



Introduction to Java: Data Types

In this tutorial, we'll learn about data types in Java. This tutorial is part of the “Java for Beginners” series.

The purpose of having data types is to let Java know what each variable represents. Just like in the real world, the “zebra” word represents textual data, while 29 represents numeric data.

In Java, we have two different data types – primitive and reference types. We use them while defining variables, parameters, and return types.

1. Primitive Data Types

Java comes with 8 embedded data types. They represent the basics for building complex (reference) types.

boolean	true / false
byte	8-bit whole value
short	16-bit whole value
int	32-bit whole value
long	64-bit whole value
float	32-bit decimal value
double	64-bit decimal value
char	16-bit Unicode value

2. Numeric Data Types

We use byte, short, int, and long types to represent the whole numbers, while the float and double represent floating-point numbers.

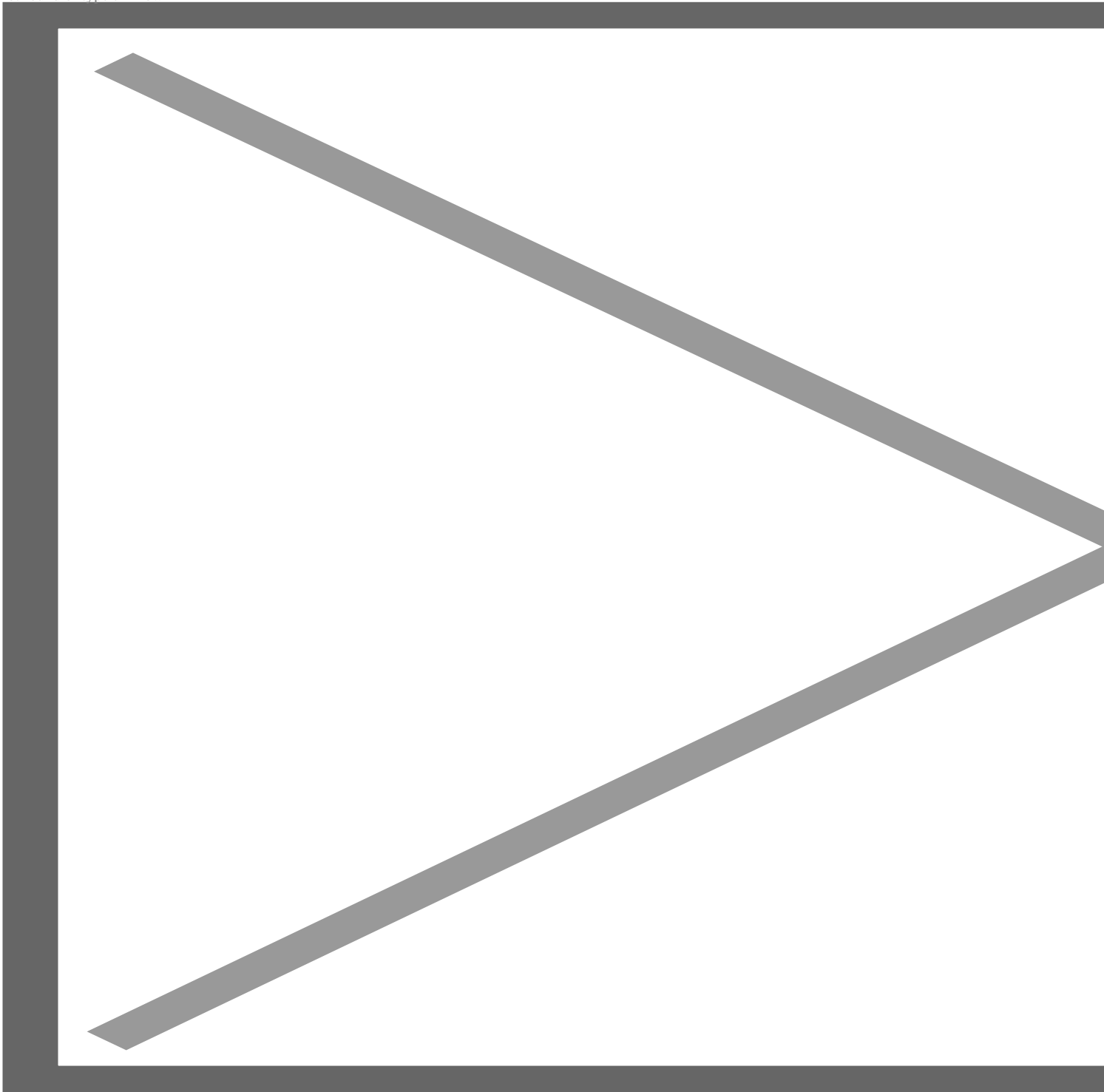
For instance, let's assign the value 5 to the int variable number:

```
int number = 5;
```

Whole numeric data types have different sizes. To put it differently, they have different maximum values we can store in them.

