



Temas Puntuales:

Casting y su uso con las variables :

Conversiones: cast

Igualmente agregamos un tabla comparativa para tener presente al momento de codificar nuestro trabajo.

Ejemplo	Comentario
<code>int n= 34;</code>	Entero en base 10
<code>long j = 4056L;</code>	Con la terminación de l o L se toma como entero largo
<code>long H = 0xFFFFFFL;</code>	Número Hexadecimal
<code>doublé f1 = 1.345;</code>	Doble es por defecto
<code>float f2 =1.356F</code>	Con la terminación de f o F se toma como de tipo float
<code>int n2 = 48;</code>	Necesita un casting <code>int n2=(int)48;</code>
<code>short s2 = 34000;</code>	Genera un error de desbordamiento
<code>float f2 = 1.356;</code>	Necesita un casting <code>float f2 =(float)1.356;</code>
<code>byte mini = 56;</code>	Necesita un casting <code>byte mini=(byte) 56;</code>

- Java realiza conversiones implícitas de menor a mayor rango.
- Eventualmente hay pérdida de significancia al convertir enteros a punto flotante.
- Todas las conversiones de mayor a menor rango deben ser explícitas por medio del operador de cast:

(tipo)expresión

Ejemplos:

```
int i= 1;
short s= 2;
int j= i+s;

s= a; // error short < int
s= (short)a; // Ok
```



```
i= 1.0; // error int < double
i= (int)1.0; // Ok

f= 1.0; // error float < double
f= (float)1.0; // Ok
f= 1.0f; // Ok
```

boolean v=(boolean)a; // error no existe
conversión

Vamos a ver otro ejemplo

```
public class PruebaApp {
1
2     public static void main(String[] args) {
3
4         int a=1;
5
6         double b=2.5;
7
8         b=a;
9
10        System.out.println(b);
11
12    }
13 }
```

El código anterior, no nos dará ningún error, ya que **double** si puede contener un **int**, pero si cambiamos la linea **b=a;** por **a=b;** nos informara un error, esto es porque **int** no puede contener un **double**, pero si usamos un casting **sí** nos dejara ejecutar.

```
public class PruebaApp {
1
2     public static void main(String[] args) {
3
4         int a=1;
5
6         double b=2.5;
7
8         a=(int)b;
9
10        System.out.println(a);
11
12    }
```

13 }



Como vemos, un casting se indica entre paréntesis con el tipo de dato al que queremos pasar, en este caso `int`. La variable `a`, contendrá un 2 y no un 2.5.

Mensajes de error que puede dar Java por causa del Cast.

El error relacionado con los castings es **Type mismatch: cannot convert from (dato1) to (dato2)**

No se puede hacer castings entre todos los tipos, por ejemplo, entre un `int` y un `String`. El error que aparece es este **Cannot cast from (dato1) to (dato2)**.

Vamos a ver otro ejemplo muy común para usar un casting., un `int` y `char` son compatibles, si tenemos un `int` y hacemos un casting a `char` este se convertirá en un carácter asociado de la tabla ASCII. Veamos un ejemplo:

```
1    public class PruebaApp {
2
3        public static void main(String[] args) {
4
5            int codigo=97;
6
7            char codigoASCII=(char)codigo;
8
9            System.out.println(codigoASCII);
10        }
11    }
```

Precedencia de Operadores

```
.  ( )  [ ]
unarios:  -  +  (cast)
          *  /  %
          +  -
          <  >  <=  >=  ==  !=
          !
          ||
          &&
```



Nota:

Recordemos que cuando le asignamos un valor a una variable usamos una expresión del tipo siguiente, usamos un **solo igual**

```
short s= 2;
```

pero cuando usamos la variable en una comparación lógica tipo

```
if ( x == 10) {
```

Usamos doble ==

También con la declaración de la variables recordar que sino las declaro al principio , y las coloco en una construcción de este estilo... condicional como en este ejemplo.....va tirar error pues las debo definir fuera de las estructuras no dentro de una condición... por ejemplo

```
public class b5 {

    public static void main(String args[]) throws IOException {
        BufferedReader bufEntrada = new BufferedReader(new
InputStreamReader(System.in));
        NO FUE DECLARADA LA VARIABLE a
        double b;

        double c;
        double pro;
        double sum;
        System.out.println("Nro A: ");
        a = Double.parseDouble(bufEntrada.readLine());
        System.out.println("Nro B: ");
        b = Double.parseDouble(bufEntrada.readLine());
        System.out.println("Nro C: ");
        c = Double.parseDouble(bufEntrada.readLine());
        if (a=0) {
            pro = a*b*c;
            System.out.println("El Producto de los 3 es:
"+pro);
        } else {
            sum = a+b+c;
            System.out.println("La suma de los 3 es: "+sum);
        }
    }

}
```