:

- Todo literal entero es de tipo int.
- Todo literal decimal es de tipo double. Hay que tener precaución a la hora de asignar un literal decimal a una variable. Por ejemplo, la siguiente instrucción provocaría un error de compilación:

```
float g=3.5;
```

Al ser el literal un double, no se puede asignar a una variable float. Si queremos que el valor sea considerado como float, le pondremos la letra f al final:

```
float g=3.5f; //ok
```

- Los literales boolean son *true* y *false*.
- Un literal char se representa entre comilla simple: '@', o bien mediante el valor unicode del caracter en hexadecimal: '\u3af1'

### Representación

Los literales enteros se pueden representar en:

- Decimal: 289
- Octal: 0453
- Hexadecimal: 0xAF7
- Binario: 0b100011

Para representar un literal numérico, se puede utilizar el símbolo \_

```
int n=2_345;
```

```
double d= 45.6_9;
```

Pero no se puede utilizar ese símbolo\_ al principio, al final o junto al punto decimal. Las siguientes instrucciones son errores de compilación:

```
int n=_345;
```

```
double d= 45._9;
```

```
long ln=234_;
```

## Conversiones de tipos

Habitualmente, es necesario almacenar en una variable un dato de un tipo diferente al de la propia variable. Al hacerlo se produce una conversión del dato al tipo de la variable destino. Es posible hacer conversiones entre todos los tipos primitivos Java a excepción del boolean. El tipo boolean es incompatible con el resto, lo que significa que un boolean no puede ser convertido a ningún otro tipo Java, ni viceversa.

Dejando, por tanto, al tipo boolean de lado, una conversión entre tipos primitivos Java puede ser de dos tipos:

- **Implícita.** El dato se convierte directamente al tipo de la variable durante la asignación. En el siguiente ejemplo, el dato int existente en la variable x es convertido implícitamente a double al asignarlo a n:

```
int x=45;
```

```
double n=x;
```

- **Explícita.** Se debe indicar en la instrucción de asignación el tipo al que hay que convertir el dato:

```
double r=4.5f;
```

```
int x=(int)r; //conversión explícita o casting
```

Pero, ¿cuando una conversión debe realizarse de forma explícita?. Las conversiones implícitas se realizan en los siguientes casos:

- El tipo de la variable destino es de igual o mayor tamaño que el del dato.
- La regla anterior tiene las siguientes excepciones que no permiten conversiones implícitas:
  - El tipo de origen es numérico (cualquier tipo) y el de destino es char:

```
byte b=5;
```

```
char n=b;//error
```

- El tipo de destino es entero y el de origen es decimal:

```
float r=3.4f;
```

```
long l=r; //error
```

En todos aquellos casos en que no sea posible realizar conversiones implícitas, podrán realizarse de forma explícita. Por ejemplo:

```
byte b=5;
```

```
char n=(char)b;//ok
```

```
float r=3.4f;
```

```
long l=(long)r; //ok
```

```
double k=489.5;
```

```
byte s=(byte)k; //ok, el dato será truncado
```