Lesson 1.3 Data Types and Variables

For computers to be able to work with data, they need to be able to know what type of data they're working with: if we say 2 + 3 do we want the computer to print those characters? or do we want it to perform that calculation?

We also need to be able to keep track of the data that they were working with in the computer's memory. Data stored in the computer's memory is referred to by a reference to that memory location called a variable.

Let's see how this all works.

Types and Variables

Examples of three most common types that we'll be working with:

1. **int**

2. **double**

3. **String**

Variables are references to memory

Declarations

So to store the value `13` in a variable called `luckyNumber`, you first have to declare the variable and the type of data that you'll be storing in it:

Assigning a value to a variable

Once you've established a variable, you can then use the assignment operator—in Java this is the equals sign **=** to identify the primitive or object that the variable will refer to.

If you know that **`luckyNumber`** will start out with this value, you can combine these two steps into one step:

If you try this, though, you'll get an error:

```
int luckyNumber = “13";
```

If you want to change the value that a variable refers to, use the equals sign again:

Whatever value was stored in **`luckyNumber`** before is no longer available to us. We have overwritten its value with this new, updated value.

Identifiers

There are rules for naming identifiers, some imposed by Java, and some we'll impose on ourselves.

Common conventions: