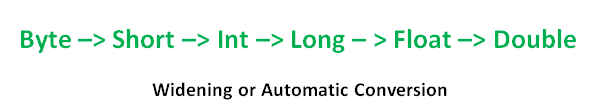
When you assign value of one data type to another, the two types might not be compatible with each other. If the data types are compatible, then Java will perform the conversion automatically known as Automatic Type Conversion and if not then they need to be casted or converted explicitly. For example, assigning an int value to a long variable.

**Widening or Automatic Type Conversion**

Widening conversion takes place when two data types are automatically converted. This happens when:

* The two data types are compatible.
* When we assign value of a smaller data type to a bigger data type.

For Example, in java the numeric data types are compatible with each other but no automatic conversion is supported from numeric type to char or boolean. Also, char and boolean are not compatible with each other.



int i = 100;

**// automatic type conversion**

long l = i;

**// automatic type conversion**

float f = l;

System.out.println("Int value "+i);

System.out.println("Long value "+l);

System.out.println("Float value "+f);

Output:

Int value 100

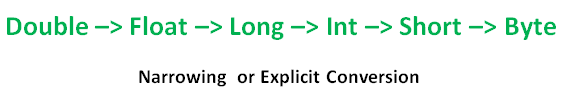
Long value 100

Float value 100.0

**Narrowing or Explicit Conversion**

If we want to assign a value of larger data type to a smaller data type we perform explicit type casting or narrowing.

* This is useful for incompatible data types where automatic conversion cannot be done.
* Here, target-type specifies the desired type to convert the specified value to.



char and number are not compatible with each other. Let’s see when we try to convert one into other.

char ch = 'c';

int num = 88;

ch = num;

error: incompatible types: possible lossy conversion from int to char

ch = num;

^

1 error

How to do Explicit Conversion?

double d = 100.04;

        //explicit type casting

        long l = (long)d;

        //explicit type casting

        int i = (int)l;

        System.out.println("Double value "+d);

        //fractional part lost

        System.out.println("Long value "+l);

        //fractional part lost

        System.out.println("Int value "+i);