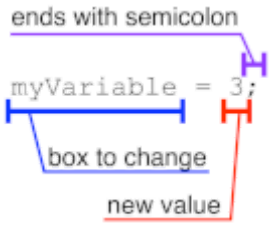
Assigning a Variable

For variables to be useful, we must be able to change their values. To accomplish this, we use assignment statements—statements which change the value contained in a box. An assignment statement starts with an lvalue on the left. An lvalue (pronounced “el-value”) must be something that “names a box”—indicating which box the assignment statement will change. The simplest lvalue is a variable, which names the variable’s own box. (Later we shall see how to name boxes in other ways, but for now, we will only consider variable names.) After the lvalue, comes a single equals sign (called the assignment operator), followed by an rvalue on the right, then a semicolon. The rvalue (pronounced “are-value”) must be an expression whose value shall be placed in the box.

An assignment statement

An expression is a combination of values, and operations which evaluates to a value. For the moment, we will just consider numeric constants (such as 3), which evaluate simply to themselves (that is, 3 evaluates to the number 3). We will discuss more expressions shortly. Evaluating any assignment statement is a matter of figuring out what box the left side names, evaluating the right side to a value (e.g., a number), and then changing the value in the box named on the left side to the value from the right side.

The figure below shows an example of an assignment statement, and identifies the individual pieces. This assignment statement assigns the value 3 to the variable myVariable. Its effect is to change the value in the box named myVariable to be 3.



Declarations and Assignments

The declaration and initialization—the first assignment—of a variable may be combined into a single statement, such as int x = 3; which has the same effect as the two individual statements int x; x = 3;. The next video shows the execution of a combination of variable declarations and assignment statements.

✓

Completed