

Tipos de datos en Java

Clasificación

- Tipos primitivos. Números, datos lógicos y carácter.

200

true

'c'

- Tipos objetos. Cualquier objeto Java.

new Object()

“es un texto”

Tipos primitivos

Tipo	Valores	Ejemplo
boolean	true o false	true
byte	Entero de 8 bits	39
short	Entero de 16 bits	780
int	Entero de 32 bits	59400
long	Entero de 64 bits	200000
float	Decimal 32 bits	45.6f
double	Decimal 64 bits	80.4
char	código unicode de 16 bits	'@'

Literales

➤ Los literales enteros se pueden representar en:

- Decimal: 289

- Octal: 0453

- Hexadecimal: 0xAF7

- Binario: 0b100011

➤ Se puede utilizar el símbolo `_` para representar un literal numérico:

```
int n=2_345;  
double d= 45.6_9;
```

Literales II

- No se puede utilizar el símbolo `_` al principio, al final o junto al punto decimal. Los siguientes son errores de compilación:

```
int n=_345;  
double d= 45._9;  
long ln=234_;
```

- Los literales enteros son `int`, si queremos que sea `long`:

345l

- Los literales decimales son `double`, si queremos que sea `float`:

4.7f



float g=6.7;//Error de compilación:

Conversiones de tipos

- Todos los tipos primitivos son convertibles en otros, salvo boolean.
- Las conversiones pueden ser implícitas

```
int x=45;  
double n=x;
```

- O explícitas:

```
double r=4.5f;  
int x=(int)r; //casting
```

Reglas conversiones implícitas

➤ El tipo de destino tiene que ser de igual o mayor tamaño que el de origen.

➤ 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
```

➤ Cuando no sea posible conversión implícita, siempre se podrá realizar explícitamente:

```
char n=(char)b;  
long l=(long)r;
```

Ejemplos conversiones

Conversiones correctas:

```
char c='@';  
int p=c; //ok  
int num=3450;  
byte r=(byte)num; //ok  
c=(char)34.5; //ok
```

Conversiones incorrectas:

```
char c='@';  
byte p=c; //error  
boolean b=false;  
int n=(int)b; //error
```


Tipos objeto

- Son todos los objetos de cualquier clase Java.
- Se manejan a través de variables de su tipo (clase)
- La variable contiene una referencia al objeto
- Mediante la variable se accede a los métodos del objeto:

```
String s=new String ("hola");  
s.length();
```

- NO se puede hacer conversión ni implícita ni implícita entre tipos primitivos y objetos

```
String s=new String ("27");  
int n=(int)s; //error de compilación
```