

JAVA FUNDAMENTALS COURSE

VARIABLES IN JAVA



Author: Ubaldo Acosta

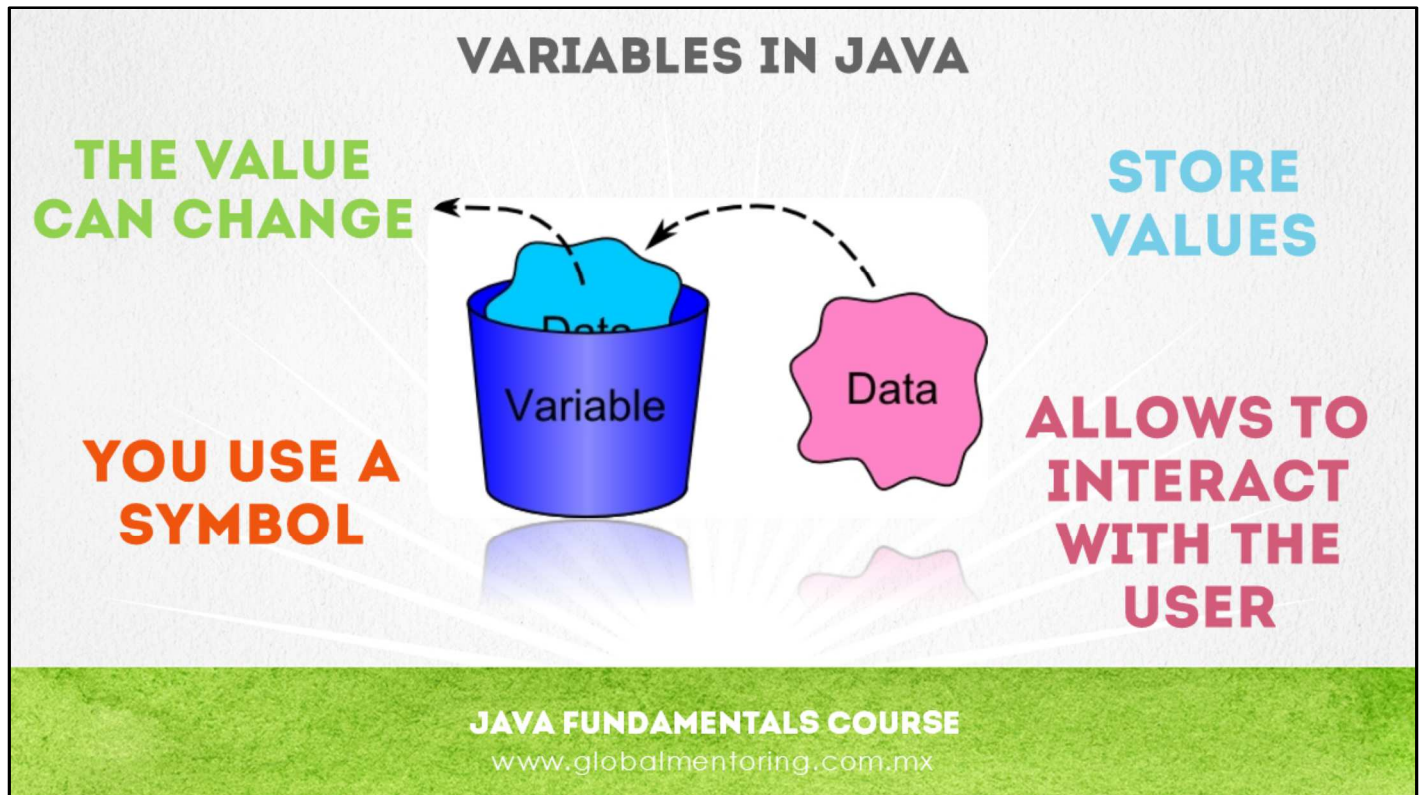


JAVA FUNDAMENTALS COURSE
www.globalmentoring.com.mx

Hello, Ubaldo Acosta greets you. Welcome again. I hope you are ready to start with this lesson.

We will study the concept of variables in Java. This is the basis for the management of information when creating our programs in Java.

Are you ready? OK let's go!



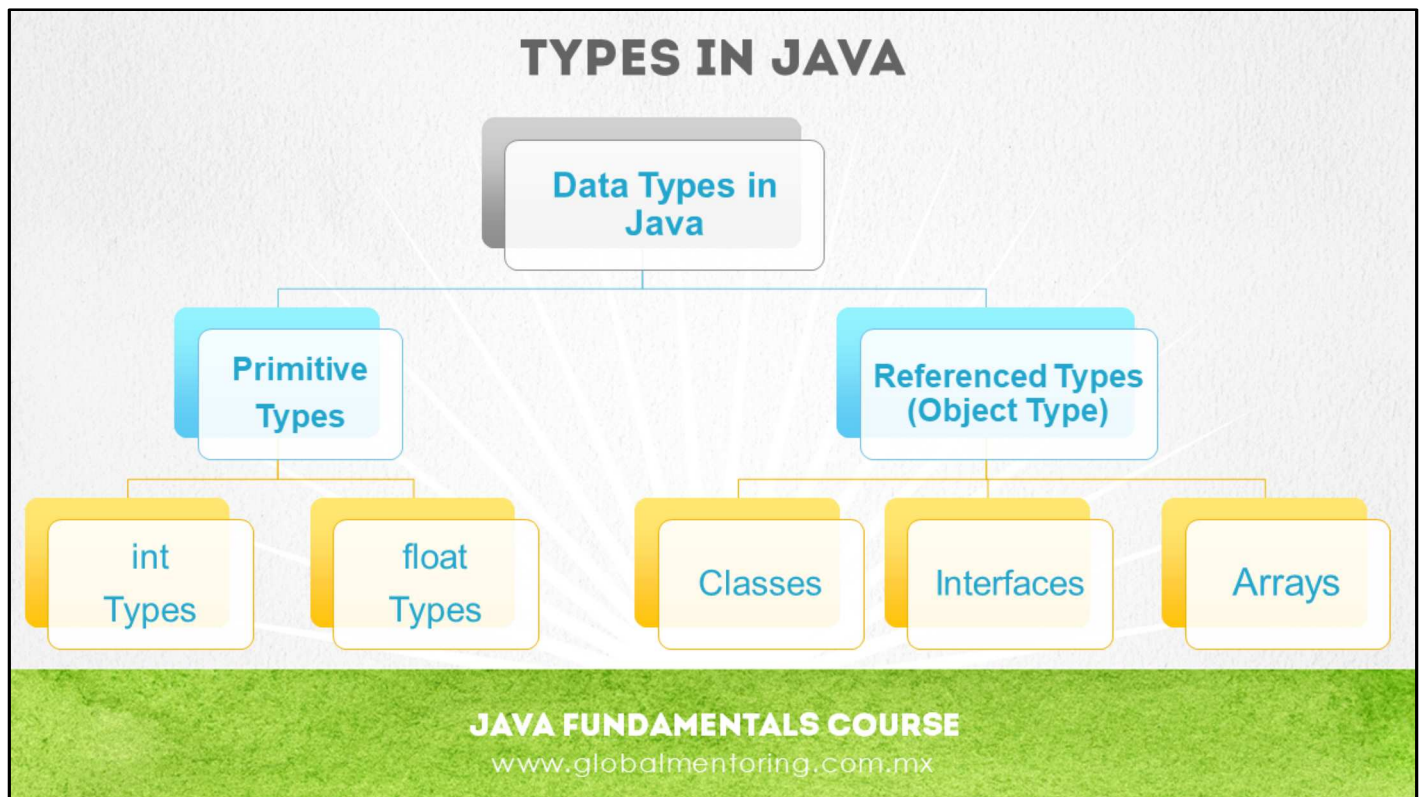
In Java, as in any programming language, to store information, it is necessary to create variables, which will allow us to store data from our program temporarily.

The objective of declaring a variable is to reserve memory space depending on the type that we are going to use.

These variables also allow us to make dynamic programs, so in most cases the values will change during the interaction with the user and the program.

For example, as we can see in the figure we have data, which can be changing throughout the execution of our program, and we have variables, which are what allow us to store our data during the execution of it.

In Java there are different types of data, so we will study in more detail the types of data and therefore the types of variables that we need to create to store a specific type of data. So let's continue.

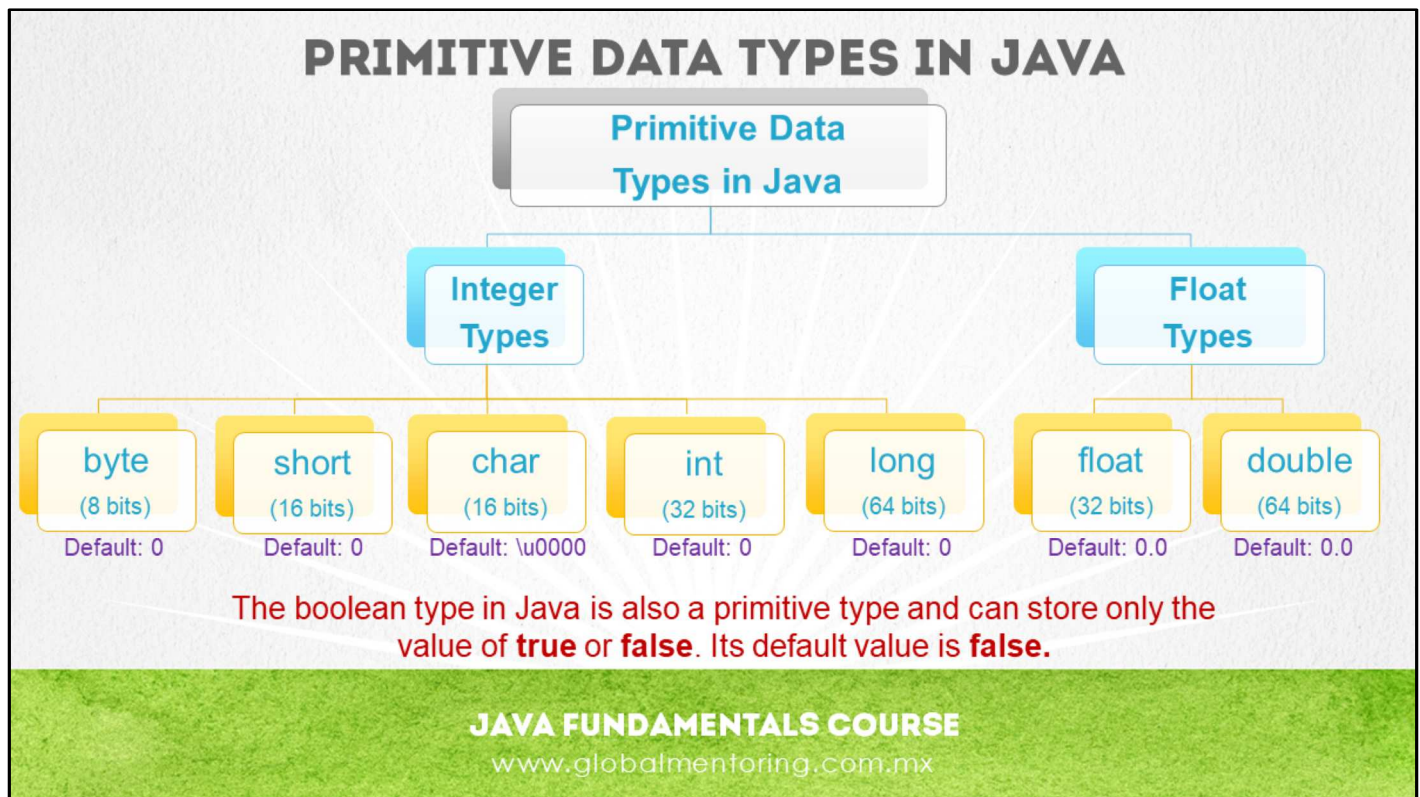


There is a wide classification regarding the types that are handled in Java, however we can summarize it in the figure shown.

