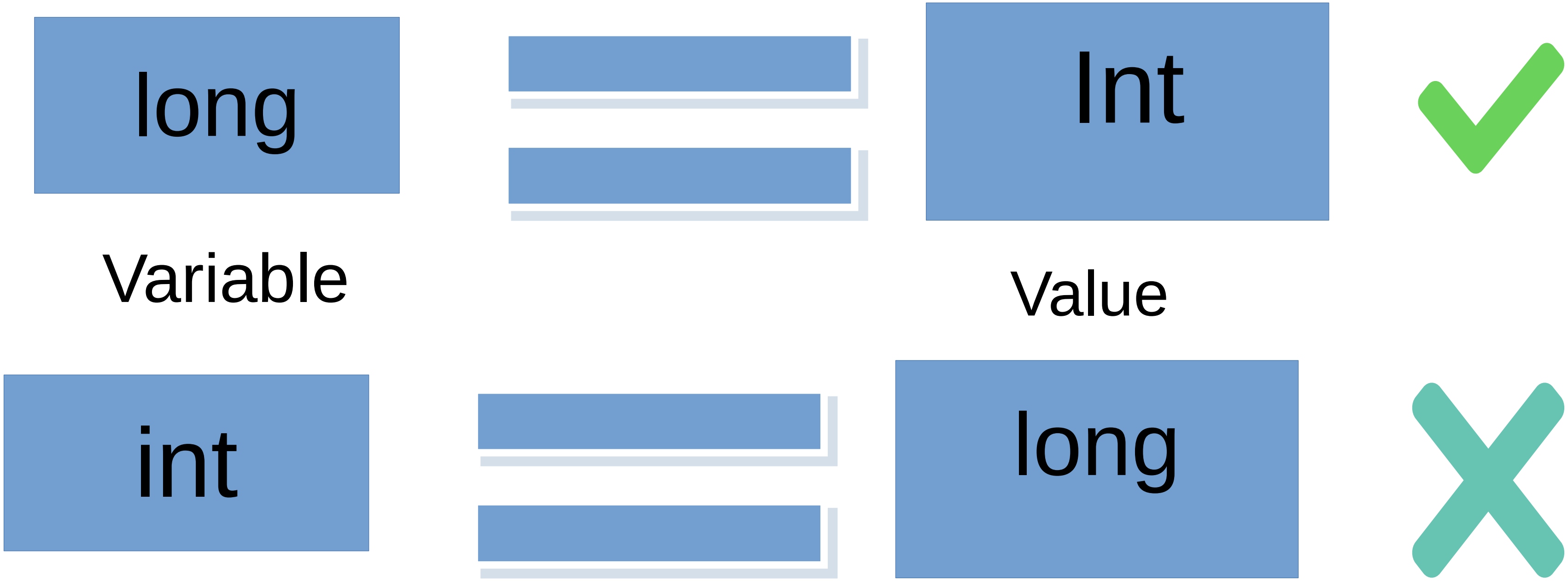
Type Casting

When we need to store one type of value into some other type in such cases we use type casting.

For example: when method returns a type which is different than what we require.



long = Int ✓

Variable

Value

int = long ✗

Implicitly not allowed

2 categories of type casting

01 Widening Conversion

Also called automatic conversion

02 Narrowing Conversion

Also called explicit type casting

Widening Conversion

Automatic conversion will take place if following conditions are met:

i) The 2 types are compatible

ii) Destination type is larger than source type.

When these 2 conditions are met, widening conversion takes place.

type	No. of bits
byte	8
short	16
char	16
int	32
long	64
float	32
double	64

```
Ex: double x = 18  ✗
    float y = x;

    float x = 18  ✓
    double y = x
```

Narrowing Conversion

- Also called explicit type casting
- When a value has to be explicitly made smaller so that it will fit in the target type, it is called narrowing conversion.
- To create a conversion between 2 incompatible types we must use a cast.
- Syntax: (target type)value;
- Eg: (int)2.44;

Automatic Type Promotion

- Java has automatic type promotion if there is more than one type of value available.
- The Rule is that if one operand is of type byte, short, char it is promoted to int.
- And if the operand is long, float, double it is promoted to same.

Thank you!

Growing Liberty by Budi Aryo @tokofoss