Lesson 3: Data Types and Variables

3.1 Introduction

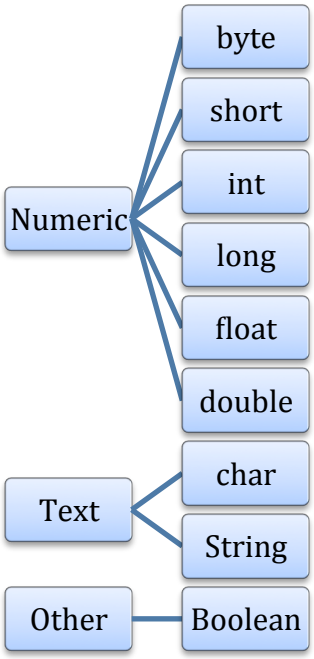
In this lesson we will introduce you to the concept of variables and data types. A natural starting point is to define what exactly these are.

> Variable: A store of value that can be used in place of numbers and phrases, and manipulated using numerous operations.
>
> Data Type: A tool used to tell the computer what it is working with, whether it is a number, phrase, logical true/false or something else entirely.

Think of a variable like a whiteboard at first it is blank or "uninitialized". You then write the number 3 on it. Now you go off to make lunch and come back an hour later and the number 3 is still written on it. Variables work like this, they hold their value unless changed by the program or the user. You can walk up to the whiteboard and add 5 to it, making the whiteboard say 8. Same idea with variables. In Java, we have different kinds of variables, this is where Data Types come into play.

3.2 Data Types

In the most generic sense, we can split variables into three categories: numeric, text-based and other.



If this seems like a lot, the good news is that many behave similarly and are actually quite intuitive. In the last lesson, we dealt with basic arithmetic using

numbers that our computer interpreted as ints. This lead to integer division, which caused the strange rounding problem. Treating one or both numbers as a different type, either double or float, eliminates that problem. How do we do that? You may have solved this on your own already, but all we need to do is add a decimal point. The computer interprets 2 as an int, but 2.0 as a double.

In earlier assignments, we used the word "phrase" to describe the words "Hello World". This is actually called a "String". Strings are denoted by quotation marks and are a complicated data type, but are incredibly important in Java so I am introducing them in this lesson. Strings are actually sequences of Chars, which is short for Character. Chars are denoted by single quotation marks such as a 'a' and can only contain a single character.

3.3 Variables

As mentioned before, variables are stores of value. Variables are powerful because they can be changed throughout the execution of a program and referenced in multiple places. Imagine that you own a bakery selling one thousand different types of cookies. Each cookie costs 1 dollar. Next to each cookie, you have a small sign that says the price of the cookie; customers need to know what they are paying! Suddenly, you decide to raise the price of your cookies from 1 dollar to 2 dollars. Because you repeated the value for each cookie, you now have to change all one thousand signs. What if instead, you had a large board behind the counter that said "all cookies cost 1 dollar". In this situation you only need to change one sign. This is one of the most common features of variables. It enables you to set a value one time, reuse it throughout a program, and change the value in the simplest way possible.

Variables in Java have two parts: the identifier and the value. The identifier is what we use to refer to a variable. Usually, this is a descriptive name. In our cookie example, we could call this identifier "cookieCost". The value in our example would be 1, or after the switch to premium cookies, 2.

How are data types and variables related? A variable always has a type in Java, and each type is only allowed to have certain values. For example an int can only have numeric, whole number values. The chart on the next page can be used as a reference for data types. It contains a lot of information, most of which is unnecessary for your initial understanding. A few key points to remember are that int and double are the most commonly used numeric types. Use an int for whole numbers, positive or negative, and use a double when fractions or decimals are involved. Strings are the most common text based data type, but are relatively complicated when compared to other data types. For now, assume that Strings can not be changed once created, unlike other variables.

	Type	Size	Minimum	Maximum	Default
Numeric	byte	8-bit	-128	127	0
	short	16-bit	-32768	32767	0
	int	32-bit	-2^31	2^31-1	0
	long	64-bit	-2^63	2^63-1	0L
	float	32-bit Floating point	N/A	N/A	0.0f
	double	64-bit Floating point	N/A	N/A	0.0d
Text	char	16-bit Unicode character	\u0000	\uffff	\u0000
	String	N/A	null	N/A	null
Other	boolean	1-bit	FALSE	TRUE	FALSE

3.5 Using Variables

There are two steps to using a variable: declaring and initializing.

Declaring: This tells the computer that we are going to create a variable of a specific type and what its name (identifier) is

Initializing: This assigns a value to the variable that we previously declared.

To declare a variable, select the type you would like to use and a name. It is considered good Java practice to start the name with a lowercase character and use an uppercase character for subsequent words. Let's declare an integer:

```
int myFirstInteger;
```

It is as simple as that. Now let's initialize it

```
myFirstInteger = 8;
```

If you're lazy, like most computer programmers, you can actually combine these two steps into one. The following is a perfectly valid combination of the above two lines.

```
int myFirstInteger = 8;
```

Numeric variables can be treated just like numbers, and text-based variables can be treated just like text. For example:

```
int myFirstInteger = 8;
System.out.println(myFirstInteger+5);
```

This will print out 13, just like if we had simply added 8 and 5. We can also change the value of a variable (except Strings!) like this:

```
int myFirstInteger = 8;
myFirstInteger = myFirstInteger + 5;
System.out.println(myFirstInteger);
```

This also prints out 13.