On the one hand we have the primitive types and on the other hand we have the types that are considered as extensions of the Object class, also known as reference to objects.

Regarding the primitive types we can classify them in Integer and Floating types, however we also have the Boolean type, each of these types we will see in more detail below.

On the other side, we have the types Object, which can be Classes, Interfaces or Arrays in Java. We will study these types in the topic of Classes and Objects in Java, so for now we will focus on the primitive types and how to declare them and use this type of data in Java.



As we discussed earlier, we have different types of data in Java, in total there are 8.

On the one hand we have the integers, among which we have the byte type which occupies 8 bits. Later we have the short type, which occupies 16 bits. We also have the char type, which occupies 16 bits but manages the UNICODE code to store char values. In turn we have the int type which occupies 32 bits, and finally the long type which occupies 64 bits.

On the other hand we have the floating types, on the one hand the float type which occupies 32 bits, and the double type that occupies 64 bits. The boolean type in Java is also a primitive type and can store only the value of true or false. Its default value is false.

These types of Java are the most basic and are those that we will use if we need the fastest and most resource-saving, with the aim of making our algorithms as efficient as possible. However, in practice we will also use already created functions that can make indirect use of these primitive types, so in any case we will be managing them in our programs, either directly or indirectly.

Each of these primitive types has a default value, which is important to know since when we use them together with our Java classes, the compiler will take into account this value by default according to the type of data we use.

In the figure we can see both the number of bits that each data occupies, as well as the default values that are assigned to each type when they are used as attributes of our classes in Java.

Later we will create several exercises to put into practice the use of primitive types in Java.

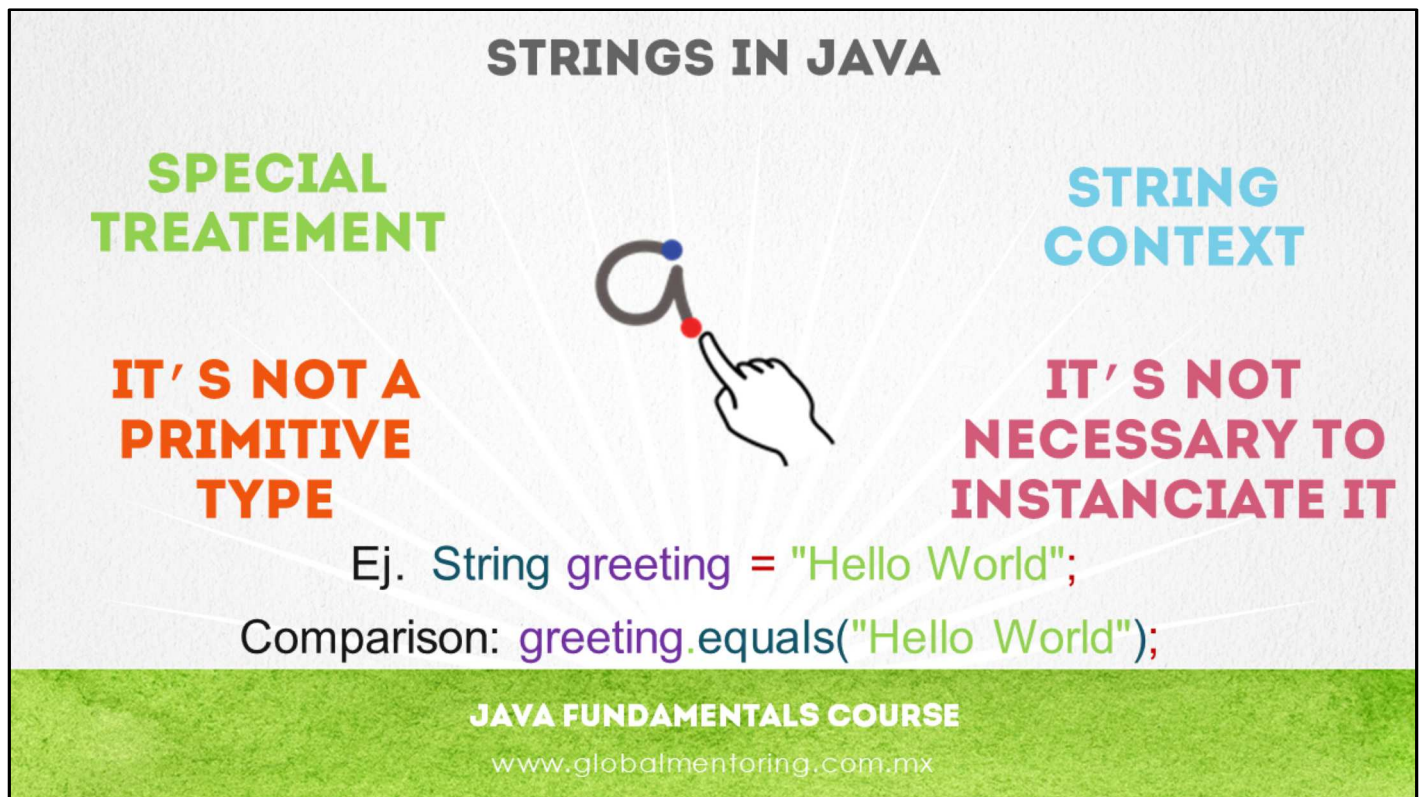
RANGE VALUES OF PRIMITIVE DATA TYPES IN JAVA

