# Variables and Assignment

Each set-off line in this section should be tried in the Shell.

Try

width = 10

Nothing is displayed by the interpreter after this entry, so it is not clear anything happened. Something has happened. This is an assignment statement, with a variable, width, on the left. A variable is a name for a value. An assignment statement associates a variable name on the left of the equal sign with the value of an expression calculated from the right of the equal sign. Enter

width

Once a variable is assigned a value, the variable can be used in place of that value. The response to the expression width is the same as if its value had been entered.

The interpreter does not print a value after an assignment statement because the value of the expression on the right is not lost. It can be recovered if you like, by entering the variable name and we did above.

Try each of the following lines:

height = 12

area = width \* height

area

The equal sign is an unfortunate choice of symbol for assignment, since Python’s usage is not the mathematical usage of the equal sign. If the symbol ↤ had appeared on keyboards in the early 1990’s, it would probably have been used for assignment instead of =, emphasizing the asymmetry of assignment. In mathematics an equation is an assertion that both sides of the equal sign are already, in fact, equal. A Python assignment statement forces the variable on the left hand side to become associated with the value of the expression on the right side. The difference from the mathematical usage can be illustrated. Try:

10 = width

so this is not equivalent in Python to width = 10. The left hand side must be a variable, to which the assignment is made. Reversed, we get a syntax error. Try

width = width + 5

This is, of course, nonsensical as mathematics, but it makes perfectly good sense as an assignment, with the right-hand side calculated first. Can you figure out the value that is now associated with width? Check by entering

width

In the assignment statement, the expression on the right is evaluated first. At that point width was associated with its original value 10, so width + 5 had the value of 10 + 5 which is 15. That value was then assigned to the variable on the left (width again) to give it a new value. We will modify the value of variables in a similar way routinely.

Assignment and variables work equally well with strings. Try:

first = 'Sue'

last = 'Wong'

name = first + ' ' + last

name

Try entering:

first = fred

Note the different form of the error message. The earlier errors in these tutorials were syntax errors: errors in translation of the instruction. In this last case the syntax was legal, so the interpreter went on to execute the instruction. Only then did it find the error described. There are no quotes around fred, so the interpreter assumed fred was an identifier, but the name fred was not defined at the time the line was executed.

It is both easy to forget quotes where you need them for a literal string and to mistakenly put them around a variable name that should not have them!

Try in the Shell:

fred = 'Frederick'

first = fred

first

Now fred, without the quotes, makes sense.

There are more subtleties to assignment and the idea of a variable being a “name for” a value, but we will worry about them later, in [Issues with Mutable Objects](http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/graphics.html#issues-with-mutable). They do not come up if our variables are just numbers and strings.

**Autocompletion: A handy short cut.** Idle remembers all the variables you have defined at any moment. This is handy when editing. Without pressing Enter, type into the Shell just

f

Then hold down the Alt key and press the / key. This key combination is abbreviated Alt-/. (On a Mac, that may give you a funny character: In that case you need to hold down both the control key and the alt/option key when pressing the ‘/’. This may hold in other places the Alt key is called for in Windows.)

You should see f autocompleted to be

first

This is particularly useful if you have long identifiers! You can press Alt-/ several times if more than one identifier starts with the initial sequence of characters you typed. If you press Alt-/ again you should see fred. Backspace and edit so you have fi, and then and press Alt-/ again. You should not see fred this time, since it does not start with fi.