Transcript: Variables Values

>> Now let's consider variables. A variable is a name for a location in memory that holds a value. Essentially, variables allow us to store data in the computer and then work on the data. There are many types of values, or data. We have integer values. Examples might be -60, 0, 1, and so on.

Notice no decimal point. Then we have floating point values, which do have a decimal point. 5.6, 0.0, and so on. Then we have character values that represent a single character in the character set. And then boolean values, which only can be true or false. Notice that true and false are in blue or purple, indicating that these are reserved words.

So you can't use true and false for anything except assigning to a boolean value. And then everything else is gonna be, turn out to be references to objects. So let's begin by focusing on int type variables, that hold integer values, and then we'll examine the other types later.

Int is, is one of the integer integer types rather and there are four of them. This is a 32 bit integer. So a variable must be declared with a type so that the computer will know how to interpret the value, that's it's gonna hold. For example if we want to declare an int, we could say int total.

Total is our variable name. Notice it begins with lowercase. All variable names and methods, begin with lowercase, with respect to our style checker. We could declare, more than one variable on a single line by, separating them with commas. So here we've got int count comma, temp comma, result semicolon.

Now we need to give the variable an initial value, and you can do that when you declare it. Here we're declaring int sum and we're getting it the initial value zero. Int base equal 32, max equal 149 as an example declaring two on one line. Now, when you reference a variable in a program, whatever its value is at that time is what's used.

So up above we set base equal to 32, and then if we said System.out.println, base is, as in the string. Base is plus base, well base is a variable that holds 32, and so, 32 will be concatenated with base is and we would print out base is 32.

Transcript: Variables Values

An assignment statement actually changes the value of a variable. In this case, total has to have already been declared, and, we're going to, assign 55 to it. We call it the assignment operator. It looks like an equal sign, and, we just, take what's on the, right side, in this case 55 but it could have been a more complicated, arithmetic expression.

It would get evaluated, and then whatever, that result is, it's stored on the left. In this case, it, it's stored in total, and the previous value is overwritten. Now, Java is strongly typed and the variable type and expression must be compatible. For instance in this example we're, we're, we're storing an int, 55, in the total, which was declared an int.

If we attempted to assign 55.0, which would make it a float or a floating point value, a double, it actually wouldn't compile. So, so again, the types have to be compatible, and we'll talk more about that as we get there. So, let's take a look at some, variable examples, in a program.

This again, is a simple program where we're going to, begin with up here, declare int j, and notice we're not assigning it anything. And then, below that, we're gonna declare, declare int m equals zero. Int n equals 10, p equals 20. Then we're gonna print out m, and the string m equals in front of it.

And then n equals n, p equals and the value of p. And here on line 24 we're going to assign j to be 9999, so 9,999. And notice we hadn't assigned j before. This is its first assignment. Then finally, we would, we're gonna print out j. Now I've set a break point up here on line 14.

But it turns out we can't stop there because, as far as Java concerned, is concerned, j really hasn't been initialized, so it really can't be viewed, so it doesn't let us stop there. So, if I hit the, debug button, notice put a little X on that stop sign, went all the way to line 18.

So, we're about to, assign, m. We're going to create m and assign it zero, so when I step we should see it over in the variables tab. So, here's m equals zero, and it's of type int. And next statement gets us n and p. We see they've been assigned 10 and 20.

And now these are simply print statements. And this, third one here on line, 23 will print out p equals, and the value of p, 20. Now finally on line 24, we're about to assign j to, 9,999. And this is the first time j will appear over in the variables tab.

There it is. And then we print out j. Now just as a side here, you can see the output and and the, variables over in the variable tab. If you go to interactions, these are live variables over here. So in interactions, you can actually make assignments. So if we wanted to let m equal, 50.

We could say m equals five zero. And if we want to treat that as a statement, we could put a semicolon. And, we see m got updated to 50 over here. The red means it just changed, and so on. So you can interact with your running program in Interactions.

That, that's essentially what we just did there. So looking at primitive data, just to sorta sum up here, and, and, revisit. We got, eight primitive data types in Java. Everything else are called reference types, and we'll visit those a little later. But looking at the eight primitive types, four of them are integer types.

We've got byte, short, int, and long. An example there of int is int age equals 19. We declared age to be of type int. In most of our work we will use int, and that's why I've got it bold here. A byte is 8 bits, a short is 16 bits, int is 32, and a long is 64 bits.

In most of our work we can get by 32 bit, so int will be the popular one. Then we have floating point types. These are numbers that you can actually put decimal point in. And in this example here, we're declaring average to be of type double. And, and assign it 94.8.

The other type is float. Float is a 32 bit floating point type and a double of 64 bits, and you'll see that we'll mostly use double. Now part of the reason for this is an integer literal, 19, is actually type int and the floating point literal, 94.8 is actually of type double.

And so since this is a double anyway we mostly use doubles here. The next primitive type is char, and, and, it's declared with char as we see here. So we're, we're declaring char letter equal, in this case, the letter, the capital letter A. Notice the single quotes there.

Transcript: Variables Values

And then the last type here, the last primitive type is, boolean. And notice although boolean is a person's name, George Boole, we're actually using it lower case in Java. And so we say boolean isCold equals false. And it's this boolean and it can only take on the values true and false, and this is an example here where we're letting it be equal to false.