Type	Size in Bytes	Range
byte	1 byte	-128 to 127
short	2 bytes	-32,768 to 32,767
int	4 bytes	-2,147,483,648 to 2,147,483, 647
long	8 bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	approximately $\pm 3.40282347\text{E}+38\text{F}$ (6-7 significant decimal digits) <i>Java implements IEEE 754 standard</i>
double	8 bytes	Approximately $\pm 1.79769313486231570\text{E}+308$ (15 significant decimal digits)
char	2 byte	0 to 65,536 (unsigned)
boolean	not precisely defined*	true or false

www.globalmentoring.com.mx

In the table we can see the range of values of the primitive types, as well as the size in bytes that each primitive type occupies.

These ranges are important to determine the maximum or minimum size that a certain type can store.



In Java, string handling is an Object type, but it has several characteristics in particular. Although we do not yet detail the handling of objects, but it should be mentioned that in Java it is necessary to use the new operator to create a new object.

However when we talk about chains this is not necessary. We can simply declare a String type and assign a string value directly to this variable, which is why it seems that the String type is a primitive type but it is not, however due to the frequent use of this type at the time of programming, is that it was decided to simplify the process of creating and assigning values in this type String in particular. Example:

```
String greeting = "Hello World";
```

We can see that to declare a string in Java, you only need to use the String type, later define its name (identifier), and finally assign a value, without the need to use the new operator. Thus, we can directly assign the value of a string, simply using double quotes to wrap the desired value.

In Java, unlike other languages, overloading of operators is not allowed, which means that we can not alter the function of operators to do what we want, however there is an exception to this rule when we handle string types. When we use the + operator and a String type is detected in the operation, we say that we have the String context, and therefore instead of adding values, what Java does is concatenate the values that are in the operation. We will see this later with an exercise, but it is important to note this point since it is another important simplification when using chains in Java.

One of the most important details that we will study later, is the way in which objects are compared in Java, however in this section we will only clarify that when using objects the == symbol is not used as with the primitive types, but the equals method (). In the case of chains this will allow us to compare the content of the chain. In the subject of objects we will study this in more detail, but for now it is important that they know this detail so they know how to compare chains.

ESCAPE CHARACTERS WHEN USING STRINGS IN JAVA

Escape Sequence	Description
\t	Insert a tab
\b	Insert a backspace
\n	Insert a new line
\r	Insert a carriage return
\f	Move to the next page (Form feed). It is used for printers, not consoles.
\'	Insert a single quote
\"	Insert a double quote
\\	Insert a backslash

JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

In the table shown, we can observe the escape characters that are used when displaying messages in the standard output.

These characters will be put into practice in the exercises that we are going to do next.

ONLINE COURSE

JAVA FUNDAMENTALS

Author: Ubaldo Acosta



JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

