

Intro. to Programming: Outline for Week 1

Liam J. Cattell

May 18, 2017

1 Programming Languages

A programming language is called *high-level* if it is closer to human languages and further from machine languages. Ultimately a machine language is a made up of 0's and 1's. This is called *micro-code* and depends on what kind of CPU you have in your computer. The next step up in level is called *Assembler*. This is an easier way of wrting CPU instructions using symbols and commands such ADD, MULT, SUB, DIV, JUMP, and TEST.

We will use the high-level language *Python*. There are two basic types of high-level programming languages: *compiled* and *interpreted*. The most widely used compiled languages are C++ and C. Interpreted languages are generally slower than compiled languages since the code is read line by line by an interpreter, but they have the advantage that you don't need to compile them for a particular chip set. These languages are also called scripting languages. Python is a scripting language. You can either write your code into a text file or write directly into the Python shell.

We will see in class how to download Python and use the Python shell. It is possible to compile your script in Python and turn it into faster running machine code but we won't need to do this.

2 Python reserved words

and	elif	global	or
assert	else	if	pass
break	except	import	print
class	exec	in	raise
continue	finally	is	return
def	for	lambda	try
del	from	not	while

3 Variable names

- cannot begin with a number
- cannot contain spaces
- cannot be a reserved word
- cannot contain certain special characters such as:
 - math symbols: * - + / =
 - comment symbol: #
 - parentheses
 - other reserved characters such as: & ! [=
- For example:
 - *myvar* is a valid variable name.
 - *x* is a valid variable name.
 - *8x* is not a valid variable name.
 - *x*y* is not a valid variable name.
 - *else* is not a valid variable name.

4 Assignment of values to variables

The equals sign is used for assignment. The left handside of $x = 8$ is the variable name and the right hand side is the value. In this case x is assigned the value of 8 which is a special kind of number called an *integer*. If I assign x the value 8.0 with the programming statement $x = 8.0$ then this type of number is called a *float*. You might have come across the *real* numbers in your mathematics classes, and float is basically like these numbers. The number is called a float because the CPU handles numbers that are not whole numbers or integers with a floating point processor. This sometimes called a math *coprocessor*.

In addition to number values, you might want to assign to a variable a *string* value, which is another name for text. In this case you must put single or double quotes around the value. For example $name = 'John'$.

NOTE: the equals sign is used differently in Python than in it is in algebra. If you want to determine if x is equal to 8, you would need to use a double equals sign. For example $x==8$.

5 Boolean values

Every programming language needs to use the concept of true or false. In Python these values are called boolean and can be either *True* or *False*. If you type `x==8` into the Python shell then it will return a value of *True* or *False* depending on the actual value of `x`. For example if `x = 9` then the statement `x == 8` will return *False*.

6 Types

Every assigned variable in Python has a type associated with it. Python is a *dynamically-typed* language, meaning that a variable can change its type depending on what value you assign to it. In many languages such as C++ the variable type is fixed and must be declared before the variable can be used. Python has the advantage that you don't need to fix the type of a variable and it can change when you assign some value to it. To find out the type of a variable you use the function *type*. We will learn more about functions in another lesson.

For example, when `x` is assigned 8 with the statement `x = 8`, then `type(x)` will return `< class 'int' >`. When `x = 8.0` then `type(x)` will return `< class 'float' >`. We will go more into what the word *class* means in a later lesson when we discuss *objects*. The types that will concern us most are *int*, *float* and *str*. In English this would be integer, floating point number, and string.