

Data Types

In this lesson, we will discuss primitive data types in Java.

We'll cover the following ^

- Numeric data types
 - Integer
 - Floating points
- Textual data types
 - Char
- Boolean data type
- Null data type
- Size of data types

Numeric data types

This classification contains all data types that are numeric.



Did you know?

Variable Types are more often referred to as “**Data Types**”.

Integer

An *integer* is a whole number; that is, it does not have any decimal places.

Examples of an integer include **4**, **5**, **2000**, whereas numbers that are not integers include **4.3**, **4.55**.

Within integers, there are different data types that are based on the space they take within memory. The types include `short`, `int`, and `long`.

```
// A short type is a 16 bit signed integer
short age; // Stores from +32,767 to -32,768
```



```
// An int type is a 32bit signed integer
int age; // Stores from +2,147,483,647 to -2,147,483,648

// A long type is a 64 bit signed integer
long age; // Stores from +9,223,372,036,854,775,807 to -9,223,372,036,854,775,808
```

Note: Data types in Java are all signed. This means that they can store positive as well as negative values of integers.

Floating points

This data type stores numbers that include a *decimal point*. Hence they will include numbers like **4.34** or even **12233.333922**. Keep in mind that floating points will only store an approximation of the decimal value and not the *exact* value.

Floating points have two types, **floats** and **double**. The primary difference between the two is that **double** is able to store more places after the decimal point more accurately than **float**. The **float** data type is fine for most applications. But, scientific applications that deal in really small values like the radius of an atom or really large values like the distance between two galaxies are more amenable to using **double**.

```
// Assigning a float to a radius of a circle

public class Type_Floats {
    public static void main(String[] args) {
        float radius = 8.56f;
        System.out.println(radius);
    }
}
```



Note: If we have to assign a value to float, we must add an 'f' at the end of the number

Now let's look at the assignment of data type double.

```
// Assigning a double to a radius of a circle

public class Type_Double {
    public static void main(String[] args) {
```



```

float radius1=0.00000000000863872078+;
System.out.println(radius1);
double radius = 0.00000000000863872078;

System.out.println(radius);
}
}

```



In the above code, you can see **double** data type stores very small values from precisely than the **float** data type.

Textual data types

This data type class deals with data that holds characters. The data may even have single-digit stored; however, this digit can not be used in arithmetic calculations when used in this context.

Char

A **char** is a 16-bit data type that contains either a single digit or a character. Each character is represented by a unique number according to a standard code like Unicode. Let us look at a few examples below.

```

public class Type_Char {
    public static void main(String[] args) {
        char letter = 'A';
        System.out.println(letter);

        char number_as_text = '6';
        System.out.println(number_as_text);

        char number = 65;
        System.out.println(number);
    }
}

```



Note: Letters in char type must always be in single inverted commas

Note: A character literal is always enclosed in single quotation marks.

Boolean data type

This data type refers only to two values, *true* and *false*. The default value is false. To understand the use of this data type, let's consider an example. Suppose we have an employee, and we have to store details like 'Does this employee have a car', 'Is this employee part-time?' and so on. Here we can simply store the values as 'true' or 'false', making the variable easier to understand and store. This example is illustrated in the code below.

```
public class Type_Bool {  
    public static void main(String[] args) {  
        boolean part_time = false;  
        System.out.println(part_time);  
    }  
}
```



Null data type

Null is a very interesting data type. This basically refers to a value that does not refer to any object. Confused? Basically, we attempt to represent an uninitialized state by using the null value. If we try to read the value of a variable that hasn't been assigned anything yet in our program, we will get the value **Null**.

The following code will generate **Error**, that **line** has not been initialized and hence has a null value.

```
public class Type_Null  
{  
    public static void main(String[] args)  
    {  
        String line;  
        System.out.println(line);  
    }  
}
```



Reading compilation errors: The line that shows an error has a specific structure. It shows the name of the file containing the erroneous line of code, followed by a line number and a description of the error.

Size of data types

The following table describes the different data types we covered in this lesson, along with their corresponding memory requirements.

Data Type	Size in bytes
Integer	4 bytes
Float	4 bytes
Double	8 bytes
Character	2 bytes
Boolean	1 byte

Now that we have an idea about most primitive data types let us look at how to declare a variable in Java.