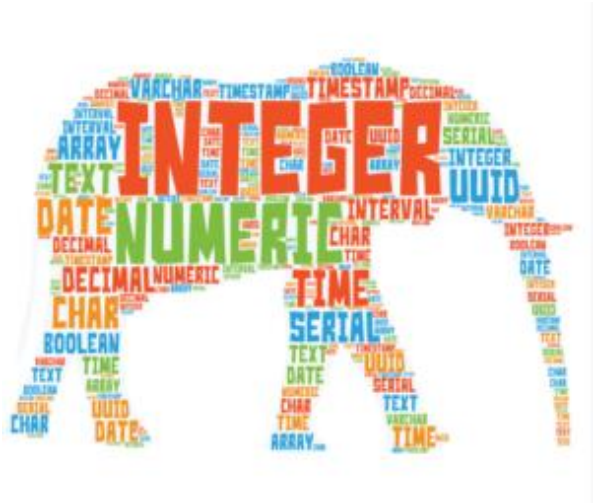

DATA TYPES

Before going ahead with what is data types why do we need it. We should first know **how a data is stored in a system**.

In every electronic system we have **RAM** which **stores the data temporarily**. All the data that we enter is in high level but it is always stored in low level inside the system so that the system/machine can understand.

So **RAM** consists of **several bytes** which consists of 8bits, each bit has two **transistors** which stores **high and low value** (1's and 0's).

What does *Data Type* mean?

The data type of a value (or variable in some contexts) is an attribute that tells what kind of data that value can have. Most often the term is used in connection with static typing of variables in programming languages like C/C++, Java and C# etc, where the type of a variable is known at compile time. Data types include the storage classifications like integers, floating point values, strings, characters etc.

Data types define particular characteristics of data used in software programs and inform the compilers about predefined attributes required by specific variables or associated data objects.

Why Data Types Are Important

Data types are especially important in Java because it is a strongly typed language. This means that all operations are type-checked by the compiler for

type compatibility. Illegal operations will not be compiled. Thus, strong type checking helps prevent errors and enhances reliability. To enable strong type checking, all variables, expressions, and values have a type. There is no concept of a “type-less” variable, for example. Furthermore, the type of a value determines what operations are allowed on it. An operation allowed on one type might not be allowed on another.

Why Data types are required?

Data types are required so that you know what "kind" of data your variable holds, or can hold. If you (and compiler, and runtime and what not) know this information in advance, you can save a lot of runtime issues at compile time only.

For example, in Java, when you declare an int variable, compiler and JVM knows that this variable is going to hold only integer values, nothing else. But when you declare a variable in Javascript using var, you are just declaring a name. This variable can hold any value, be it a primitive type or a complex object. Here, you can use a variable that has string value in an arithmatic expression, and you won't be able to notice until you run and test it.

There are many more such reasons but first, **reasonable flexibility**. Let’s imagine the opposite: a language with a single type. Let’s make that *integer*; you can only have operations on integer numbers. Naturally, this language is very limited in its usefulness: it doesn’t have *strings* to ask people for names or addresses. It doesn’t have *booleans* with which to evaluate the truthfulness of a logical proposition over those numbers and therefore is incapable of decision-making. And finally, as you can only have one type of number, you’re missing on the possibilities of calculating anything that may fall outside that numeric set. It’d be less useful than a pocket calculator.

As you can see, a greater variety data types allow you tackle a great number of problems, and any programming language worth learning should even give you tools to create *your own types*.