We can imagine the size of the numeric data types like squares:

Image not found or type unknown



The same applies to floating-point data types:

"Decimal

Image not found or type unknown

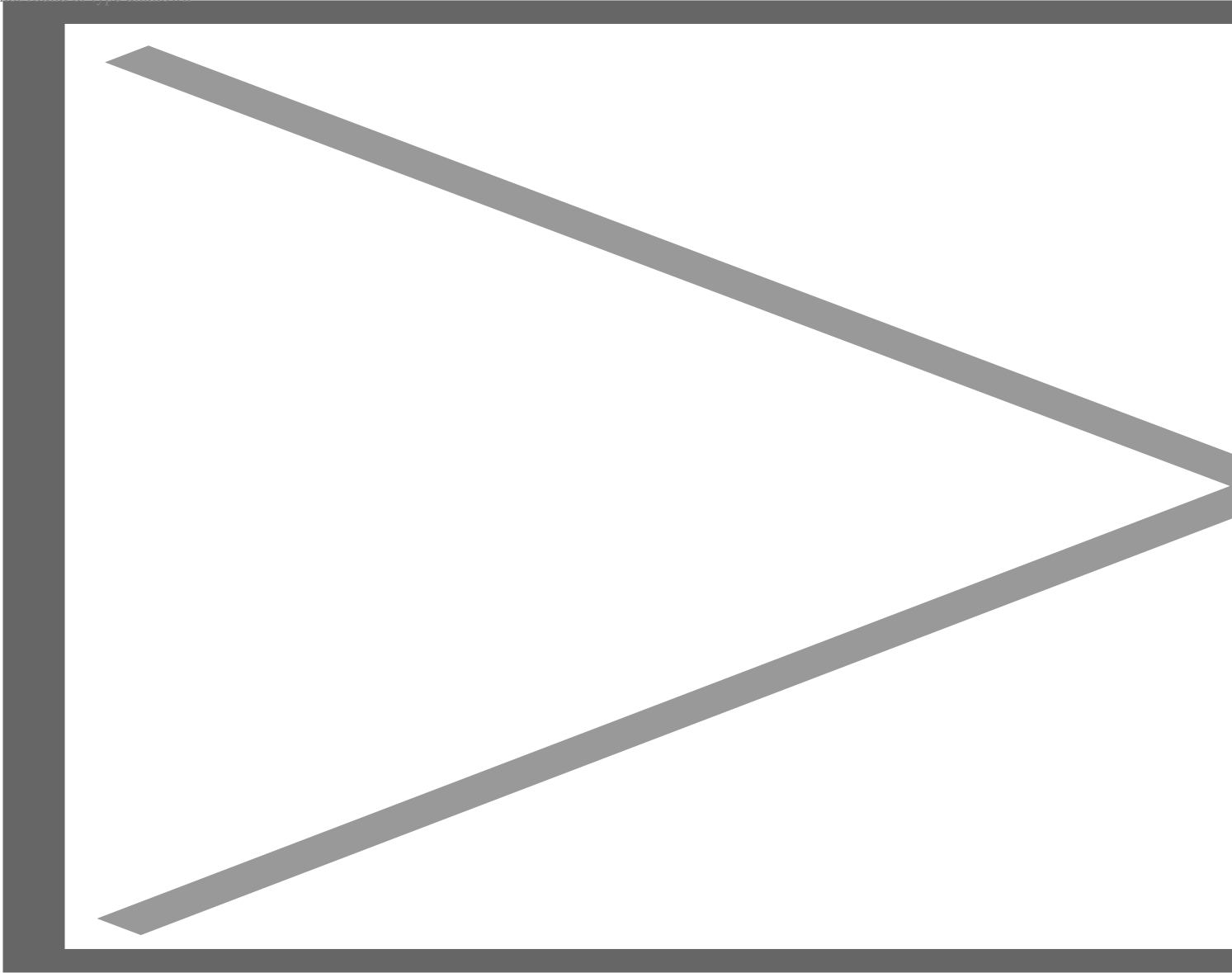
Moreover, when deciding which type to use, we should be careful. We need to make sure that the value we are going to save fits in the variable.

Now, let us imagine two boxes with different sizes. If we moved things from a smaller box to a bigger one, we wouldn't have any problems. But what about when we move things from a bigger box to a smaller

one? There would be a problem. We could move some things, but not everything.

The same logic applies to data types:

Image not found or type unknown



To save us from data spillage, the compiler does not allow us to put things from the bigger box into the smaller one. We cannot convert the value of type long into the int variable by default. We can use the cast operator to explicitly tell the compiler “I know what I am doing.” to perform the mentioned conversion.

It is important to note the float data type requires “f” after the numeric value. Why? Java assumes every decimal value is of type double , so we need to explicitly say we want float data type:

```
float a = 1.5; // this will not work  
float b = 1.5f;
```

3. Boolean

Now, let’s discuss the boolean data type. It could contain one of the two values: true or false.

We would use this type to store values that can be either true or false. For instance, the light can be on (true) or off (false).

```
boolean light = true; // the light is on
```

Additionally, we often use them in if-else statements.

4. Char

Lastly, we use the char data type to store a single character. The value must be put inside single quotes:

```
char c = 'a';
```

It is important to note we could assign numbers to a char type as well. However, it will not represent a number but the character from the [Ascii code](#).

```
char c = 65; // 65 is decimal representation of 'A'
```

5. Key Takeaways

In this tutorial, we learned about primitive data types in Java.

To sum up, Java comes with 8 data types – byte , short , int, long , float, double , boolean, and char. Numeric data types differ in size and we should pay extra attention when deciding which one we will use. Additionally, the boolean type can be either true or false. Finally, we store a single character in the char data type.

[Read More](#)
